

# An integrated neural network model for pupil detection and tracking

Lu Shi<sup>1</sup> · ChangYuan Wang<sup>1</sup> · Feng Tian<sup>2</sup> · Hongbo Jia<sup>3</sup>

the date of receipt and acceptance should be inserted later

**Abstract** The accurate detection and tracking of pupil is important to many applications such as human computer interaction, driver's fatigue detection and diagnosis of brain diseases. Existing approaches however face challenges in handling low quality of pupil images. In this paper, we propose an integrated pupil tracking framework namely LVCF, based on deep learning. LVCF consists of the pupil detection model VCF which is an end-to-end network, and the LSTM pupil motion prediction model which applies LSTM to track pupil's position. The proposed network was trained and evaluated on 10600 images and 75 videos taken from 3 realistic datasets. Within an error threshold of 5 pixels, VCF achieves an accuracy of more than 81%, and LVCF outperforms the state of arts by 9% in terms of percentage of pupils tracked. The project of LCVF is available at <https://github.com/UnderTheMangoTree/LVCF>.

**Keywords** eye-tracking · pupil detection · deep Learning · convolutional neural networks · long short-term memory

---

Lu Shi  
E-mail: sl31415@163.com

ChangYuan Wang  
E-mail: cyw901@163.com

Feng Tian  
E-mail: ftian@bournemouth.ac.uk

HongBo Jia  
E-mail: jiahongbo@163.com

<sup>1</sup> State and Provincial Joint Engineering Lab. of Advanced Network, Monitoring and Control, Xi'an Technological University, Xi'an, China;

<sup>2</sup> Bournemouth University, Fern Barrow, Poole, Dorset, BH12 5BB, UK;

<sup>3</sup> Institute of Aviation Medicine, Military Medical University, Air Force, Beijing, China;

## 1 Introduction

The analysis of eye information is beneficial to researches in human computer interaction, human mental state analysis, and behavior analysis (Hansen and Ji 2009). The eye information, which relies on eye tracking technology, includes gaze direction, eye mobile trajectory, speed, etc. In recent years, eye tracking technologies have promoted developments in many areas. One widely used area is the driver and pilot fatigue detection. It is remarkable that PERCLOS (percent eye closure) is the detected standard of driver fatigue in the field of machine vision (Dinges and Grace 1998; Dinges et al. 1998; Mallis 1999). But beyond that, eye movement data provide a detailed reflection of cognitive information process (Ford et al. 1989; Lohse 1997). Also, eye tracking has been used in virtual reality as interaction techniques and advertisement design as locating consumer's attention patterns (Meißner et al. 2019; Resnick and Albert 2014; Jaiswal et al. 2019).

Despite being widely applied, accurate and robust eye tracking still faces many challenges. According to (Schnipke and Todd 2000), the pupil detection often suffers from odd shape and size of pupil by occlusion and off-axis of camera, motion blur or camera defocusing. Due to these issues, eye tracking algorithms based on pupil appearance may lose object (Li et al. 2005; Fuhl et al. 2016).

In computer vision and image processing, great improvements in dealing with object detection have been achieved due to the rapid advancement of computer performance and deep learning (Krizhevsky et al. 2012). Compared to traditional approaches, deep learning can give robust feature extraction and powerful representation of objects. Two representative eye tracking algorithms based on deep learning are DeepVOG (Yiu

et al. 2019) and DeepEye (Vera-Olmos et al. 2019). The former is based on pupil segmentation network, thresholding and the direct least squares method (Fitzgibbon et al. 1999), while the latter relies on residual and atrous convolutional network (He et al. 2016; Chen et al. 2017). Both of them have demonstrated a competitive performance in terms of pupil detection and tracking. However, DeepVOG’s pipeline is complex and its performance is prone to the illumination’s influence on the pupil images. DeepEye is more robust to the illumination effect but difficult to deal with the occlusion from eyelid or glasses.

Inspired by the power and effectiveness of deep learning, we propose a new approach to detect and track pupil in this paper. The approach is comprised of two parts: pupil detection and motion prediction. For the pupil detection, we use V\_net (Milletari et al. 2016), a network for object segmentation, as our thresholding tool. We then propose a novel neural network CF to take the output of V\_net and compute the pupil’s parameters which include its radius and center coordinates. We term the detection framework as VCF. For the pupil motion prediction, we use long-short term memory (LSTM) (Hochreiter and Schmidhuber 1997). To develop a model requiring no decisive prior conditions, we have built nine models with different input sequence size and architecture, and then chosen the best one as our final LSTM prediction model. We integrate the two networks above and term it as LVCF. The main contributions of this paper include:

1. We have proposed a novel pupil tracking framework based on neural network, utilizing both the appearance of pupil and timing feature of pupil motion.
2. We are the first to apply LSTM model for learning pupil transition rule, to our best knowledge. The model shows great performance on tracking pupil in video.
3. We have demonstrated the effectiveness and competitiveness of the proposed approach by training LVCF and evaluating its performance with multiple realistic datasets.

This paper is organized as follows: Section 2 describes the related work in the field of pupil tracking model. Section 3 introduces our proposed pupil tracking framework, including its modeling, training, and optimization. Section 4 evaluates the performances of the proposed model from component to the whole system through experiments, followed by the conclusion and future work in Section 5.

## 2 Related Works

Pupil tracker aims to obtain the change of pupil states (center point coordinates, contour, and so on) in eye movement video. The pupil tracking methods can be broadly classified into two categories: appearance-based (see Bhunia et al. 2019) and behavior-based (see Zhu et al. 2002b; Zhu et al. 2002a; Zhu and Ji 2005).

The appearance-based pupil tracking methods tend to detect pupil of each frame in the video. C. H. Morimoto et al. (2000) directly segmented the location of pupil by subtracting bright pupil image and dark pupil image under two near-infrared time multiplexed light sources. Their method is simple and fast, but restricted by experimental device and interfered easily by external illumination. Lech Swirski et al. (2012) divided pupil tracking process into three stages: approximately locate the pupil region by Haar-like feature detector, precisely locate the pupil center by automatic threshold calculation, pupil ellipse fitting by Canny edge detector and a novel Random Sample Consensus (RANSAC) ellipse fitting. Their model tried to solve the problem from pupil ellipse for image captured by head-mounted eye tracker. For their Haar-feature detector bases on pixels value in individual areas, their model fails to detect pupil precisely when there exists similar pixel distribution in an area of image without pupil.

The behavior-based methods track pupil movement mainly relies on temporal information of sequences (see Yilmaz et al. 2006). These methods obtain object information in the first several frame images by object detection algorithm, and then iteratively update object state from previous frames in tracking interval. W. Ketchantang et al. (2005) proposed a method based on dynamic Gaussian Mixture Model (GMM) and Kalman filter. They built a pupil template in gray-level by GMM whose parameters are learned through the EM algorithm. Kalman filter (1960) is then used to predict the location of pupil. Zhi Wei Zhu et al. (2005) proposed a real-time eye detection and tracking model for measurement of eye position under various face orientations by integrating mean-shift and Kalman filter. The model utilizes variable lighting conditions to produce the dark and bright pupil for highlighting pupil region among other image places for detecting pupil. Note that both of the two methods above use Kalman filter as tracking methods. For the pupil transition model has not been specified a prior so far, Kalman filter is applied under the assumption of a linear model, which is a crude approximation to reality.

Deep learning based approaches have recently achieved the state-of-the-art results in many computer vision tasks (Redmon and Farhadi2018; Ren et al. 2015; Good-

fellow et al. 2014; Rajpal et al. 2019). Nianfeng Liu et al. (2016b) proposed a pairwise filter bank based on convolutional neural networks (CNNs) to evaluate the similarity between the input iris image pairs for heterogeneous iris detection. The team (2016a) also put forward hierarchical convolutional neural network (HCNN) and multi-scale fully convolutional network (MFCN) to pupil segmentation without pre- and post-processing. Yuk-Hoi Yiu et al. (2019) utilized fully convolutional neural network (FCNN) with post-processing, and ellipse fitting to obtain pupil parameters, location, and eccentricity of pupil ellipses, which are used to estimate gaze by reconstructing a 3D model of eyeball as input. Cycle Generative Adversarial Networks (GANs) (2017) was used to segment pupil and eyelid by Wolfgang Fuhl et al. (2019) who also applied the generator of Cycle GANs to generate pupil image for increasing sample richness. These deep learning based pupil detection models have improved the accuracy and robustness of pupil tracking compared to conventional methods. However, they have ignored, to a large extent, the timing information for pupil motion. Therefore, we propose in this paper a pupil detection and tracking framework based on both the appearance and behavior information of pupil and develop it through CNN and LSTM.

### 3 Materials and Methods

#### 3.1 Datasets

For this study, we use three datasets for training of VCF and the LSTM prediction model and testing their performances. Some example images are shown in Figure 1.

The first dataset was acquired at our laboratory by recruiting 9 students with healthy eyes (4 females, 5 males, 22-26 years old) from the School of Computer Science and Engineering, Xi'an Technology University, China. All subjects agreed to participate in the study. We utilized a head-mounted acquisition device that places a near-infrared camera in front of the eyes, yielding video-oculography (VOG) videos at a 320x240 pixel resolution and a framerate of 30 Hz. Nine subjects were asked to do random and slow pupil movements for 2 minutes. As a result, we collected 9 videos, each video contains an average of 3600 pictures. To obtain the position and radius of pupil in every frame of these videos, we used a deep learning image annotation tool, VGG Image Annotator<sup>1</sup>, by manually placing a circle on the visible part of the pupil. We tested the LSTM prediction

model with the 9 videos. And then, we excluded those duplicate and inappropriate images (like closed-eye image) and chose 5000 images as dataset A for training VCF.

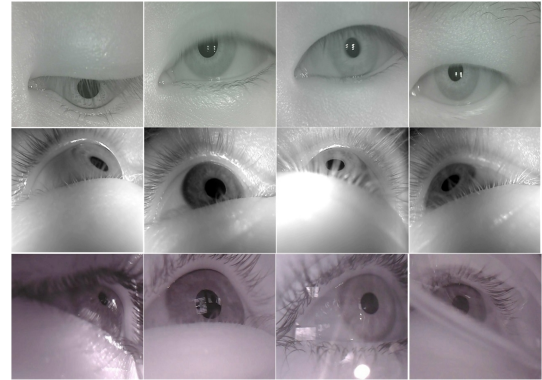


Fig. 1: Examples of images from datasets. The first line of pictures from a dataset acquired in our laboratory, the second line from Świrski, and the third line from LPW.

The second dataset was constructed from Świrski (Swirski et al. 2012), which consists of 3760 pupil images. We chose only 600 images that have been provided with morphologic information of pupil as our dataset B. It was divided into 300 for training and 300 for testing of VCF.

The third dataset was the Labelled Pupil in the Wild (LPW) (Tonsen et al. 2016), which provided 66 videos at a framerate of 120Hz from 22 participants. First, we randomly chose 5000 images as dataset C to test VCF. Then, we trained the LSTM prediction model using 59 videos, and the remaining 7 videos were used to test the LSTM prediction model and LVCF.

#### 3.2 LVCF: pupil tracking framework

LVCF takes raw video frames as input and returns the parameters of pupil including its center coordinates and radius as shown in Figure 2. It consists of LSTM model for prediction and VCF model for detection. LVCF tracker can be expressed in a piecewise function:

$$L(t) = \begin{cases} P(L_t|L_t - 1, L_t - 2), & t \in T \\ F(x_t), & t \notin T \end{cases} \quad (1)$$

where  $L_t$  and  $x_t$  are the parameters of pupil and input frame at time step  $t$ , respectively.  $T$  denotes tracking interval, in which LSTM prediction model  $P$  predicts pupil parameters at the current time based on the pupil

<sup>1</sup> <http://www.robots.ox.ac.uk/~vgg/software/via/via.html>

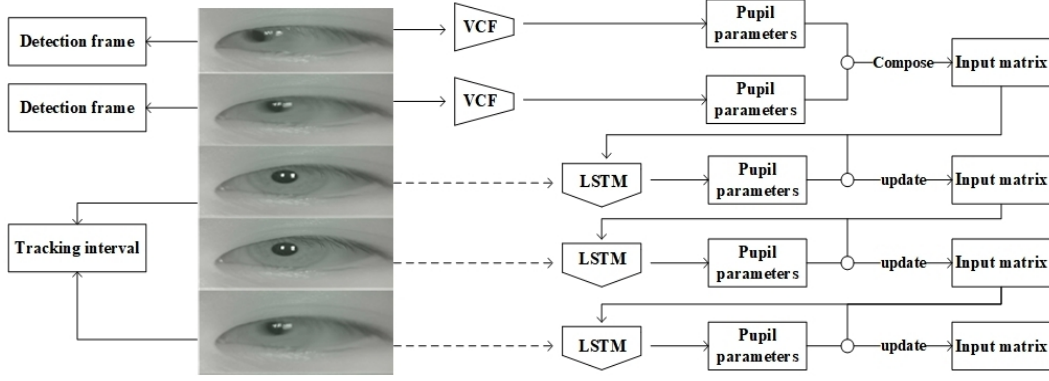


Fig. 2: Overview of LVCF. The dotted line denotes that the parameters of the pupil in the image are predicted through the corresponding LSTM prediction model, and the circle denotes the composition or update of LSTM prediction model’s input matrix. First, the pupil parameters of the two frames are computed by VCF, and the parameters are composed into an input matrix. Then, the matrix is fed into the LSTM prediction model for predicting the third frame’s pupil parameters, from which LVCF enters into the tracking interval. After the tracking interval, LVCF re-enters the detection stage. The process repeats until all frames are processed.

parameters of the previous two frames. In other periods,  $F$  for VCF obtains pupil parameters from input frame for detection. In the following sections, we introduce VCF and the LSTM prediction model in detail.

### 3.3 VCF

*Network architecture* A common approach in pupil detection is to assume that the pupil is the darkest element in the concerned image. The traditional method of detecting pupil is based on segmentation through intensity thresholding. But this approach is easily affected by changes in illumination or camera settings. Taking a different approach, V\_net utilizes the capabilities of fully convolutional neural networks to learn hierarchical representation of raw input data. It has been successfully applied to medical image segmentation. In this paper, we use the modified V\_net (Yiu et al. 2019) which has convolution layer with 10x10 filter and approach padding for achieving up-sampling and down-sampling.

Inspired by YOLO (Redmon et al. 2016), we obtain the parameters of pupil through a single neural network. Because our network consists of CNN and fully-connect layers, we name it “CF”. CF computes the parameters of pupil from the output of V\_net. The parameters are defined by  $x$ ,  $y$ , and  $r$ .  $(x, y)$  represents the center coordinates of pupil, and  $r$  represents the pupil’s radius.

CF divides an input image into a 7x10 grid. When the center of pupil falls into a grid cell, this cell is called responsible cell and used for obtaining the pupil’s parameters. For each grid, there is information about the pupil’s 3 parameters and whether it is a responsible cell.

Therefore, CF’s final prediction is a 7x10x4 tensor. CF has 13 convolutional layers followed by 3 fully connected layers. Eventually, we combine V\_net with CF into VCF as shown in Figure 3, which is an end-to-end approach to compute pupil’s parameters.

*Network training* Because VGG obtains more abstract representations through ImageNet (Krizhevsky et al. 2012), we use the pre-trained weights of VGG as the initial weights of CF’s top 6 layers. Initially, the residual layers are set with randomly uniformed weights.

To increase the convergence rate of CF’s training, the pupil’s  $x$  and  $y$  coordinates are replaced with offsets of the corresponding grid cell location, and the pupil radius is divided by 64 which is usually the upper limit of pupil radius so that these data fall between 0 and 1. We use the ReLU activation function for the convolutional layers, and the sigmoid function as activation function for fully connected layers, except a special layer predicting the responsible grid cell. This “special” layer is set to the softmax activation function. The loss function of CF for training is :

$$\begin{aligned} \lambda_1 \sum_{i=0}^{70} \xi_i^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_1 \sum_{i=0}^{70} \xi_i^{obj} (r_i - \hat{r}_i)^2 \\ + \sum_{i=0}^{70} \xi_i^{obj} (c_i - \hat{c}_i)^2 + \lambda_2 \sum_{i=0}^{70} \xi_i^{noobj} (c_i - \hat{c}_i)^2 \end{aligned} \quad (2)$$

where  $\xi_i^{obj}$  denotes whether pupil appears in cell  $i$ , which is responsible for detection. For many grid cells do not contain the center point of pupil, the output of those

Table 1: Structural information of 9 LSTM prediction models

Model name	model.2.1	model.3.1	model.4.1	model.2.2	model.3.2	model.4.2	model.2.3	model.3.3	model.4.3
Size of input layer	(2,2)	(3,2)	(4,2)	(2,2)	(3,2)	(4,2)	(2,2)	(3,2)	(4,2)
Middle layers	LSTM layer with 25 neurons LSTM layer with 25 neurons			LSTM layer with 50 neurons			LSTM layer with 100 neurons		
Output layers	Fully connect layer with 2 neurons								

cells often overpowers the responsible cell. This phenomenon will cause model instable. To address this issue, we increase the loss from the responsible cell and decrease that from other cells. We set  $\lambda_1 = 5$  and  $\lambda_2 = 0.5$  to accomplish this.

We train the network for 100 epochs on the training data from dataset A and B. Throughout training we use the batch size of 32 and Adam (Kingma and Ba 2014) optimizer. For the first 23 epochs, the learning rate is  $10^{-3}$ , and  $10^{-4}$  for the remaining 77 epochs. We use the weights provided by Yuk-Hoi Yiu et al (2019) for V\_net.

### 3.4 LSTM prediction model

*Model theory* As the basic model of NLP, recurrent neural networks (RNNs) (2015; 2012; 1997; 2015; 1997) are connectionist models with the ability to selectively pass information across sequence steps, while processing sequential data one element at a time. To overcome RNN’s issue on vanishing gradients, Hochreiter and Schmidhuber (1997) introduced the original LSTM model which added memory cell against RNN. Gers et al. (1999) proposed an extended LSTM model by adding forget gates and achieved better performance of NLP. We base our prediction model on this extended LSTM. The calculations of the LSTM model are described by:

$$\tilde{C}^t = \phi(W^{ct}x^t + W^{ch}h^{(t-1)} + b_c) \quad (3)$$

$$i^t = \sigma(W^{ix}x^t + W^{ih}h^{(t-1)} + b_i) \quad (4)$$

$$f^t = \sigma(W^{fx}x^t + W^{fh}h^{(t-1)} + b_f) \quad (5)$$

$$o^t = \sigma(W^{ox}x^t + W^{oh}h^{(t-1)} + b_o) \quad (6)$$

$$C^t = \tilde{C}^t \odot i^t + C^{t-1} \odot f^t \quad (7)$$

$$h^t = \phi(C^t) \odot o^t \quad (8)$$

It can be seen that the model consists of the forget gate  $f^t$  for resetting model, the input gate  $i^t$  for writing to model and the output gate  $o^t$  for reading from model, where  $\sigma(\cdot)$  is the sigmoid function as activation function proposed in Hochreiter and Schmidhuber (1997).  $\tilde{C}^t$  is the intermediate memory cell at the current moment, where  $\phi(\cdot)$  is the  $\tanh$  function designed for Zaremba and Sutskever (2014).  $C^t$  is the intermediate memory cell at the current moment which implements loop constructs with  $\tilde{C}^t$ . The above gates are calculated from the input state  $x^t$  and  $h^t$ , which denotes the value of hidden layer of model at current time  $t$ . In the learning process of LSTM, all  $W$  and  $b$  of the model are trained towards the output with increasing accuracy compared with label. The above equations are common to multiple-layer LSTM, where each layer takes input from the layer below at the same time step and from the same layer at the previous time step.

*Model architecture* At each time step  $t$ , the LSTM prediction model takes in an input matrix composed by pupil location of previous frames. We build nine models with different sizes of previous pupil state sequences, and choose the one with the best performance as the final LSTM prediction model to establish LVCF. The details of these models are shown in Table 1. The pupil state consists of 2 predictions: the pupil’s center coordinates,  $x$  and  $y$ .

*Model training* We use a linear activation for every layer of all nine models and optimize the Mean Absolute Error (MAE) given by Equation (9) in the output

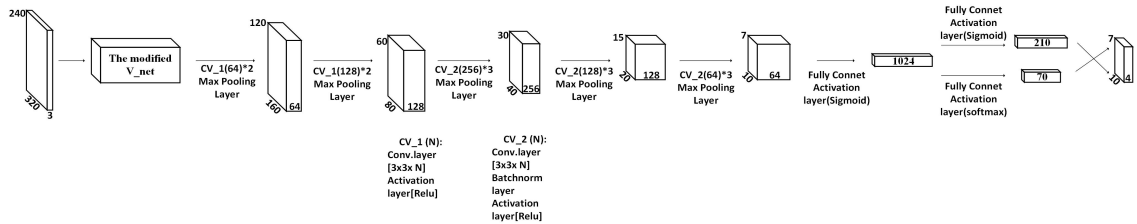


Fig. 3: Architecture of VCF. The modified V\_net has 7 convolutional layers and 8 de-convolutional layers. The subsequent 13 convolutional layers and 3 fully connected layers form CF.

Table 2: Training of 9 LSTM prediction models

Model name	model.2.1	model.3.1	model.4.1	model.2.2	model.3.2	model.4.2	model.2.3	model.3.3	model.4.3
Batch size	1024	5120	1024	5120	5120	1024	6500	6500	6500
Epochs	10	30	10	40	20	10	50	50	50
Optimizer	Adam Kingma and Ba (2014)								
Learning rate	$10^{-4}$								

of all models. To avoid overfitting, we use the dropout layer (Hinton et al. 2012) with rate 0.5 after every layer of LSTM prediction model.

$$LOSS_{MAE} = |L_{output}^i - L_{label}^i| \quad (9)$$

Different models need different training to obtain the optimum weights. The training methods corresponding to 9 LSTM prediction models are shown in Table 2.

## 4 Results and Discussion

### 4.1 Pupil detection

*Accuracy* In this experiment, the accuracy of pupil’s center detection is measured by the Euclidean distance between the output coordinates of VCF and the labeled coordinates (i.e. the ground truth), while the accuracy of pupil’s radius is measured by the absolute distance between the output radius of VCF and the labeled radius. We compare VCF with DeepVOG (Yiu et al. 2019) and DeepEye (Vera-Olmos et al. 2019) on the dataset B and C.

First, we compare the results visually, by showing some examples of pupil detection in Figure 4. Apparently, VCF has achieved competitive results in both pupil’s center detection and radius. Note that DeepEye doesn’t detect radius, but it gives good results in center detection.

Then, we evaluate the three algorithms quantitatively by comparing the percentage of images in which the error of the pupil detection is within a threshold of 1-5 pixels. The results are illustrated in Figure 5. It can be seen from Figure 5a that VCF has achieved a much higher detection rate of over 21% within the error threshold of 1 pixel, while DeepEye has a detection rate of 10% and DeepVOG even less than 5%. However, for the thresholds of more than 1 pixel, VCF’s performance drops. In the next subsection, we provide detailed analysis of the drop. For the dataset C, Figure 5b shows that DeepVOG has the best performance while DeepEye gives the worst. VCF’s performance increases with bigger error thresholds and it achieves an accuracy of more than 81% in terms of percentage of

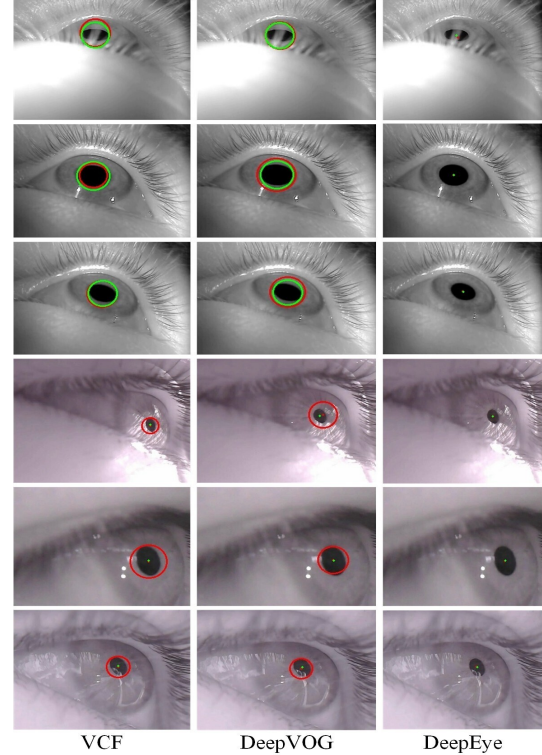


Fig. 4: Example results of VCF, DeepVOG, and DeepEye. The first three rows are sampled from dataset B, and the rest from dataset C. The red circles or dots denote the detection result (pupil’s center or radius) while the green ones denote the ground truth. When the detected center is nearly perfect, i.e. with minimum error, only is a green dot shown.

pupils detected within the error threshold of 5 pixels, which is competitive to DeepVOG. Figure 5c shows the error distribution of the radius detection of DeepVOG and VCF. Clearly, VCF obtains a smaller radius detection error than DeepVOG. Note that the dataset C doesn’t have the ground truth of the radius, so it’s not used for the comparison on radius.

*Error analysis* We can see from Figure 5 that, compared to other approaches, VCF does not always produce the best results. In this section, we dissect the process of VCF to analyze and understand the cause of



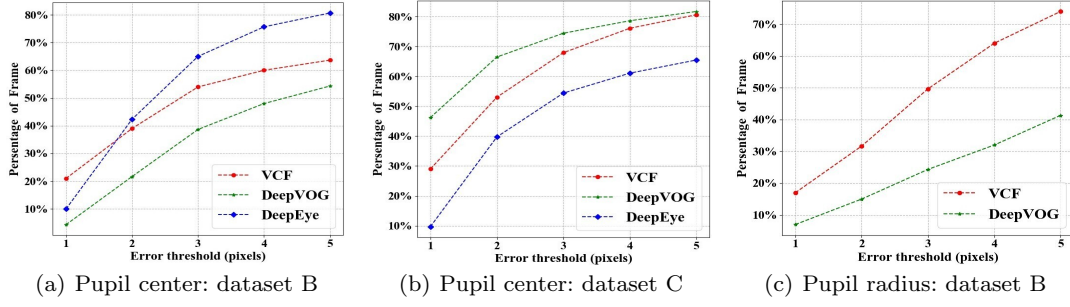


Fig. 5: Quantitative comparison of VCF, DeepVOG, and DeepEye for the detection of pupil's center and radius.



Fig. 6: Sample of the images which the modified V\_net failed to segment correctly.

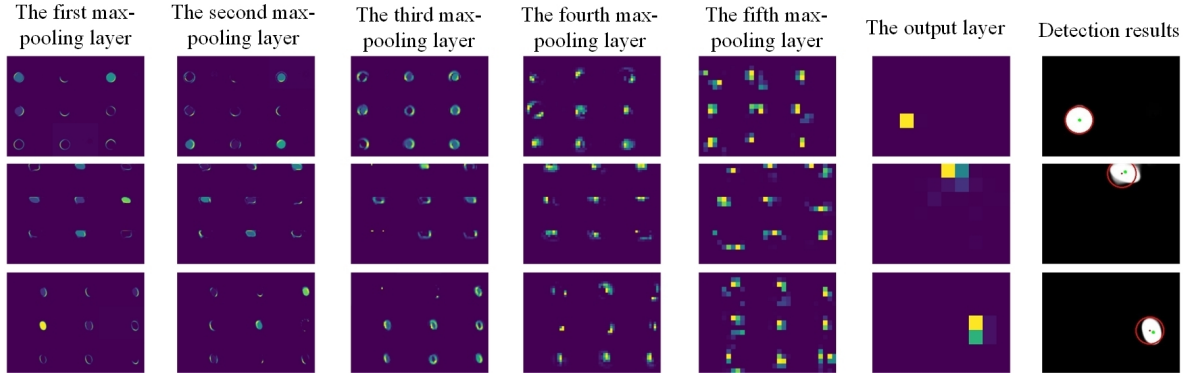


Fig. 7: Visualization of fully trained CF and detection results. The red circles or dots denote the detected center or radius while the green dots are the ground truth of the center. When the detected center is nearly perfect, only is a green dot shown.

failure of detection (with more than 5 pixels of error), particularly on pupil's center.

First, we found that some failures are caused by the modified V\_net which failed to segment the pupil from input image. The segmentation produced completely or nearly black images. This happened to 10% images from the dataset B and 0.04% from the dataset C. In these images, the area around the pupil is shaded or comparably dark, as shown in Figure 6. The similarity of intensity between the pixels in the pupil area and surrounding pixels could be the reason of the failure of segmentation by the modified V\_net.

Then, we analyze the failure cases, in which the modified V\_net does segment pupil correctly, by visual-

izing some convolutional layers and the output layer of CF. Figure 7 shows a random subset of feature maps of the max pooling layer (the last layer of every convolutional stage) and the output layer related to pupil's center. In the feature map of output layers, the higher a pixel's value, the brighter the pixel. We select the brightest one (small yellow block) as final result which corresponds to the responsible cell in the 7x10 grid. The reason for failure is that the brightest one does not always give correct results. For example, it can be seen from the third feature map of output layers that the center of the pupil should have been detected correctly if the green block were chosen as responsible cell. Moreover, as shown in Figure 7, the feature map of the

first four max-pooling layers exhibits the salient feature of the pupil detected, i.e. oval-like; however it is not the case for the fifth max-pooling layer. According to Zeiler and Fergus (2014), the invariance (of the properties such as compositionality of the concerned object) increases as the layer ascends. In future work, we will investigate this inconformity in order to improve the accuracy of detection.

**Ablation study** Compared with the last two fully connected layers of VCF, the role of the other layers is unclear. To evaluate how the layers of VCF affects performance, we conducted another comparison experiment with different structure of the network by removing some layers of VCF, as shown in Table 3. We removed the fully connected layers, FC(1024) for VCF\_D1, the convolution layers CV\_2(64) for VCF\_D2, and the convolution layers CV\_2(64) & CV\_2(128) for VCF\_D3. Note that since the parameter magnitude of VCF\_D3 has reached hundreds of millions, further reduction of the convolution layers would make the model untrainable.

Table 3: Models in ablation study

V_net	CV_1(64)	CV_1(128)	CV_2(256)	CV_2(128)	CV_2(64)	FC(1024)	Parameter	Name
✓	✓	✓	✓	✓	✓	✓	7,351,576	VCF
✓	✓	✓	✓	✓	✓	✓	3,733,400	VCF_D1
✓	✓	✓	✓	✓	✓	✓	41,937,624	VCF_D2
✓	✓	✓	✓	✓	✓	✓	316,007,256	VCF_D3

To eliminate other distractions, we train VCF\_D1, 2 and 3 with the same datasets and approaches as VCF. The images used for testing are randomly selected from LPW. There is no same image in the training dataset and test set. The results are illustrated in Figure 8. It can be seen that, among the four models, VCF gives the best results for all error thresholds. VCF\_D1 is slightly better than VCF\_D2 within an error threshold of 1 pixel, with 32% and 31% accuracy respectively. But the situation changes within the 5 pixels error threshold, in which VCF\_D2 has achieved a much higher detection rate of over 2% than VCF\_D1, with 84% and 82% accuracy respectively. Given a large number of parameters, VCF\_D3 offers the worst performance, with the overall accuracy rate not even reaching 30%. Therefore, VCF is chosen as our pupil detection model.

## 4.2 Pupil tracking

In this section, we conduct 3 experiments. First, we test LSTM prediction models on our laboratory videos and then on the videos from LPW testing dataset to demonstrate the model’s influence factor. In both tests,

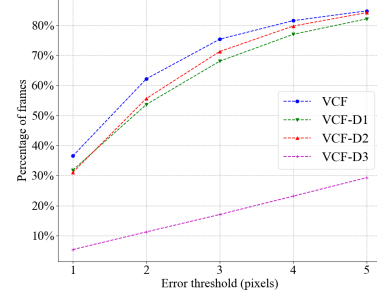


Fig. 8: Detection results of VCF, VCF\_D1, VCF\_D2 and VCF\_D3

we use the labeled center coordinates of the detection frames so that the performance of LSTM prediction model can be independently evaluated without influence of the detection accuracy of VCF. Finally, we test the entire pipeline of LVCF, from detection to tracking, using the videos from LPW testing dataset.

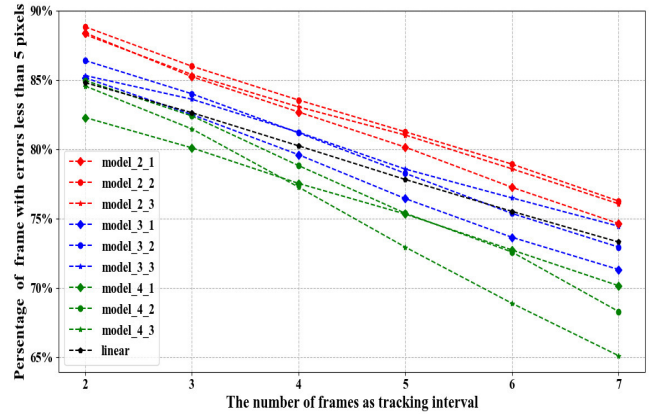


Fig. 9: Results of pupil’s tracking with nine LSTM prediction models and the linear motion model Kalman (1960) on our laboratory videos. Overall, the accuracy decreases when the interval increases, and four of the nine models, i.e. model\_2.1, model\_2.2, model\_2.3 and model\_3.2 are always better than the linear model regardless of the interval.

**Tracking comparison and model selection** Like the experiment for VCF, the accuracy of tracking in this experiment is measured by the Euclidean distance between the pupil’s center tracked and the labeled (ground truth), within an error threshold of 5 pixels. Since the prediction interval of tracking process is an important parameter for the stability of our model, we took the experiment on videos acquired from the laboratory with



6 different intervals. Besides, we compare our model with a linear motion model, the state transition function of Kalman filter. The tracking results in Figure 9 show that, with the increasing of the interval, the accuracy rate decreases for all LSTM models. Because the difference of running time between LSTM prediction model and VCF is up to 55ms in the experiment with a NVIDIA GeForce RTX2060 graphics processing unit (GPU), we set the tracking interval to 3 which balances the accuracy and speed of LVCF. It can be seen that, among all LSTM prediction models, “model\_2\_1”, “model\_2\_2” and “model\_2\_3” give the better results. For instance, they have achieved 85%, 86%, and 85% of accuracy within an error threshold of 5 pixels, respectively. To further determine the best tracking model, we examined the detailed error distribution against different error thresholds. As shown in Figure 10, the “model\_2.2” achieves the best overall results. Therefore, we use it as our LSTM prediction model with 3 frames as the tracking interval for our subsequent experiments below.

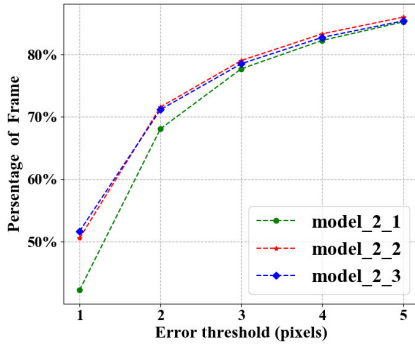


Fig. 10: Prediction results of “model\_2.1”, “model\_2.2” and “model\_2.3”

**Results analysis** We randomly chose one of 7 videos in LPW testing dataset and used 2000 frames to evaluate the model’s tracking performance. With the model\_2.2, Figure 11 shows the tracking trajectory of pupil in the video. This figure shows that LSTM prediction model tracks pupil motion with high accuracy. Our hypothesis is that the framerate of video is one of important factors which affect the predication performance. To prove it, we further conduct two experiments, one with 7 videos of LPW testing dataset with the framerate 120Hz, and the other with 9 laboratory videos of the framerate 30Hz. As shown in Figure 12, the higher frequency gives higher accuracy within an error threshold of 5 pixels, though the low frequency performs better

under an error threshold of 2 pixels.

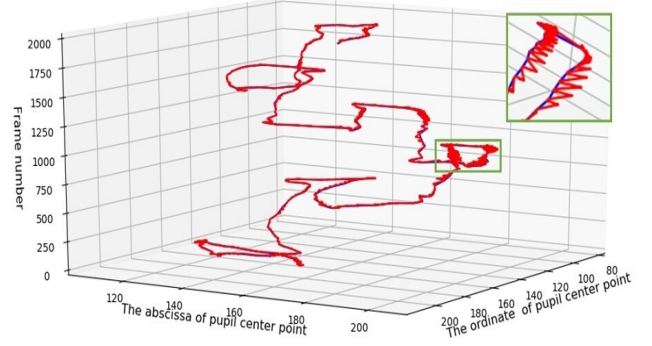


Fig. 11: Pupil’s motion trajectory. The red line illustrates the trajectory of tracking by our LSTM prediction model. The blue line represents the ground truth (of the pupil’s position). Our tracking is very accurate (within the error threshold of 5 pixels), and the failure frames, like those in the green box, only account for 1.8% of total number of frames.

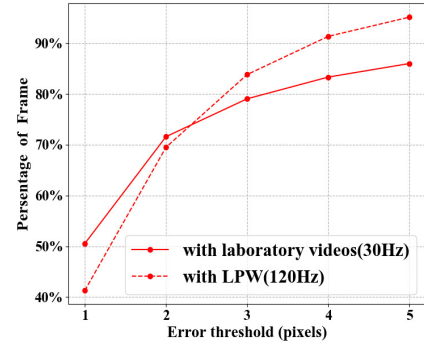


Fig. 12: Prediction results of “model\_2.2” in videos with different framerate.

#### 4.3 Detection and tracking with LVCF

To evaluate the performance of LVCF, we compare it with DeepVOG and DeepEye using videos of LPW testing dataset. In this experiment, we focus on the accuracy of pupil’s center detection and tracking. It can be seen in Figure 13 that, for the error thresholds of 4 or more pixels, LVCF outperforms DeepVOG and DeepEye, but DeepEye produces the best results for smaller thresholds. The reason could be that, when an error occurs in the detection phase, the error will propagate

to the tracking phase until it enters the next detection phase. This is a common problem of behavior-based methods, which we will investigate further in the future.

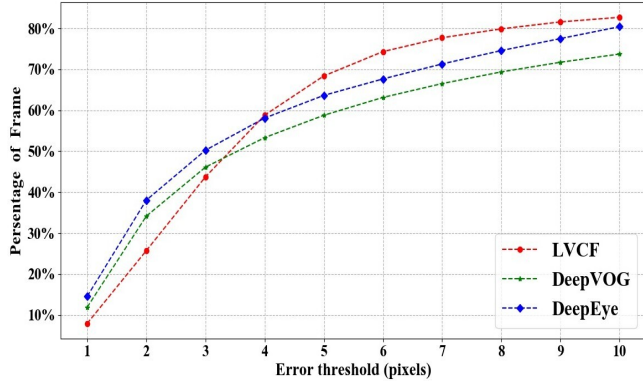


Fig. 13: Pupil’s center detection and tracking with LPW. LVCF achieves the best results - a 68% detection rate with less than 5 pixels of error, which outperforms DeepEye by 4% and DeepVOG by 9%. It can also be seen that LVCF achieves an 83% detection rate with less than 10 pixels of error.

## 5 Conclusion

We have presented a novel pupil-tracking framework, LVCF, which consists of two parts: VCF for detection and LSTM model for prediction. We evaluate VCF on a total of 5300 eye images from two publicly-available datasets in comparison with two state-of-the-art approaches. VCF has achieved competitive performance and detected 81% of frames with less than 5 pixels error. We have demonstrated the application of LSTM for learning the rule of pupil motion and analyzed 9 models of different output layer, middle layer, and size of input layers to establish the best one for tracking pupil. Combining the detection and tracking, LVCF has obtained 68% of accuracy with less than 5 pixels error. The components of LVCF are available at <https://github.com/UnderTheMangoTree/LVCF>. In future work, we will investigate the common constraint of behavior-based methods to optimize the performance of LVCF, and meanwhile research the learned feature at high layers of neural network to further improve the accuracy of detection.

## Funding

Authors gratefully acknowledge the financial support provided by National Defense Science and Technology Innovation Zone (No. ZT001007104).

## Conflict of interest

The authors declare that they have no conflict of interest.

## Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

## References

- Bhunia AK, Bhattacharyya A, Banerjee P, Roy PP, Murala S (2019) A novel feature descriptor for image retrieval by combining modified color histogram and diagonally symmetric co-occurrence texture pattern. *Pattern Analysis and Applications* pp 1–21
- Chen LC, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*
- De Mulder W, Bethard S, Moens MF (2015) A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech & Language* 30(1):61–98
- Dinges DF, Grace R (1998) Perclos: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance. US Department of Transportation, Federal Highway Administration, Publication Number FHWA-MCRT-98-006
- Dinges DF, Mallis MM, Maislin G, Powell JW, et al. (1998) Evaluation of techniques for ocular measurement as an index of fatigue and as the basis for alertness management. Tech. rep., United States. National Highway Traffic Safety Administration
- Fitzgibbon A, Pilu M, Fisher RB (1999) Direct least square fitting of ellipses. *IEEE Transactions on pattern analysis and machine intelligence* 21(5):476–480
- Ford JK, Schmitt N, Schechtman SL, Hulst BM, Doherty ML (1989) Process tracing methods: Contributions, problems, and neglected research questions. *Organizational behavior and human decision processes* 43(1):75–117

- Fuhl W, Santini TC, Kübler T, Kasneci E (2016) Ellipse selection for robust pupil detection in real-world environments. In: Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications, ACM, pp 123–130
- Fuhl W, Geisler D, Rosenstiel W, Kasneci E (2019) The applicability of cycle gans for pupil and eyelid segmentation, data generation and image refinement. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp 0–0
- Gers FA, Schmidhuber J, Cummins F (1999) Learning to forget: Continual prediction with lstm
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680
- Graves A (2012) Supervised sequence labelling. In: Supervised sequence labelling with recurrent neural networks, Springer, pp 5–13
- Hansen DW, Ji Q (2009) In the eye of the beholder: A survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence* 32(3):478–500
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
- Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:12070580*
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural computation* 9(8):1735–1780
- Jaiswal S, Virmani S, Sethi V, De K, Roy PP (2019) An intelligent recommendation system using gaze and emotion detection. *Multimedia Tools and Applications* 78(11):14231–14250
- Kalman RE (1960) A new approach to linear filtering and prediction problems. *Journal of basic Engineering* 82(1):35–45
- Ketchantang W, Derrode S, Bourennane S, Martin L (2005) Video pupil tracking for iris based identification. In: International Conference on Advanced Concepts for Intelligent Vision Systems, Springer, pp 1–8
- Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*
- Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
- Li D, Winfield D, Parkhurst DJ (2005) Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops, IEEE, pp 79–79
- Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*
- Liu N, Li H, Zhang M, Liu J, Sun Z, Tan T (2016a) Accurate iris segmentation in non-cooperative environments using fully convolutional networks. In: 2016 International Conference on Biometrics (ICB), IEEE, pp 1–8
- Liu N, Zhang M, Li H, Sun Z, Tan T (2016b) Deepiris: Learning pairwise filter bank for heterogeneous iris verification. *Pattern Recognition Letters* 82:154–161
- Lohse GL (1997) Consumer eye movement patterns on yellow pages advertising. *Journal of Advertising* 26(1):61–73
- Mallis MM (1999) Evaluation of techniques for drowsiness detection: Experiment on performance-based validation of fatigue-tracking technologies. PhD thesis, Drexel University
- Meißner M, Pfeiffer J, Pfeiffer T, Oppewal H (2019) Combining virtual reality and mobile eye tracking to provide a naturalistic experimental environment for shopper research. *Journal of Business Research* 100:445–458
- Milletari F, Navab N, Ahmadi SA (2016) V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV), IEEE, pp 565–571
- Morimoto CH, Koons D, Amir A, Flickner M (2000) Pupil detection and tracking using multiple light sources. *Image and vision computing* 18(4):331–335
- Rajpal S, Sadhya D, De K, Roy PP, Raman B (2019) Eai-net: Effective and accurate iris segmentation network. In: International Conference on Pattern Recognition and Machine Intelligence, Springer, pp 442–451
- Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*
- Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
- Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
- Resnick M, Albert W (2014) The impact of advertising location and user task on the emergence of banner ad blindness: An eye-tracking study. *International Journal of Human-Computer Interaction* 30(3):206–219

- Schnipke SK, Todd MW (2000) Trials and tribulations of using an eye-tracking system. In: CHI'00 Extended Abstracts on Human Factors in Computing Systems, ACM, pp 273–274
- Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681
- Swirski L, Bulling A, Dodgson NA (2012) Robust real-time pupil tracking in highly off-axis images. In: *Etra*, pp 173–176
- Tonsen M, Zhang X, Sugano Y, Bulling A (2016) Labelled pupils in the wild: a dataset for studying pupil detection in unconstrained environments. In: *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pp 139–142
- Vera-Olmos F, Pardo E, Melero H, Malpica N (2019) Deepeye: Deep convolutional network for pupil detection in real environments. *Integrated Computer-Aided Engineering* 26(1):85–95
- Yilmaz A, Javed O, Shah M (2006) Object tracking: A survey. *Acm computing surveys (CSUR)* 38(4):13–es
- Yiu YH, Aboulatta M, Raiser T, Ophey L, Flanagan VL, zu Eulenburg P, Ahmadi SA (2019) Deepvog: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *Journal of neuroscience methods*
- Zaremba W, Sutskever I (2014) Learning to execute. *arXiv preprint arXiv:14104615*
- Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: *European conference on computer vision*, Springer, pp 818–833
- Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *Proceedings of the IEEE international conference on computer vision*, pp 2223–2232
- Zhu Z, Ji Q (2005) Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. *Computer Vision and Image Understanding* 98(1):124–154
- Zhu Z, Fujimura K, Ji Q (2002a) Real-time eye detection and tracking under various light conditions. In: *Proceedings of the 2002 symposium on Eye tracking research & applications*, ACM, pp 139–144
- Zhu Z, Ji Q, Fujimura K, Lee K (2002b) Combining kalman filtering and mean shift for real time eye tracking under active ir illumination. In: *Object recognition supported by user interaction for service robots*, IEEE, vol 4, pp 318–321