A Machine Learning Approach for a Vision-Based Van-Herick Measurement System

A Machine Learning Approach for a Vision-Based Van-Herick Measurement System

Tommaso Fedullo
Dept. of Management and Engineering
University of Padova
Vicenza, Italy
tommaso.fedullo@phd.unipd.it

Davide Cassanelli
Dept. of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
davide.cassanelli@unimore.it

Giovanni Gibertoni
Dept. of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
giovanni.gibertoni@unimore.it

Federico Tramarin
Dept. of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
federico.tramarin@unimore.it

Luciano Quaranta
Diagnostic and Pediatric Sciences
Section of Ophthalmology
University of Pavia
IRCCS Fondazione Policlinico San Matteo
Pavia, Italy
luciano.quaranta@unipv.it

Giovanni De Angelis
Diagnostic and Pediatric Sciences
Section of Ophthalmology
University of Pavia
IRCCS Fondazione Policlinico San Matteo
Pavia, Italy
giovanni.deangelis@unipv.it

Luigi Rovati
Dept. of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
luigi.rovati@unimore.it

***Abstract*—The application of Artificial Intelligence to the instrumentation and measurements field is nowadays an attractive research area. Indeed, Artificial Intelligence gives the possibility to perform activities also in case of inability to perfectly model a phenomenon or a system. Furthermore, making machines learn from data how to perform an activity, rather than hard code sequential instructions, is a common and effective practice in many modern research areas. This paper investigates the possibility to use Machine Learning techniques in an ophthalmic vision–based system performing automatic Anterior Chamber Angle measurements. Currently, this procedure can be performed only by appropriately trained medical personnel. For this reason, Machine Learning and Vision–Based techniques may greatly improve both test objectiveness and diagnostic accessibility, by allowing to automatically carry out the measurement procedure.**

***Index Terms*—Artificial Intelligence, Machine Learning, CNN, Vision Based Measurement, Van Herick, Computer Vision**

I. Introduction

Closure Glaucoma (Glaucoma and others eye diseases are affecting more and more people in the last years. Prevention is essential to avoid the progression of that disease, but in some cases, the screening exams are invasive or quite expensive, and it is not possible to periodically monitor the eye condition [1]. It has been demonstrated that people with a narrower Anterior Chamber Angle (ACA), i.e. the iridocorneal angle, are more vulnerable to the most aggressive form of glaucoma, the Primary Angle Closure Glaucoma (PACG) [2]. Nowadays, the gold technique used for ACA measurement is the gonioscopy, but it is invasive and requires high medical skills [3]. Among the various assessed techniques, one of the most interesting is the van Herick maneuver, which exploits the correlation between the thickness of the cornea and the ACA [4]. The ratio between these two thicknesses represents the width of the ACA, and so it can be used to detect the PACG. Van Herick approach requires a slit lamp, illuminating the limbus with a 60° angle between the light source and the eye optical axis [5]. Despite the simplicity of the approach, any PACG diagnosis derived from the estimation of the ACA with the van Herick technique must be performed by direct observation of an expert ophthalmologist with the help of a traditional slit lamp and a microscope. Such considerations inevitably make this type of diagnosis intrinsically subjective and results are strictly related to the ability and experience of the observing ophthalmologist [3].

In this paper, we present a new Machine Vision (MV) based approach that can overcome such subjective limitations. This new method relies on a modified optical setup to perform the van Herick measurement in a semi-automated way, with a compact CMOS camera that acquires images of a human eye. A slit light generated with an RGB LED-based digital light projector (DLP) is used in the place of the traditional slit light. The possibility to apply MV techniques to classify the collected images for automatically measure the ACA is further investigated in this paper. Particularly, the presented test case perfectly fits the definition of Vision–Based Measurement (VBM) system, i.e. the application of MV to the instrumentation and measurement field [6]. Machine Learning (ML) techniques have been widely applied to VBM systems [7].

This paper is organized as follows. First, in Section II, we sketch the experimental setup and the main goals of this research activity. Moreover, Section III presents the specific

ML techniques used and the evaluation techniques. Results are then discussed in Section IV and the paper conclusions are reported in Section V.

II. Experimental Setup and Research Goals

The Van Herick procedure used to measure the ACA must be accomplished with specific alignment constraints [3]. It has been shown by Leung et al. [8] that both illumination and observation angles affect the ACA openness assessment. As a consequence, particular attention was paid during the realization of the optical setup for such an experimental evaluation. A schematic diagram of the optical setup used to perform the Van Herick measurement is shown in Fig. 1. The optical setup consists of two main devices. Firstly, a

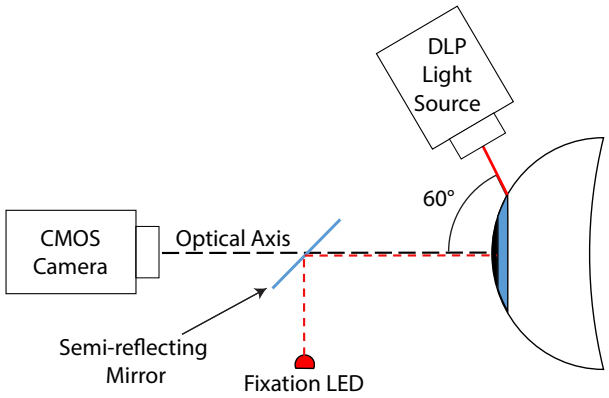


Figure 1. Schematic diagram of the optical setup.

digital CMOS camera Basler Dart (daA1600-60uc S-Mount, Basler© AG, Ahrensburg, Germany) is positioned in front of the analyzed eye, aligned with his optical axis. A 16 mm focal length lens (Evetar Lens M12B1618IRM12 F1.8) is also used in the optical setup. Secondly, a LED DLP (DLP2010EVM-LC, Texas Instruments©, Dallas, Texas, U.S.) was used as the illumination unit, instead of the traditional slit light. The DLP relies on modern micro-mirror technology to project structured light onto a specific target. The illumination unit was placed at a 60 degree rotation angle with respect to the eye and camera optical axis as shown in Fig. 1. Moreover, the DLP has been configured to emit a uniform red slit light that scans the whole surface of the eye under examination. Thus, during the measurement procedure, the emitted slit light scans the eye from the external corner of the sclera toward the nose. This scan takes place during the frames acquisition with the digital CMOS camera. A fixation target, i.e. a small light pointer, was placed on the eye-camera optical axis, through a 45° semi-reflecting mirror, to help the patient to look straight ahead. The patients head is then placed on a chin rest to guarantee the steadiness and alignment to the optical setup. A single measurement procedure has been designed to perform two entire scans of the eye in 4 seconds, allowing for a total of 120 raw pictures to be collected at 30 fps.

As mentioned, the Van Herick technique involves comparing the depth of the peripheral anterior chamber to the thickness of the cornea when a narrow slit of light is shone within the limbus, i.e. the edge between the cornea and the sclera. As a consequence, within the entire set of images acquired during the scan, only a few of these (referred to as *central* images) can be used to measure the ACA. Both previous and subsequent acquisitions, where the light is placed respectively on the left and on the right of the limbus, must be discarded. Indeed, since patient eye position may change between different measurements, an a–priori images selection can not be performed. Consequently, a wider area must be scanned.
As a result of a single measurement, each dataset coming from the acquisition system consists of two scans of the entire eye, each one composed by 60 images with a $1600x1200$ px resolution. As an example, a set of images acquired during a scan is represented in Fig. 2.

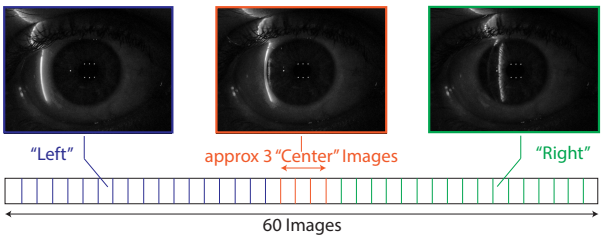


Figure 2. Example of a set of images acquired during a scan.

The approach, we followed in this work, is based on a Convolutional Neural Network (CNN), solving a three–class image classification problem. The CNN takes as inputs the images acquired by the setup and performs the classification activity by tagging the images as *left*, *right* or *central*, respectively labeled with 0, 1, 2. The analysis of the performance of the network is carried out through an objective approach.

III. ML–based classification and Evaluation Technique

Image Classification is a key task in a VBM perspective, and has been addressed for years using ML techniques [9]. In particular, CNNs reviled their suitability for VBM systems in many different applications [10]. For example, authors of [11] proposed a CNN–based system for railway networks inspection. Differently, [12] presented a method for head pose estimation in vehicles, demonstrating high versatility in the application of CNNs. Furthermore, an interesting review paper [13] summarizes the most widespread Deep Learning (DL) techniques for Image Classification, with regards to ophthalmic applications. They identified several CNN network models that proved to be promising such as, among others, AlexNet [14], GoogleNet [15] and resNet [16]. These networks showed the best results in the ImageNet Large Scale Visual Recognition Challenge [17] and are all based on Convolutional Layers.

A. Convolutional Neural Network Design

Different network structures and settings have been tested during an extensive experimental campaign, where a trial and error methodology has been adopted. In particular, several typical parameters, such as precision and recall (both defined

in the following), have been used to optimize the network architecture. Afterward, the best configuration for our application has been chosen and it is presented in this Section. An AlexNet [14] structure has finally been chosen for this application. *Relu* has been adopted as it is the common–used activation function [18], that is in charge to manage the Input–Output behavior of neurons. Moreover, in this situation, it achieves excellent performances and it demonstrated to learn faster. AlexNet structure, as typical CNNs, is formed by both Convolutional and Pooling layers, where the latter are placed between each convolutional layer and the subsequent one. Specific image patterns are identified by means of a Kernel slicing on the entire layer. Afterward, a common practice consists in the usage of a pooling layer, that downsamples the input patterns, aiming to increase the robustness of the network to slight variations of the detected features. The extraction of such meaningful features is done taking the maximum from each kernel acquisition, namely *max pooling*. Finally, the last fully connected layers and the *softmax activation* compute the probability of each image to belong to each of the 3 classes. A slight modification has been made in respect to the typical AlexNet structure: we used a bilinear interpolation algorithm to obtain resized $400x300$ px input images. It is worth noting that downsized images are used only for classification purposes, while the ACA measurement will be performed with native resolution ones. Indeed, there are no network performance improvements when using the typical $227x227$ px AlexNet input size. Likely, an higher quality of the image is, for our purposes, more important than a fine tuning of layers and kernel sizes. Figure 3 shows the network structure we used.

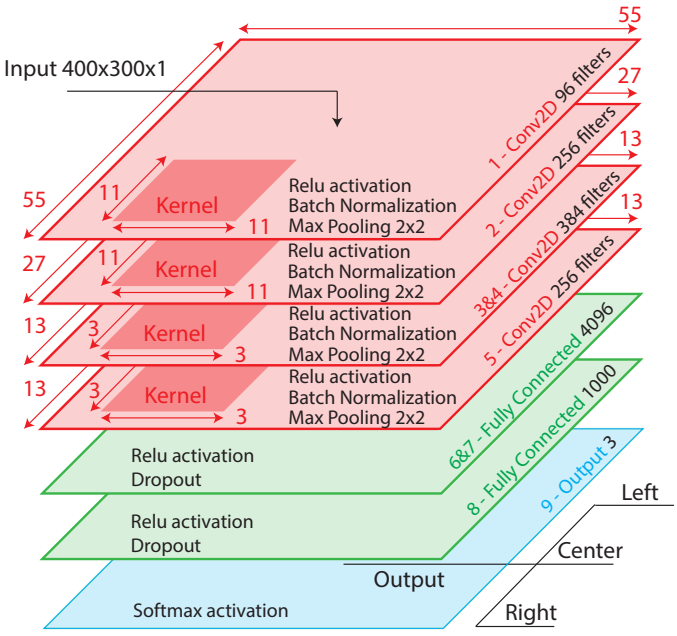


Figure 3. AlexNet structure.

As it is possible to see from Figure 3, the dropout technique, which consists on the deactivation of some randomly chosen neurons, is used to reduce inter–dependency between fully connected layers, thus preventing over–fitting. The Dropout Rate has been set to 50%.

Data preparation is a fundamental task to be done before the training stage. Hundreds of acquisitions were made and the data have been manually split into the three classes to train the network. Data are normalized between 0 and 1, and then 70% of the total have been used to train the network, while the other portion for validation purposes. One important feature of the available dataset is the low number of *central* images (i.e. those ones to be identified with the network), usually three or four out of 120 images acquired during the measurement procedure. To over the hump, *data augmentation techniques* have been adopted. Data augmentation usually foresees the generation of modified training examples, applying different and pre–tuned image transformations, e.g. horizontal and vertical shifts, rotations and brightness modification, only to name a few. This technique is typically used to generalize the behavior of the network, trying to generate new and meaningful examples targeted for the application. Instead, we used data augmentation also to generate more *central* images, this way giving the possibility to train the network with comparable amounts of *left*, *right* and *central* images. In Table I are listed our data augmentation parameters.

Table I
DATA AUGMENTATION SETUP PARAMETERS.

Horizontal Shift Range (% of the width)	$(-1\%, 1\%)$
Vertical Shift Range (% of the height)	$(-1\%, 1\%)$
Rotation Range (deg)	$(-7, 7)$
Brightness Multiplication Factor Range	$(0.5 - 1.75)$
Zoom Range (% of the picture size)	$(90\% - 110\%)$

Indeed, little movements to the right and to the left, rotations and zooms may occur. Furthermore, different brightness levels could take place in different surrounding environments. After the data preparation stage, the prepared data set had the properties listed in Table II. It is worth observing that we generated more central images than the other two classes, aiming to increase the precision of the classification of the *central* images.

Table II
DATA SET PROPERTIES.

	Original	Data Augmentation	Total
Left	8537	5711	14248
Central	593	14994	15587
Right	7877	5471	13348

B. Evaluation Technique

The goodness of the CNN performance is evaluated through the study of the difference in pixels between the position of the limbus and the position of the light line on the eye surface: the lower this difference the higher the CNN accuracy. This

procedure has been divided into 4 steps: i) recognition of the iris from a reference image to have a reference for the intensity profile extraction; ii) extraction of the limbus position from a reference image captured at the first instant of the measurement procedure; iii) evaluation of the line position for every image labeled as *central*; iv) computation of the difference between these two positions.

The extraction of the limbus position is performed by considering the limbus intensity profile, that can be fitted as an error function such as the one in Eq.(1), where A, B, C and D are fitting parameters that change for each reference image. Then, the maximum of the fit function derivative, Eq.(2), is computed to represent the limbus position, X_{limbus}.

$$f(x) = A + B\frac{2}{\sqrt{\pi}}\int_0^{D(x-C)} e^{-t^2}dt \tag{1}$$

$$f'(x) = D \cdot B\frac{2}{\sqrt{\pi}}e^{-D^2(x-C)^2} \tag{2}$$

In distinction, since the light line intensity profile can not be easily fitted by a standard distribution, the X_{line} position of the light line is estimated as in [19] [20]:

$$X_{line} = \frac{\sum_i I_i x_i}{\sum_i I_i}, \tag{3}$$

where x_i is the single–pixel x-coordinates, while I_i is the single pixel intensity. The error between the limbus position X_{limbus} and the line position X_{line} is finally computed as:

$$E = X_{limbus} - X_{line}. \tag{4}$$

The value of E can be positive or negative (i.e. left or right, respectively) based on the light line position with respect to the limbus on the images selected by the neural network.
The error E measurement is evaluated through a statistical analysis with a histogram of the error occurrences, observing its center of gravity and standard deviation.

IV. Results

A. The Neural Network

The network described in Section III-A has been implemented in Python, within the *Keras* (Tensorflow Version 2.1.0) framework and trained according to the set–up described in Table III.

Table III
CNN training setting.

Optimizer	Stochastic Gradient Descend (SGD)
Learning Rate (LR)	10^{-2}
Epochs	75
Batch Number (BN)	128
Loss Function	Sparse Categorical Crossentropy
Validation Data (% of total)	30

At each training stage, the error gradient was calculated through the Stochastic Gradient Descent (SGD) algorithm. By controlling the error gradient the optimizer updated the weights so that the error decreased step by step. The next weights choice was done evaluating the *error cost*, through a specific loss function. Given the training settings of Table III, considering the n–th observation was labeled as a specific class c with probability $p_{n,c}$ and N was the total number of observations, the Sparse Cathegorical Crossentropy loss function was expressed as:

$$Loss(p_{n,c}) = -\frac{1}{N}\sum_{n=1}^{N}[log(p_{n,c})]. \tag{5}$$

In this context, the Learning Rate (LR), that represents how quickly the network learn, was surely a key parameter, that has been chosen after several experimental tests. Weights were updated at each step proportionally to the LR and to the calculated error. Both training and validation data have been divided into BN chunks that were individually loaded in memory to increase computational efficiency. Furthermore, it was important to highlight that the validation data was the 30% of the original images: data augmentation was performed only on the training data. Typically, a first evaluation of the network performances includes the analysis of the accuracy. Accuracy was simply calculated as the percentage of correctly labeled data (y_t) vs. total observations ($y_t + y_f$):

$$accuracy(\%) = \frac{y_t}{y_t + y_f} * 100. \tag{6}$$

The resulting accuracy curve is shown in Fig. 4. It is worth noting that in our test case the train dataset was used 75 times to feed the network (i.e. 75 epochs). The training has been stopped when the loss and accuracy did not increment for several consecutive epochs, to avoid over–fitting.

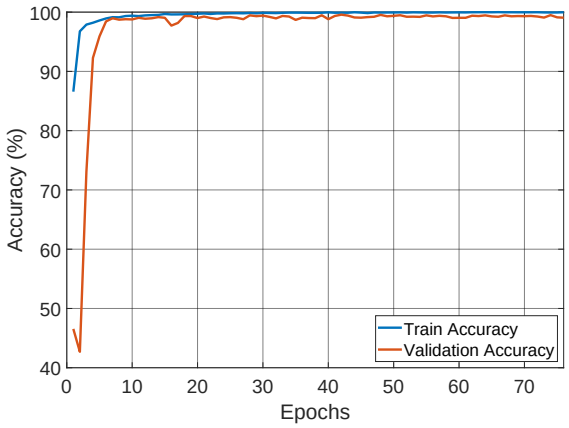


Figure 4. Training and Validation Accuracy

Results are encouraging, since it is possible to achieve high accuracy. Despite this, a more accurate analysis is needed, and a deeper evaluation is now conducted by means of different metrics. Indeed, the objective of the network is to identify images belonging to the *center* class, that has very low validation examples. For this reason, high accuracy values can be obtained also if the *central* class is not well predicted.

Among all the *central* predictions made by the network (Total Positive, TotP), a group of them (True Positive, TP) is well predicted while others are not (False Positive, FP). Moreover, it is possible that the network wrongly labels some *left* or *right* images, as central, namely False Negative (FN). Given these definitions, Precision and Recall for the *central class* are defined respectively as:

pok

$$Precision(\%) = \frac{TP}{TP + FP} * 100, \tag{7}$$

$$Recall(\%) = \frac{TP}{TP + FN} * 100. \tag{8}$$

Recall indicates the network ability to mark all the real *Central* images as *Central*, while precision represents the amount of *left* and *right* images labeled as *central*. It would be important to obtain high values of both recall and precision. Moreover, there is an impact of the threshold value on the precision. The threshold is defined as the probability level that an observation belongs to the *central* class above that the observation is finally marked as *central*.
The resulting recall and precision value obtained are shown in Fig. 5.

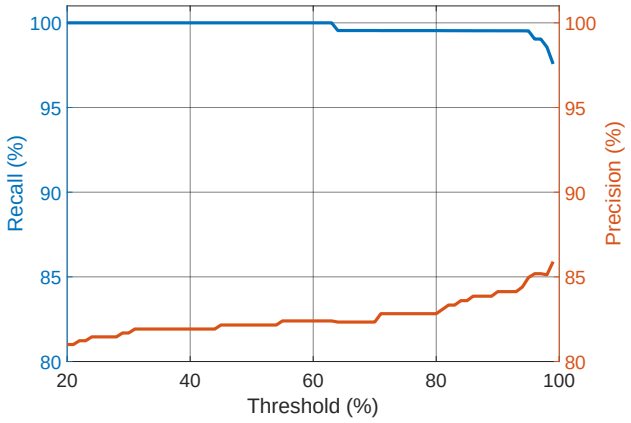


Figure 5. Precision and Recall

It hit the eyes that recall remains constant (at the maximum) for most of the threshold values, while precision grows with the threshold. Indeed, a threshold increase reflects on a low number of total positive, i.e. $TotP = TP + FP$. The growth of the precision is a clear indication that the total positive decrease goes together with a decrease of false positives that mostly become true negatives. Indeed, the recall decrease is very slight and mostly concerns the last part of the curve. This allows to choose an high value of the threshold to increase the precision of the whole network. It is worth observing that all the reasoning is made on the global number of test sequences, but high values of threshold may involve in a small number of total positive for a single data–set. For this reason, in this work the chosen threshold is of 89%, allowing to achieve a precision much greater than the 80%.

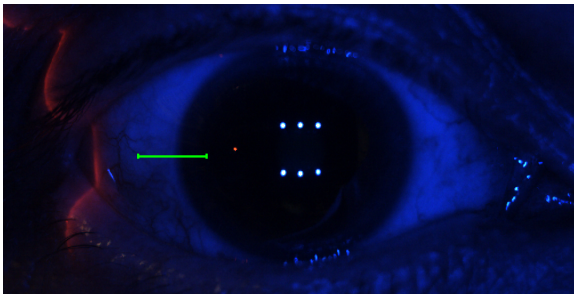
The trained network can be finally used to predict *central* images on newly acquired data set. The classification algorithm can be run in a common–on–the shelves Personal Computer, with additional few seconds of execution that does not increase the acquisition time, as it is performed after the scans are completed. This is surely an important aspect to underline, as it is demonstrated that patients ability to look steady toward the fixation target, is time limited.

B. Test Results

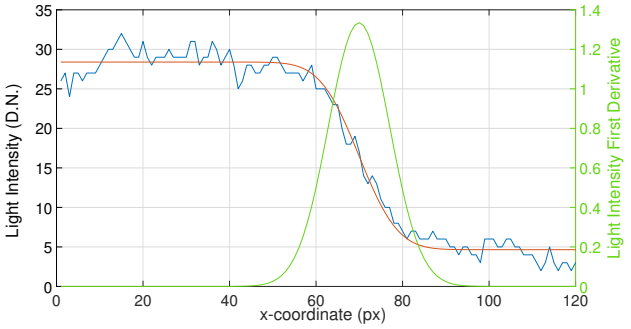
The performance of the network and the instrument were tested by performing two different measurements on volunteers, as from section III-B. The images were collected in grey-scale by extracting the red channel from the original image, to emphasize the light line. In distinction, the reference image was collected in RGB to simplify the iris recognition.
To evaluate the network 6 different subjects have been tested 5 times, for a total of 60 eye scans (i.e. 30 acquisitions). The number of *center* images chosen by the network ranged from 3 to 5 per each acquisition. The total amount of analyzed images was 125. The six subjects on which the measurements were carried out were chosen with different colors of the iris to exclude this variable as a likely possible source of error in the validation phase. In Fig. 6(a) it is possible to observe a typical reference image from which the limbus position is extracted (a reference image is collected for each eye acquisition). In Fig. 6(b) the limbus intensity profile, extracted in correspondence to the eye center y-coordinate, is plotted along with its fitting as an error function and its derivative. The maximum of the derivative represents the limbus position. In Fig. 7 a typical image used to extract light line position is shown. It can be observed that the intensity profile is noisy and irregular. Therefore, the line position is computed as the centroid of that profile. Finally, in Fig. 8, the histogram of the error occurrences is shown. As it is possible to see, the histogram has a Gaussian behavior with parameters: $\mu = 5.99$ px and $\sigma = 5.92$ px.

V. Conclusions and Future Work

The use of Machine Learning techniques to extract measurement information from ophthalmic images is a very promising technique that can overcome some critical aspects of the manual Van Herick approach. The proposed Neural Network provides very good results in the *center* images choice. The presented results shows a mean error $\mu = 5.99$ px with a standard deviation $\sigma = 5.92$ px. The maximum error observed during the experiment, i.e. $E \approx 20$ px, is visually comparable to the line width. After the 'center' images are identified, a vision–based algorithm is currently under development to measure the displacement between the slit light that hits the cornea and the refracted light inside the anterior chamber.
A refined tuning of the vision–Based system will be carried out based on a comprehensive inter–subject experimental campaign that takes the entire Martin-Schulz scale into proper account. Moreover, the ongoing activity is aimed at a careful analysis of the measurement uncertainty.

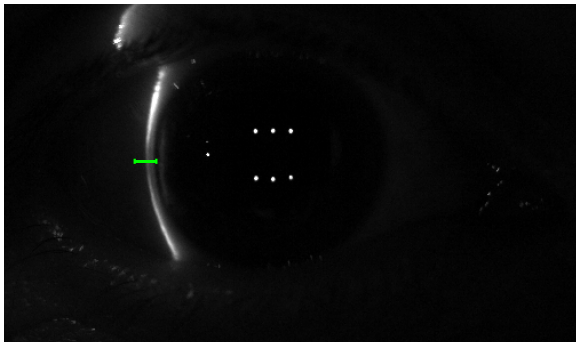


(a)

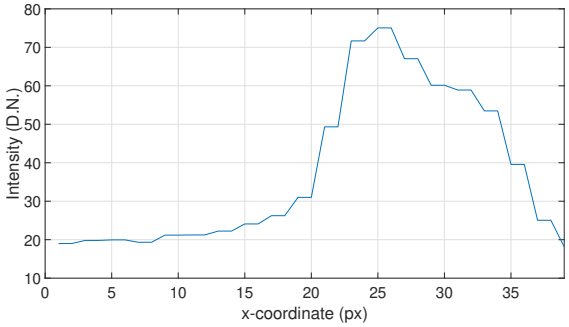


(b)

Figure 6. Typical reference image used to estimate the limbus position (a); the corresponding intensity profile along the y-coordinate of the eye center (•) (green line in (a)), the fitting error function (•) and, its derivative (•) (b).



(a)



(b)

Figure 7. Typical reference image used to estimate the light line position (a); intensity profile extracted along the y-coordinate of the eye center (green line in (a)) (b).

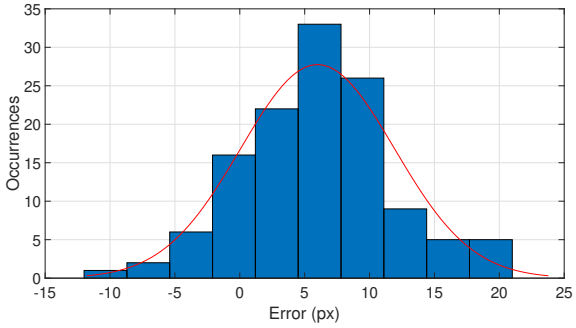


Figure 8. Histogram of occurrences of the error between the limbus and line position in pixel. The (•) line represents the Gaussian fitting of the histogram.

References

[1] Y.-C. Tham, X. Li, T. Y. Wong, H. A. Quigley, T. Aung, and C.-Y. Cheng, "Global prevalence of glaucoma and projections of glaucoma burden through 2040: A systematic review and meta-analysis," *Ophthalmology*, vol. 121, no. 11, pp. 2081–2090, 2014.

[2] A. Javed, M. Loutfi, S. Kaye, and M. Batterbury, "Interobserver reliability when using the Van Herick method to measure anterior chamber depth," *Oman Journal of Ophthalmology*, vol. 10, no. 1, pp. 9–12, Apr. 2017.

[3] I. Riva, E. Micheletti, F. Oddone, C. Bruttini, S. Montescani, G. De Angelis, L. Rovati, R. N. Weinreb, and L. Quaranta, "Anterior chamber angle assessment techniques: A review," *Journal of Clinical Medicine*, vol. 9, no. 12, p. 3814, Nov 2020.

[4] J. Gispets, G. Cardona, M. Verdú, and N. Tomàs, "Sources of variability of the van Herick technique for anterior angle estimation," *Clinical & Experimental Optometry*, vol. 97, no. 2, pp. 147–151, Mar. 2014.

[5] P. L. Dabasia, D. F. Edgar, I. E. Murdoch, and J. G. Lawrenson, "Noncontact Screening Methods for the Detection of Narrow Anterior Chamber Angles," *Investigative Ophthalmology & Visual Science*, vol. 56, no. 6, pp. 3929–3935, Jun. 2015.

[6] E. S. dos Santos, W. B. Xavier, R. N. Rodrigues, S. S. da C. Botelho, and A. V. Werhli, "Vision based measurement applied to industrial instrumentation," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 788 – 793, 2017, 20th IFAC World Congress.

[7] M. Khanafer and S. Shirmohammadi, "Applied ai in instrumentation and measurement: The deep learning revolution," *IEEE Instrumentation Measurement Magazine*, vol. 23, no. 6, pp. 10–17, Oct 2020.

[8] M. Leung, S. S. O. Kang, J. Turuwhenua, and R. Jacobs, "Effects of illumination and observation angle on the van Herick procedure," *Clinical and Experimental Optometry*, vol. 95, no. 1, pp. 72–77, Jan 2012.

[9] I. goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[10] G. Lee and H. Fujita, *Deep learning in medical image analysis: challenges and applications*. Springer, 2020, vol. 1213.

[11] J. Zhong, Z. Liu, Z. Han, Y. Han, and W. Zhang, "A cnn-based defect inspection method for catenary split pins in high-speed railway," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 8, pp. 2849–2860, Aug 2019.

[12] M. Venturelli, G. Borghi, R. Vezzani, and R. Cucchiara, "Deep Head Pose Estimation from Depth Data for In-Car Automotive Applications," in *Understanding Human Activities Through 3D Sensors*, ser. Lecture Notes in Computer Science, H. Wannous, P. Pala, M. Daoudi, and F. Flórez-Revuelta, Eds. Cham: Springer International Publishing, 2018, pp. 74–85.

[13] Y. Tong, W. Lu, Y. Yu, and Y. Shen, "Application of machine learning in ophthalmic imaging modalities," *Eye and Vision*, vol. 7, 2020. [Online]. Available: https://doi.org/10.1186/s40662-020-00183-6

[14] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.
[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
[18] Y. Wang, Y. Li, Y. Song, and X. Rong, “The influence of the activation function in a convolution neural network model of facial expression recognition,” *Applied Sciences*, vol. 10, no. 5, p. 1897, 2020.
[19] B. H. Pui, B. Hayes-Gill, M. Clark, M. Somekh, C. See, J.-F. Piéri, S. Morgan, and A. Ng, “The Design and Characterisation of an Optical VLSI Processor for Real Time Centroid Detection,” *Analog Integrated Circuits and Signal Processing*, vol. 32, pp. 67–75, Jul. 2002.
[20] K. K. Chavali, P. M. Furth, and P. R. Surkanti, “A low-voltage, adaptive CMOS centroid image sensor with improved bandwidth,” in *2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2014, pp. 937–940, iSSN: 1558-3899.