

Received May 30, 2021, accepted July 7, 2021, date of publication July 19, 2021, date of current version July 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3098032

A Statistical Robust Glaucoma Detection Framework Combining Retinex, CNN, and DOE Using Fundus Images

WHEYMING TINA SONG¹, ING-CHOU LAI²,
AND YI-ZHU SU¹

¹Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 30013, Taiwan

²Department of Ophthalmology, Chiayi Chang Gung Memorial Hospital, Puzi City 61363, Taiwan

Corresponding author: Wheyming Tina Song (wheymingsong@gmail.com)

This work was supported in part by the National Science Council under Grant MOST-109-2221-E-007-070.

ABSTRACT Motivated by the challenge that manual glaucoma detection is costly and time consuming, and that existing automated glaucoma detection processes lack either good performance or any statistical robustness testing procedures, we proposed an effective, robust, and automated framework for glaucoma detection based on fundus images. The proposed framework using 1450 color fundus images provided by Kaohsiung Chang Gung (KCG) Memorial Hospital in Taiwan. The proposed framework combines the use of convolutional neural networks (CNN) with the proposed generalized loss function, robust design of experiment (DOE), and Retinex theory to improve the results of fundus photography flash by restoring the original colors via removing the light effect. The proposed framework outperformed most archival automatic glaucoma detection approaches in its effectiveness and simplicity. The effectiveness was demonstrated via the estimated sensitivity 0.95, specificity 0.98, and accuracy 0.97. The simplicity was shown via the adopted basic CNN model compared to deep CNNs such as GoogleLeNet and ResNet152. Further, the proposed framework outperformed all relevant archival work in terms of its robustness, illustrated in the associated standard errors (all less than 0.03). This paper demonstrated the proposed framework via intuitive graphs and clear mathematical notations to make it easy for others to reproduce our results. The proposed framework and demonstration have the potential to become the standard automated glaucoma detection approaches in practice.

INDEX TERMS Glaucoma, CNN, DOE, robust, Retinex.

I. INTRODUCTION

Glaucoma is an eye disease that is notoriously known to be incurable, treatable with medical and surgical procedures that can only delay the aggravation of glaucoma but not restore eye health. Glaucoma is considered to be one of major causes of irreversible-blindness worldwide.

At present, existing glaucoma detection processes do not consistently provide satisfactory results. Manual glaucoma detection (i.e., assessment of the optic nerve head) is subjective, costly, time consuming, and that the variations among physician performances are high.

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Ali.

Our review of prior research found that a number of automated glaucoma detection processes using color fundus images have been proposed. Beginning in 2006, earlier works include [3], [14], [18], [2], [25], [5], [10], [15], [28], [24], [19], and [27]. Since 2018, more recent investigations include [7], [26], [8], [29], [9], [30], [32], and [4].

All of the above-cited automated glaucoma detection method studies use color fundus images as input data and adopt a common framework that includes two major steps: (1) data transformation, which converts the input data (color fundus images) into so-called transformed data; and (2) data classification, which categorizes the binary output (glaucoma or healthy eye) as a function of the transformed data.

We proposed an effective automatic classification framework that integrates Retinex transformation and

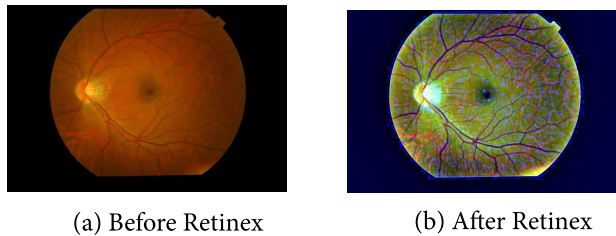


FIGURE 1. Image before and after Retinex process.

convolutional neural network (CNN) deep learning method with the proposed new loss function and robust hyper-parameters obtained via Design of experiments (DOE). Specifically, the contributions of the paper are listed below:

- 1) Effectiveness: The resulting estimated performance (sensitivity 0.95, specificity 0.98, and accuracy 0.97) are all above 0.95.
- 2) Robustness: The associated standard errors are all less than 0.03 for the estimated performance.
- 3) Creativity (To the best of our knowledge, no existing glaucoma detection literature stated the following methods):
 - a) A new loss function, shown in Eq. (14), was proposed.
 - b) Retinex theory (to improve the results of fundus photography flash by restoring the original colors via removing the light effect) was adopted.
 - c) A robust design of experiments (DOE) to generate optimal hyper-parameters (including input image size, convolution layers, and hidden layers) was adopted.
 - d) All reported digits of the estimated performance were determined through the leading digit rule (LDR [33]).
- 4) Simplicity: The basic CNN model (with 4 convolutional layers) was adopted instead of choosing deep CNNs such as GoogleLeNet (with 21 convolutional layers) or ResNet152 (with 152 convolutional layers).
- 5) Clarity: Many intuitive graphs and clear mathematical notations were proposed to make it easy for others to reproduce our results.

The remainder of this paper is organized as follows. Section II provides a literature review with expansions. Section III presents the methodology combining Retinex theory and CNN with the optimal hyper-parameters using DOE. A discussion of the associated performance is provided in Section IV. The summary and conclusion are given in Section V.

II. LITERATURE REVIEW

Twenty archived publications addressing automated glaucoma classification are summarized in Table 1, with the top row indicating the column (C) number. Significant factors (such as Retinex, ROI*, robust DOE, and the proposed new

loss function for the CNN) proposed in this study are marked in red in Table 1. C1-C6 are defined as follows: C1 is the paper number. C2 identifies the paper's referenced index, with the publication year shown in C3. C4 shows the original-data figures indicating glaucoma (G) and normal (N), while C5 indicating whether augmented data are adopted. Note that augmented data (rather than original data) are finally used to be classified. C6 lists the adapted data-ratio for the training, validation, and test sets. Figures appear as a hyphen in C5 and C6 if the associated information are not clearly shown in the cited paper. All images, except those used in [9], are independent. Images adopted in [9] are sequential images

The results of our current research are shown at the bottom of Table 1. The explanation of corresponding C4-C6 are given below. This study investigated original data for 899 glaucoma (G) cases and 551 normal (N) cases. After augmentation (see Step 4 of Section III-B), the data size was enlarged twice. C-6 shows a data-ratio of 5:2:3.

Data transformation methods are listed in C7. To conserve space, only an abbreviation is listed for each associated transformation method, including ROI, PCA, EF, IC, HF, GCM, EWT, and ROI*. Specifically, ROI (region of interest) indicates the area covering around optic nerve cup and optic nerve disc. ROI* adopted in the proposed method covers the optic nerve cup, optic nerve disc, and macula. Other transformation methods are briefly described below. PCA (principal component analysis [1] and [17]) transforms the data via the logic of reducing the dimensionality of a set of variables while retaining the maximum variability in terms of the variance-covariance structure. Texture feature [2] provides measures of properties, such as smoothness, coarseness, and regularity of the images. Higher order spectra (HOS) [2] elicited both amplitude and phase information of a given signal. EF (ellipse fitting [6]) fits an elliptical model of data into the area identified in ROI. IC (illumination correction [5]) uses anisotropic diffusion filtering to remove noise, further subtracting the estimated background from the original color image to achieve homogeneity. HF (Haralick features [13]) here refers to 28 statistical features (including the first and second moment, correlation, and entropy). EWT (empirical wavelet transform [11]) transforms time-domain data into frequency domain data. GCM (grid color moment) transforms pixels into statistical moments such as mean, variance, and skewness. Retinex [21], used to remove the effects of fundus photography flash and to restore the original colors of the fundus image, will be further explained in Section III-B.

Data classifiers, shown in C8, include ANFIS (adaptive neuro-fuzzy inference systems), ANN with the associated optimizer BP (back propagation [2]), SVM (supporting vector machine [3]), k-means clustering [16], RF (denotes for random forest [23]), RT (denotes for random tree [22]), CNN (denotes for convolutional neural network [20]) which integrates both feature extraction and classification. The loss function, such as mean squared error (mse) or cross entropy (CrossE), adopted in CNN is listed inside the parenthesis.

TABLE 1. Literature review.

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
No	Ref.	Year	Original Data (G,H)	Aug. Data	Ratio (train: validation: test)	Data Transformation	Classifier (Loss Fun.)	Robust DOE	Sen. (σ^*)	Spe. (σ^*)	Acc. (σ^*)
1	[14]	2006	(135, 206)	✓	230:0:111	ROI	ANFIS	–	0.8	0.9	0.86
2	[3]	2007	(100, 100)	–	1:0:1	ROI, PCA	SVM	–	–	–	0.88
3	[2]	2011	–	–	–	Texture HOS	SVM, RF	–	–	–	0.91
4	[18]	2013	(50, 50)	–	–	ROI	k-means	–	–	–	0.94
5	[25]	2013	(20,20)	–	1:0:1	EF, IC	ANFIS	–	0.97	0.96	0.97
							ANN	–	0.97	0.96	0.97
							SVM	–	0.99	0.98	0.98
6	[5]	2015	(168, 482)	✓	99:0:551	–	CNN	–	–	–	0.83
7	[28]	2015	(200, 255)	–	409:23:23	HF	ANN	–	0.99	0.90	0.96
8	[10]	2015	(51, 118)	–	135:17:17	GCM	ANN	–	0.87	0.87	0.87
9	[15]	2015	(32, 35)	–	50:0:17	ROI	SVM	–	1	0.90	0.94
				–	50:0:17	ROI	ANN (mse)	–	0.78	1	0.90
10	[24]	2016	(30, 30)	–	–	EWT	SVM	–	1	0.96	0.98
11	[19]	2017	(297, 202)	–	319:80:100	PCA	RF	–	0.98	0.97	0.98
12	[27]	2017	(15, 15)	–	9:0:1	ROI	RT	–	0.82	0.99	0.96
13	[26]	2018	(837, 589)	–	998:0:428	–	CNN (CrossE)	–	0.98	0.98	0.98
14	[7]	2018	(214, 2112)	✓	650:0:1676	–	CNN (CrossE)	–	0.85	0.84	–
15	[8]	2019	(200, 255)	✓	364:0:91	–	CNN (CrossE)	–	0.91	0.93	0.92
16	[29]	2019	(27,18)	✓	1662:0:45	–	D-CNN (GLN)	–	–	0.93	0.76
			(70, 31)	✓	1606:0:101	–	D-CNN (RN152)	–	–	0.74	0.63
			(194, 261)	✓	1252:0:455	–	D-CNN (RN152)	–	–	0.80	0.77
			(101, 300)	✓	1306:0:401	–	D-CNN (RN152)	–	–	0.95	0.74
			(396, 309)	✓	1002:0:705	–	D-CNN (GLN)	–	–	0.87	0.65
17	[30]	2020	(756, 786)	✓	1078:0:464	ROI	D-CNN (RN152)	–	0.93	0.85	0.89
18	[32]	2020	(396, 309)	✓	50:0:51	ROI	CNN+SVM	–	–	–	0.92
19	[9]	2021	(379, 316) seq.	✓	85:5:10	ROI	CNN+RNN	–	0.95	0.96	–
20	[4]	2021	(231, 325)	✓	56:14:30	3D	D-CNN (VGG16)	–	1	0.93	0.96
21	This Paper	2021	(899, 551)	(1798, 1102)	5:2:3	Retinex, ROI*	CNN (New Loss)	✓	0.95 (0.03)	0.98 (0.03)	0.97 (0.01)

Authors [7] claimed that the advantage of CNN is that it does not need any pre-processing or handcrafted feature extraction by other methods.

More deep CNN (D-CNN) models include DENet (denotes for Denver group chromosomal based on CNN [14]), GLN (denotes for GoogLeNet [35]), VGG (denotes for VGGNet [31]), RN152 (denotes for ResNet152, [36]), and RNN (denotes for recurrent Neural Network [37]). Unlike CNN, GLN, VGG, and RN152, RNN is designed for sequential data instead of independent data. For example, images adopted in [9] are sequential images, therefore a combination of CNN and RNN performs better than CNN alone.

The above-mentioned deep CNN models have been commonly adopted to use in automatic glaucoma detection since 2019. For example, [29] proposed deep CNN architectures including GoogleLeNet (with 21 convolutional layers) and ResNet152 (with 152 convolutional layers). [29] proposed to train and test 5 different data sets by employing four of them during training and the other during testing.

For example, the first row corresponding to No 16 (in Column 1) used no. of glaucoma and healthy images ($G=27, H=18$) and total train data 1662 ($= 70 + 31 + 194 + 261 + 101 + 300 + 396 + 309$, the sum of the rest four data sets) and testing data 45 ($= 27 + 18$). All of the proposed 5 cases corresponding to No 16 in Table 1 are inferior than the model proposed in this paper.

The last 4 columns (C9, . . . , C12) are categories designated as follows. With regard to C9, none of the studies in the 20 cited publications appear to have used statistical robust design of experiments (DOE) (see Section III-F) via blocking (see Section III-A) to determine the optimal hyper-parameters. C10, C11, and C12 indicate data for three performances: sensitivity, specificity, and accuracy, respectively. The average sensitivity, average specificity, and average accuracy and the associated standard error for the 10 block settings adopted in this paper are listed in the last two lines. The average performances are all above 0.95 and the corresponding standard errors are all

below 0.03. Further explanation about the proposed method is given in Section III. Again, figures appear as a hyphen if the associated information was not clearly shown in the cited paper.

Finally, the comparison of the performance with respect to sensitivity (shown in C10), specificity (shown in C11), and accuracy (shown in C12) is summarized below.

- Results show that the proposed framework provided better or comparable ($< 5\%$ worse) results compared with 20 archival work listed in Table 1. Specifically, results in bold indicated that our model produced worse results, but the differences are less than 5%, 2%, 1% with respect to sensitivity, specificity, and accuracy, respectively. Results in bold are listed below.
 - Sensitivity: [25], [28], [19], and [9].
 - Specificity: [25], [27], and [26].
 - Accuracy: [19], [24], [25], and [26].
- The proposed estimated performances have standard error less than 0.03, while the standard errors for archival work listed in Table 1 were not stated.

Overall, the proposed framework, combining a CNN (with 4 convolutional layers) with a pre-processing via the Retinex with the proposed parameters (see Eq. (1) to Eq. (10)) presented promising results for automatic glaucoma detection.

III. PROPOSED METHODOLOGY

The framework of the proposed method is illustrated in Fig. 2, in which there are three main parts. We first explain Part 1 (marked in light purple) and 3 (marked in blue). Part 1 confirms that 1450 fundus images (899 showing glaucoma and 551 healthy) were provided by Kaohsiung Chang Gung Memorial (KCGM) Hospital, Taiwan. The percentage adopted for the training, validation, and test sets were 50%, 20%, and 30%.

Part 3 indicates that the performance measures adopted for testing data were accuracy, sensitivity, and specificity, defined below.

- sensitivity: TP (true positive) rate, the ratio of cases correctly identified as positive to all glaucoma cases.
- specificity: TN (true negative) rate, the ratio of cases correctly identified as negative to all normal cases.
- accuracy: the ratio of correct classification to the total cases.

Next, we explain Part 2 (mark in gray) with regard to four issues: (1) the relationship among block, epoch, and batch, (2) the pre-CNN process, (3) the CNN process, and (4) the robust design of experiment (DOE). Further explanation is given in Subsections III-A through III-F.

A. BLOCK, EPOCH, AND BATCH

First, we discuss here the relationship between blocks and epochs. The proposed framework contains l blocks, where $l = 10$. The term “block” was adopted from the experimental design point of view in that the data set (in terms of the training, validation, and test sets) is identical in each block. Block i contains h_i epochs, where $h_i \leq h, i = 1, 2, \dots, l$,

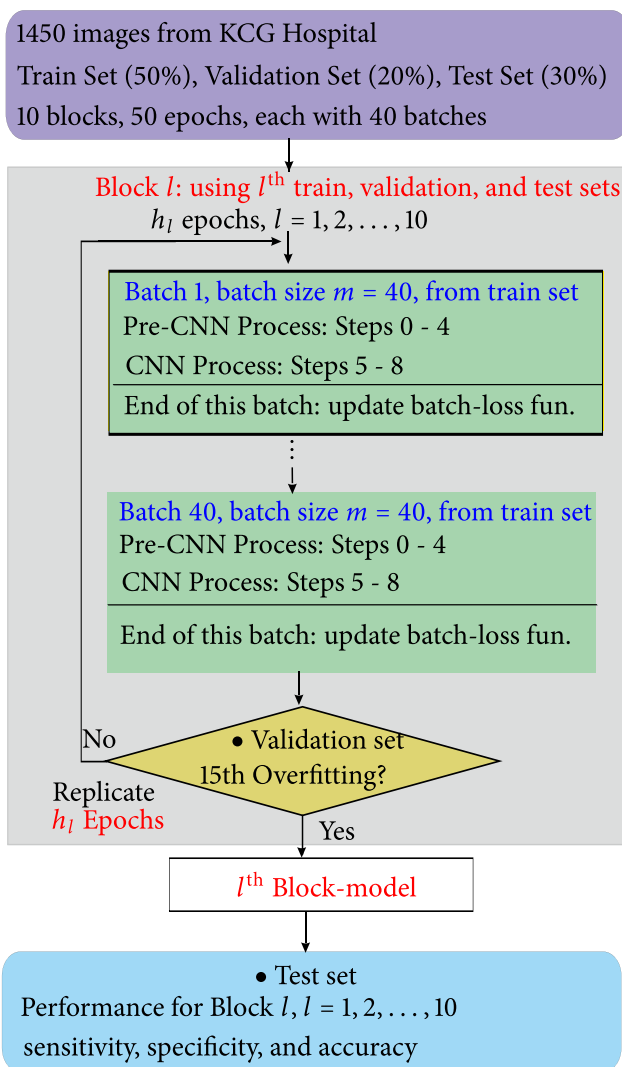


FIGURE 2. Framework-1 of the Proposed Methodology.

where we chose $h = 50$ to be the maximum value of the total epochs in each block. If the termination rule was not reached (see Section III-D), then $h_i = h$. After the final epoch was completed for each block, we constructed a block-model to be tested using the test set to generate the performance measures, shown in Part 3 of Fig. 2.

Next, we assessed the relationship between epochs and batches. Each epoch passed through $b = 40$ batches, each with batch size (images) $m = 40$. The $m = 40$ images in each batch were randomly selected from the training set in the associated block. Each image in each batch passed through the pre-CNN-process (Steps 0 to 4, as shown in Fig. 3. and CNN (Steps 5 to 8, as shown in Fig. 4. At the end of each batch, the loss function and therefore the associated parameters were updated. A pre-validation model was derived from 40 batches of 40 images. Pre-CNN-process and CNN-process are further discussed in Subsections III-B and III-C, respectively.

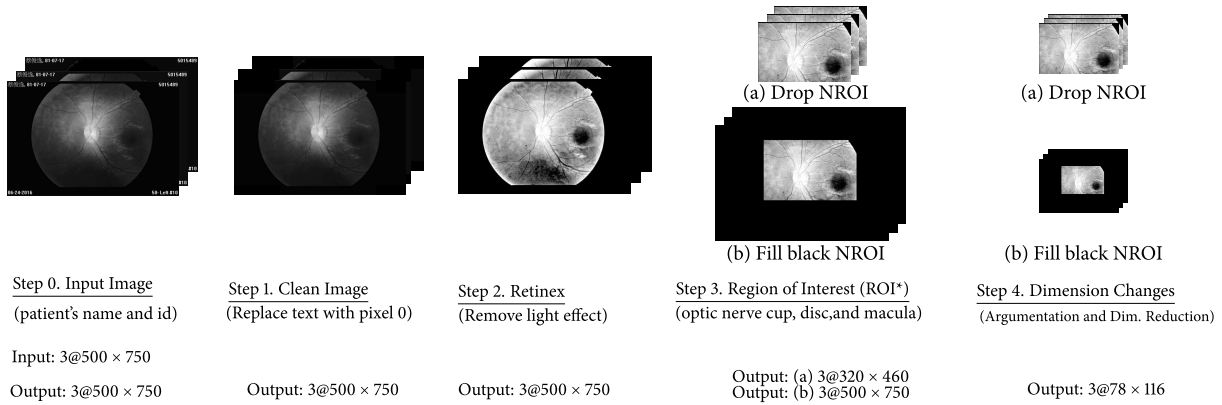


FIGURE 3. Pre-CNN process (Steps 0 - 4).

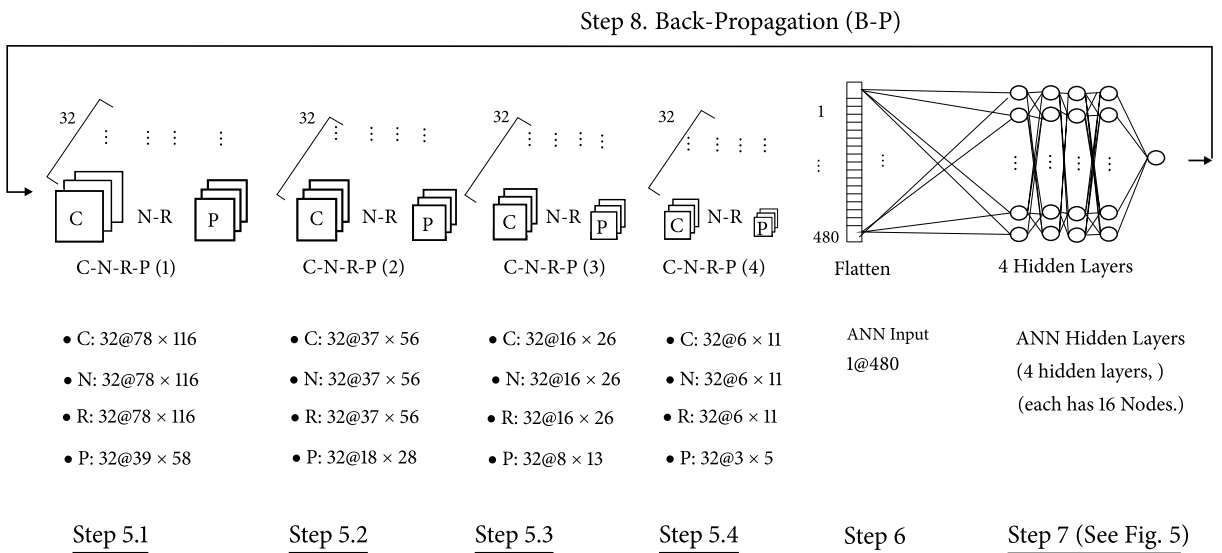


FIGURE 4. CNN framework (Steps 5 - 8).

B. PRE-CNN-PROCESS

This subsection describes the pre-CNN-process, including Steps 0 to 4, illustrated in Fig. 3. The values followed by “input and output” are the number of images with respect to R-G-B, the number of row pixels, and the number of column pixels. For example, the values 3@500 × 750 in step 0 indicates that there are 3 images regarding R-G-B, 500 row pixels, and 750 column pixels, Further descriptions of Steps 0 to 4 are given below.

- Step 0. Input Images: All input fundus images include text about patient information and the date that the associated images were collected.
- Step 1. Clean Images: All text on each image was replaced by pixel 0, which is the color black.
- Step 2. Retinex: The word Retinex combines retina and cortex, where the retina is the part of the eye that detects color, and the visual cortex is the part of a brain that processes the information it receives from the retina.

This step was used to remove the light effect in a flash image. The logic of removing the light effect is explained below.

Suppose that an original image recorded by a camera is denoted as I , which is a function of the actual color (denoted as R), light conditions (denoted as L), and other camera settings. Assuming that we can ignore the camera settings, we can therefore write the relationship among I , R and L as follows:

$$I^{(i)}(x, y) = R^{(i)}(x, y)L(x, y), \quad i = 1, 2, 3. (R, G, B) \quad (1)$$

or

$$R^{(i)}(x, y) = I^{(i)}(x, y)/L(x, y). \quad (2)$$

To prevent the effect of a small value in the denominator $L(x, y)$ leading to a huge $R^{(i)}(x, y)$, we performed a logarithm function on both sides of Eq. (2). Thus,

$$\log(R^{(i)}(x, y)) = \log(I^{(i)}(x, y)/L(x, y)). \quad (3)$$

Let the estimated image of $R^{(i)}(x, y)$ be denoted as $R_{III}^{(i)}$, which is a transformation of $R_I^{(i)}, R_{II}^{(i)}$. Below, we explain $R_I^{(i)}, R_{II}^{(i)}$, and $R_{III}^{(i)}$.

- Function $R_I^{(i)}$ is used to remove the light effect:

$$R_I^{(i)} = \omega \left[\frac{I^{(i)}(x, y)}{I_A^{(i)}(x, y) \otimes G^{(i)}(g_1, g_2)} \right], \quad (4)$$

where we chose $\omega = 1/3$,

$$I_A^{(i)}(x, y) = \begin{cases} I^{(i)}(x-1, y-1), & x = 2, 3, \dots, 501; \\ & y = 2, 3, \dots, 751; \\ 0, & x = 1, 502. \end{cases} \quad (5)$$

The image $I_A^{(i)}(x, y)$ with $\dim 502 \times 752$ is the augmentation of $I^{(i)}(x, y)$. The Gaussian blur (smoothing) function, used to reduce image noise and unimportant detail, is defined as

$$G^{(i)}(l, m) = k \exp\left(-\frac{(l^2 + m^2)}{c_i^2}\right), \quad (6)$$

where we chose $l = -1, 0, 1; m = -1, 0, 1; (c_1, c_2, c_3) = (15, 80, 250)$, and the value k is determined to guarantee $\sum_{l=-1}^1 \sum_{m=-1}^1 G^{(i)}(l, m) = 1$.

- Function $R_{II}^{(i)}$ is used to optimize the R-G-B color proportion, such that

$$R_{II}^{(i)}(x, y) = H^{(i)}(x, y) \times R_I^{(i)}(x, y) + \text{bias}, \quad (7)$$

where

$$H^{(i)}(x, y) = \beta \log \left[\frac{\alpha I^{(i)}(x, y)}{\sum_{i=1}^3 I^{(i)}(x, y)} \right], \quad (8)$$

and bias, α , and β are the color-control parameters. We chose $\text{bias} = 125, \alpha = 125, \beta = 230$.

- Function $R_{III}^{(i)}$ is used to increase the variability of pixel in $R_{II}^{(i)}$, provided that all pixels are still between (0, 255).

$$R_{III}^{(i)} = \frac{255 \times (R_{II}^{(i)} - \text{Min } R_{II}^{(i)})}{\text{Max } R_{II}^{(i)} - \text{Min } R_{II}^{(i)}}, \quad (9)$$

where $\text{Max } R^{(i)}$ and $\text{Min } R^{(i)}$ are the maximum and minimum value for the associated matrix $R^{(i)}$, respectively.

- Step 3. Region of Interest (ROI*):

Based on the knowledge of this paper's second author, a well-respected ophthalmologist in Taiwan, we recognized that the region of interest should be near the optic-nerve, cup, and macular portion. Therefore, we selected the (ROI*) as follows.

$$I_{ROI}^{(i)}(x, y) = \begin{cases} R_{III}^{(i)}(x, y), & x \in \text{ROI}^* \\ 0, & \text{otherwise,} \end{cases} \quad (10)$$

where ROI* covers the optic nerve cup, optic disc, and macula.

- Step 4. Image augmentation and dimension reduction: For each input image, we randomly randomly twisting the image among 30, 60, 90, 120, or 180 counter-clockwise. Thus, each image was augmented into twice images. Finally, we reduced each image, shifting the row and column sizes 500×750 into smaller sizes 78×116 . Let $D(i)$ denote the image processed at the end of this step, where $i = 1, 2, 3$ regarding R-G-B.

C. CNN PROCESS

This subsection describes the CNN process, including Steps 5 through 8, which were used to train the optimal parameters.

In order to implement a data transformation and data classification method, one needs to first determine the associated values of parameters and hyper-parameters. Using CNN as an example, it is necessary to begin by determining the parameter values (such as weights in each hidden layer) and hyper-parameter values (such as convolution layer, hidden layer numbers, and numbers of nodes in each hidden layer). The rules and the values of hyper-parameters used in this paper are summarized below.

Rules determined in advance:

- 1) loss function: cross-entropy
- 2) padding method: same (instead of valid)
- 3) initiation weights: Glorot_normal
- 4) optimizer: Adam.
- 5) activation functions for convolution and ANN hidden layers: Relu
- 6) activation function for ANN output layer: sigmoid

Hyper-parameters determined in advance:

- 1) termination rule: if overfitting more than 15 times, stop the replication for epoch.
- 2) kernel matrix: 3×3
- 3) number of blocks: 10
- 4) number of epochs: 50
- 5) number of batches: 40
- 6) batch sizes: 40

Hyper-parameters determined via robust DOE, see Subsection III-F:

- 1) A. sizes used in images: 78×116
- 2) B. convolutional layers: 4, each with 4 transformations (C-N-R-P)
- 3) C. number of kernels in each convolutional layer: 32
- 4) D. NN-hidden layers: 4, each with 3 transformations (C-N-R)
- 5) E. number of nodes in each ANN-hidden layer: 16
- 6) F. treatment for the area outside the ROI: delete (rather than filling with black)
- 7) G. parameters α, β used in the proposed loss function: 1, 1.5

After all hyper-parameters were set, we initiated the CNN architecture, illustrated in Steps 5 through 8, below.

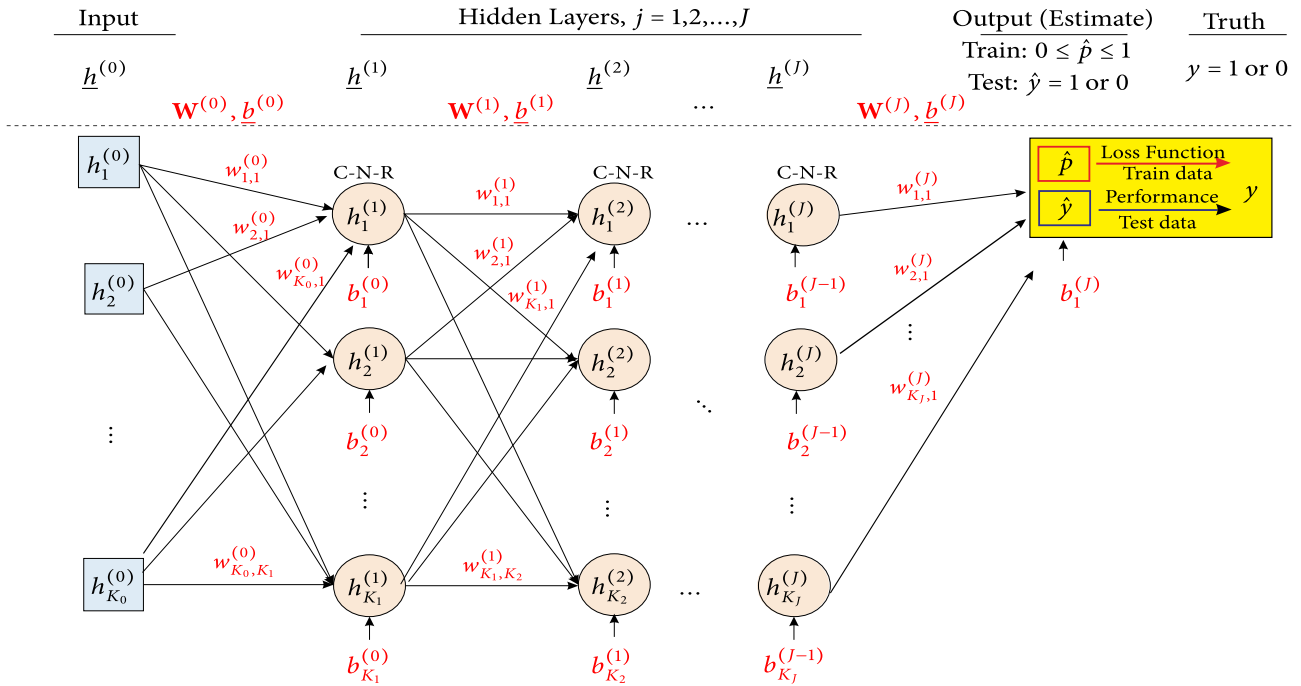


FIGURE 5. Artificial Neural Network (ANN) framework.

- Step 5. CNN with four processes: 1) Convolution (C), 2) Normalization (N), 3) Relu (R), and 4) Pooling (P) are defined below:

- 1) Convolution(C) used to extract image feature:

$$O_C^{(l,m)}(x, y) = \begin{cases} \sum_{i=1}^3 D(i) \otimes C^{(1,m)}(i), & l = 1 \\ O_{C-N-R-P}^{(l-1,m)} \otimes C^{(l,m)}, & l = 2, 3, \dots, L \end{cases}$$

where $D(i)$ is the image processed at the end of Step 4 in pre-CNN process, $C^{(1,m)}(i)$, $i = 1, 2, 3$; $m = 1, 2, 3, \dots, M$ are different kernel matrix (we chose the matrix size 3×3) used in the 1st convolution, In this research, we chose $M = 32$, $L = 4$. The notation \otimes in “A \otimes B” denotes the convolution filter, which reads all of the pixels of the initial image (A) in the action area (3×3), then multiplies the value of each of them by the corresponding value of the kernel matrix (B), finally adds the multiplication results. The operation \otimes replaces the initial pixel in A with the added result.

- 2) Normalization (N) used to balance the scale:

$$O_{C-N}^{(l,m)}(x, y) = \beta_0 + \beta_1 \left[(O_C^{(l,m)}(x, y) - \mu(x, y)) / \sqrt{\sigma^2(x, y)} \right]$$

where $\mu(x, y)$ and $\sigma^2(x, y)$ are the expectative value and variance for all of the images in the same batch of the train set at position (x, y) . The values

β_0 and β_1 are parameters that needed to be trained.

- 3) Relu (R) used to avoid the gradient vanishing:

$$O_{C-N-R}^{(l,m)}(x, y) = \begin{cases} 255, & O_{C-N}^{(l,m)}(x, y) \geq 255 \\ 0, & O_{C-N}^{(l,m)}(x, y) < 0 \\ O_C^{(l,m)}(x, y), & \text{otherwise;} \end{cases}$$

$l = 2, 3, \dots, L, m = 1, 2, \dots, M.$

- 4) Pooling (P) used to reduce the problem size:

$$O_{C-N-R-P}^{(l,m)} = \text{Pooling}(O_{C-N-R}^{(l,m)}, \max).$$

- Step 6. Flatten, used to induce the ANN input vector: Let $\underline{h}^{(0)}$ be the vector containing the flattening data with size $3 \times 5 \times 32 = 480$.
- Step 7. ANN, used to train all parameters for hidden layers and output layer (see Fig. 5.) The j th hidden layer $\underline{h}^{(j)}$ is computed via three transformations: convolution (C), normalization (N), and Relu (R), where the notation C-N-R is similar to that used in Step 5. The true response y and its estimates \hat{p} and \hat{y} , (used for the train and test set data, respectively) are shown at the top right of Fig. 5. Notations used in Step 7 will be further explained below.

- Convolution (C) used to extract image feature: Let $c_k^{(j)}$ denotes the linear sum (i.e, convolution) of $\underline{h}^{(j-1)}$, $\mathbf{W}^{(j-1)}$, and $\underline{b}^{(j-1)}$, where $\underline{h}^{(j-1)}$ is a column vector containing $h_k^{(j-1)}$; $\mathbf{W}^{(j-1)}$ is a matrix with the k th column vector, denoted as $\underline{w}_k^{(j-1)}$,

containing $w_{i,k}^{(j-1)}, i = 1, 2, \dots, K_{j-1}; \underline{b}^{(j-1)}$ is a column vector containing $b_k^{(j-1)}$. That is,

$$c_k^{(j)} = \sum_{i=1}^{K_{j-1}} h_i^{(j-1)} w_{i,k}^{(j-1)} + b_k^{(j-1)},$$

$$j = 1, 2, 3, \dots, J + 1; k = 1, 2, \dots, K_j; \quad (11)$$

where $K_0 = 480$ (the data size after flatten in Step 6), hidden layers $J = 4$, number of nodes in each hidden layer $K_j = 16, j = 1, 2, \dots, J$, and $K_{J+1} = 1$ indicating one node in the output layer.

- Normalization (N) used to balance the scale:

$$z_k^{(j)} = \gamma_0 + \gamma_1 \left[(c_k^{(j)} - \bar{c}_k^{(j)}) / \sqrt{\text{var}(c_k^{(j)})} \right], \quad (12)$$

where $\bar{c}_k^{(j)}$ and $\text{var}(c_k^{(j)}), j = 1, 2, \dots, J + 1$ are the expected value and variance over the train data set in the same batch, values γ_0 and γ_1 needed to be trained.

- Relu (R) used to avoid the gradient vanishing:

$$h_k^{(j)} = \max(0, z_k^{(j)}),$$

$$j = 1, 2, \dots, J; k = 1, 2, \dots, K_j, \quad (13)$$

- Step 8. Back-propagation implemented with optimizer Adam (see [12]) and the proposed loss, defined in Eq. (14) and Eq.(16) below. Recall that optimal weights, obtained via the back-propagation methods, are referred to as parameters rather than hyper-parameters.

D. THE PROPOSED LOSS FUNCTION

The loss function, L , used for each image in the train set is the proposed generalized cross-focal entropy, defined below.

$$L = -y(1 - \hat{p})^\alpha \log \hat{p}^\beta - (1 - y)(\hat{p})^\alpha \log(1 - \hat{p})^\beta, \quad (14)$$

where a sigmoid function

$$\hat{p} = [1 + \exp(-z_1^{J+1})]^{-1}, \quad (15)$$

is the estimate of y for the train set data ($0 \leq \hat{p} \leq 1$) and the response $y = 1$ or $y = 0$ indicates the truth being glaucoma or normal, respectively. Note that two commonly used loss functions: cross entropy ($\alpha = 0, \beta = 1$) and focal loss ($\beta = 1$) are special cases of Eq. (14).

Consequently, the average batch-loss function L_B for a batch of images used is the average of the associated m loss functions. That is,

$$L_B = (1/m) \sum_{i=1}^m \left[-y_i(1 - \hat{p}_i)^\alpha \log \hat{p}_i^\beta - (1 - y_i)(\hat{p}_i)^\alpha \log(1 - \hat{p}_i)^\beta \right], \quad (16)$$

where the batch size $m = 40$ used in this study. The average batch-loss function L_B is then used in the back-propagation

(Step 8) to update all parameters. In summary, the proposed loss function defined in Eq. (14) is for the train set. Regarding the test set, the estimate of the y for the test set data is: $\hat{y} = 1$ if $\hat{p} > 0.5$, otherwise $\hat{y} = 0$.

Here, we discuss the selection of the parameters α and β in Eq. (14). Based on the experiments studied in this paper, we conclude that the optimal parameters in the proposed loss function are $\alpha = 1, \beta = 1.5$ for the glaucoma detection. Fig. 6 shows the plot of four loss functions: mse $(y - \hat{p})^2$, cross entropy ($\alpha = 0, \beta = 1$), focal loss ($\alpha = 1, \beta = 1$), and the proposed generalized cross-focal loss ($\alpha = 1, \beta = 1.5$) for the case that the true response $y = 1$ and $\hat{p} = 10^{-2}$. Consider an example $\hat{p} = 10^{-2}$ (close to 0), which can be treated as a hard sample since it estimates the true response $y = 1$. The proposed loss (marked in green) is $-1 \times \log(10^{-3}) - 0 = 3$, which is 1.5 times larger than the cross-entropy (marked in red) $-1 \times \log(10^{-2}) - 0 = 2$, which is 2 times larger than the mse loss (marked in black) $(\hat{p} - y)^2 = (10^{-4} - 1)^2 \simeq 1$. That is, the proposed loss focuses on the hard sample (say $\hat{p} = 10^{-2}, y = 1$). For normal samples (say $\hat{p} > 0.5$, the loss between the mse (marked in black) and the proposed one (marked in green) are negligible.

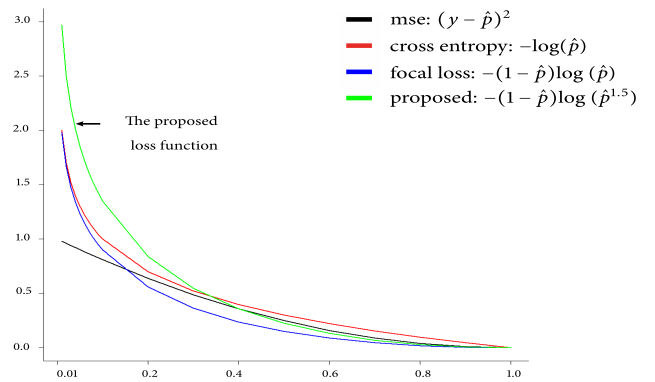


FIGURE 6. Four loss functions, assuming $y = 1$.

We further compare the performance for policy (ROI* + Retinex) using various loss functions: the proposed loss ($\alpha = 1, \beta = 1.5$), focal loss ($\alpha = 1, \beta = 1$) cross-entropy ($\alpha = 0, \beta = 1$), and mse. Table 2 showed that the proposed loss with values $\alpha = 1, \beta = 1.5$ performs the best among 4 loss functions. The difference performances for the rest of three loss functions are negligible. Note that the standard errors (se, denoted as σ^*) listed in the parenthesis, to the right of the associated estimates, are further explained in Subsection III-E.

E. THE POINT ESTIMATE AND ITS STANDARD ERROR

All values listed in Table 2 are estimated performances (estimated accuracy, estimated sensitivity, and estimated specificity). The robustness of the estimated performances depends on the associated standard errors, which are used as the quality measures of the estimated performances.

TABLE 2. Performance among various Loss functions for Policy (ROI* + Retinex).

Loss Functions	Sensitivity mean (σ^*)	Specificity mean (σ^*)	Accuracy mean (σ^*)
Proposed ($\alpha = 1, \beta = 1.5$)	0.95 (0.03)	0.98 (0.03)	0.97 (0.01)
Focal Loss ($\alpha = 1, \beta = 1$)	0.93 (0.05)	0.92 (0.05)	0.92 (0.04)
Cross Entropy ($\alpha = 0, \beta = 1$)	0.91 (0.03)	0.93 (0.05)	0.92 (0.03)
Mse	0.92 (0.03)	0.93 (0.05)	0.93 (0.03)

Let $\hat{\Theta}$ denote the estimated performance such as estimated sensitivity. And the associated standard error is denoted as σ^* , where

$$\sigma^* = \sqrt{\text{var}(\hat{\Theta})/n}, \tag{17}$$

assuming that all samples adopted to calculate $\hat{\Theta}$ are independent.

Motivated by the question of which point-estimator digits to report in a statistical experiment, [33] and [34] proposed a leading digit rule (LDR) which is used to decide the position of the right-most reported digit. LDR rule is equivalent to discard digits that the practitioner considers to be meaningless (because of sampling error). LDR rule reports the point estimate through the leading digit of the standard error, where the leading digit of the standard error is the first left-most non zero digit of the standard error. All reported digits via LDR are meaningful and unreported digits are meaningless. See [33] and [34] for further explanation of the meaningless digits in terms of the probability sense.

Below, we demonstrate two examples to show how LDR suggests the reported estimated performances. Ex 1. Let the estimated sensitivity is 0.954 and its standard error $\sigma^* = 0.03$. LDR suggests to report the estimated sensitivity 0.95 (i.e., up to the hundredth digit). Ex 2. Let the estimated sensitivity is 0.954 and its standard error $\sigma^* = 0.1$. LDR suggests to report the estimated sensitivity 0.9 (i.e., up to the tenth digit). That is, in Ex. 2, LDR does not report digits 5 and 4 because those digits are meaningless.

F. ROBUST AND OPTIMAL DESIGN

Here, we describe the optimal values of the proposed hyper-parameters in three steps.

Step 1. We first adopted a 2^{7-3}_{IV} fractional factorial design with resolution IV, where there were eight factors (denoted as A through H in subsection III-C, each with two levels. Specifically, A. (64, 96), (78, 118); B. 3, 9; C. 16, 32; D. 3, 9; E. 16, 32; F. Delete or filling with black for the area outside the ROI*, G: (0, 1), (1, 1.5).

Results of the 16 combinations (runs) in the 2^{7-3}_{IV} fractional factorial design are listed in Table 3, where the mean accuracy and the standard error (σ^*) are shown in the last two associated columns. The listed data in the right most three columns are explained as follows: $\underline{g}_j^{(i)}$ denotes a vector containing $g_j^{(i)}(k)$, the accuracy at the i th replication, j th combination,

TABLE 3. 2^{7-3}_{IV} fractional factorial design.

Comb. (Run)	A	B	...	F	Ori. Data	Mean Accuracy	Standard Error
1	-	-	...	-	$\underline{g}_1^{(1)}, \underline{g}_1^{(2)}$	$\bar{g}_1^{(1)}, \bar{g}_1^{(2)}$	$s_1^{(1)}, s_1^{(2)}$
2	+	-	...	-	$\underline{g}_2^{(1)}, \underline{g}_2^{(2)}$	$\bar{g}_2^{(1)}, \bar{g}_2^{(2)}$	$s_2^{(1)}, s_2^{(2)}$
...
16	+	+	...	+	$\underline{g}_{16}^{(1)}, \underline{g}_{16}^{(2)}$	$\bar{g}_{16}^{(1)}, \bar{g}_{16}^{(2)}$	$s_{16}^{(1)}, s_{16}^{(2)}$

TABLE 4. Effect and P-values w.r.t. mean response.

Term	Effect	Coef	T-Value	P-Value
A (Image size)	-0.0313	-0.0156	-1.06	0.303
B (Conv. layers)	-0.1313	-0.0657	-4.47	0.00
C (No. kernel)	0.0178	0.0089	0.6	0.554
D (hidden layers)	-0.0776	-0.0388	-2.64	0.018
E (No. nodes)	-0.0028	-0.0014	-0.09	0.926
F (delete or fill)	-0.1316	-0.0657	-4.48	0.00
G (para. α, β in loss fun.)	-0.0706	-0.0353	-2.4	0.029

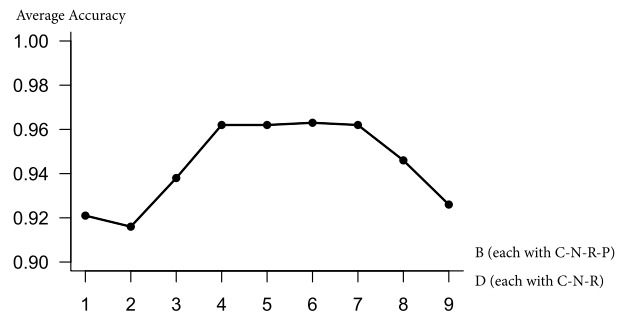


FIGURE 7. Optimal values for Factors B and D.

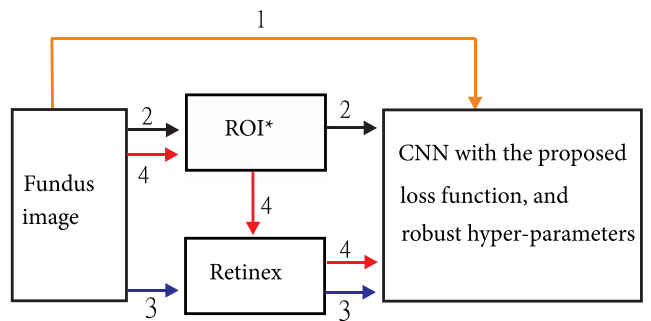


FIGURE 8. Four policies: Policy 4 (ROI* + Retinex) is the proposed model.

and k th block, $i = 1, 2; j = 1, 2, \dots, 16; k = 1, 2, \dots, 10$. The associated mean and standard deviation are denoted as $\bar{g}_j^{(i)}$ and $s_j^{(i)}$.

Step 2. Statistical inferences. Based on the two rightmost columns, we constructed the statistical inferences for all effects and their associated p-values. Results (based on mean and standard deviation) show that only main factors B, D, and G are significant. Table 4 shows that with respect to mean

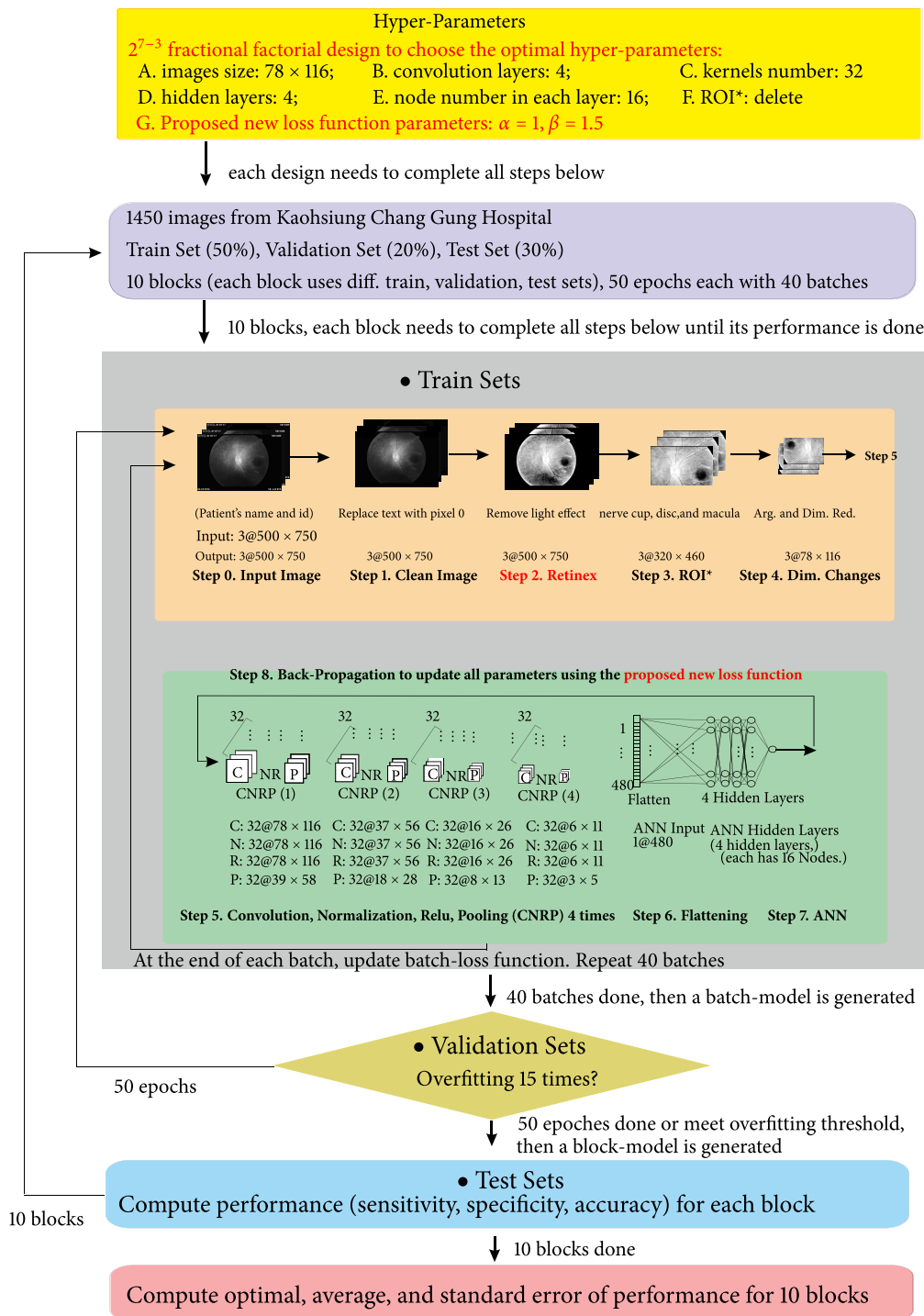


FIGURE 9. Framework-2 of the proposed methodology.

response, only the p-values for factors B, D, F, and G are less than 0.05. (To conserve space, higher-iterations factors and non-significant factors are not shown here. The effect table with respect to the standard error is not shown here either.)

Step 3. Optimal design. Because the coefficients of factors B and D are both negative (specifically, -0.1313 and -0.0776 , shown in Column 2 of Table 4), we continue to investigate the DOE for factor B and D in relation to the layer

numbers (6, 6), (5, 5), (4, 4), (3, 3), where the first figure in each parenthesis indicating the number of convolutional layers each with 4 transformations (C-N-R-P), and the second figure in each parenthesis indicating the number of NN-hidden layers each with 3 transformations (C-N-R). The average accuracy continues to increase up to (4, 4), but decreases at (3, 3). We conclude that the optimal design centers on (4, 4). That is, we chose $B = 4$ each with 4 transformations

TABLE 5. Mean performance and its standard error (σ^*) for the Four Policies: 10 Blocks, 2 Replications, CNN with the proposed loss function ($\alpha = 1, = 1.5$).

Policy	ROI*	Retinex	Sensitivity mean (σ^*)	Specificity mean (σ^*)	Accuracy mean (σ^*)
1	No	No	0.72 (0.13)	0.63 (0.23)	0.69 (0.21)
2	Yes	No	0.80 (0.12)	0.72 (0.13)	0.78 (0.13)
3	No	Yes	0.90 (0.04)	0.96 (0.05)	0.91 (0.05)
4	Yes	Yes	0.95 (0.03)	0.98 (0.03)	0.97 (0.01)

TABLE 6. Optimal Performance and its standard error (σ^*) for the Four Policies: 10 Blocks, 2 Replications, CNN with the proposed loss function ($\alpha = 1, = 1.5$).

Policy	ROI*	Retinex	Sensitivity mean (σ^*)	Specificity mean (σ^*)	Accuracy mean (σ^*)
1	No	No	0.92 (0.1)	0.53 (0.1)	0.77 (0.1)
2	Yes	No	0.90 (0.1)	0.65 (0.1)	0.80 (0.1)
3	No	Yes	0.97 (0.04)	0.98 (0.05)	0.97 (0.05)
4	Yes	Yes	0.97 (0.03)	0.98 (0.03)	0.98 (0.01)

(C-N-R-P), and $D = 4$ each with 3 transformations (C-N-R). Fig. 7 illustrates the plot of the average accuracy as a function of factors B and D, where the x-axis is the layer no. for B (each layer with C-N-R-P) and D (each layer with C-N-R).

IV. PERFORMANCE

This section compared four policies, illustrated in Fig. 8, in terms of the sensitivity, specificity, and accuracy. Policies were provided for the complete cross-section range, in which Policy 1 adopted no ROI and no Retinex, while Policy 4 adopted Retinex and ROI. Table 5 listed the associated mean and standard error (σ^*), and Table 6 listed the associated optimal combination. All statistical data (mean and its standard error σ^*) were computed based on 10 blocks setting, each with 2 replications, described in Section III-A. Note that all policies used CNN with the proposed loss function ($\alpha = 1, = 1.5$).

Results shown in Tables 5 and 6 are summarized below.

- 1) Both Retinex and ROI* were significant factors in terms of the estimated performance and its standard error (σ^*) for accuracy, sensitivity, and specificity. However, Retinex showed greater significance than ROI*.
- 2) Policies involving Retinex (Policies 3 and 4) performed significantly better than those without Retinex (Policies 1 and 2) in terms of both estimated performance and its standard error. Specifically, results in Table 5 showed that the estimated performance were all above 0.90 for Policies 3 and 4 involving Retinex, while the estimated performance were all below 0.80 for Policies 1 and 2 without Retinex. The associated standard errors were all below 0.05 for Policies 3 and 4 involving Retinex, while the associated standard errors were all above 0.1 for Policies 1 and 2 without Retinex.
- 3) Policies involving ROI* (Policies 2 and 4) performed better than those without ROI* under the same conditions. As such, Policy 2 performed better than Policy 1, and Policy 4 performed better than Policy 3.

- 4) Policy 4 significantly performed the best of all four policies. Results in Table 5 showed that the associated estimated performances for Policy 4 were all above 0.95, and the associated standard errors were all below 0.03. Results in Table 6 showed that the optimal performance for Policy 4 was (0.97, 0.98, 0.98).

V. SUMMARY AND CONCLUSION

This study proposed an effective and robust framework for detection of glaucoma that integrates deep learning technologies and statistical methodologies. These technologies include (a) Retinex, a color enhancement algorithm that removes the effects of fundus photography flash and restores the original colors of the fundus image; (b) the extraction of critical areas for analysis (including the optic nerve cup, optic disc, macula, and minimal area of optic nerve fiber layer); and (c) a basic CNN with 4 convolutional layers using the proposed loss function $L = -y(1 - \hat{p})^\alpha \log \hat{p}^\beta - (1 - y)(\hat{p})^\alpha \log(1 - \hat{p})^\beta$ with the optimal parameters $\alpha = 1, \beta = 1.5$ and with the optimal hyper-parameters (such as input image size, convolution layer, and hidden layer) obtained via a robust design of experiment.

The proposed framework was illustrated in Fig. 2 (a simple form) and Fig. 9 (a thorough form) using 1450 color fundus images provided by KCGM Hospital in Taiwan. The proposed framework outperformed other approaches with respect to effectiveness, robustness, simplicity, and clarity. The effectiveness was demonstrated via the estimated sensitivity 0.95, specificity 0.98, and accuracy 0.97 in different training, validation, and test set settings. The robustness was demonstrated via the associated standard error all below 0.03. The simplicity was shown via the adopted basic CNN model (with 4 convolutional layers) compared to deep CNNs such as GoogleLeNet (with 21 convolutional layers) or ResNet152 (with 152 convolutional layers). The clarity is demonstrated via intuitive graphs and clear mathematical notations which make it easy for others to reproduce our results. Overall, the proposed framework advances the field of glaucoma detection.

ACKNOWLEDGMENT

The authors would like to thank anonymous referees for their helpful comments, which greatly improved the quality of this article. They would also like to thank Yu-Chih Shih, Shao-Hsin Chang, Tai-Rung Chen, and Samping Chuang for help with the computer programs.

REFERENCES

- [1] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscipl. Rev., Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.
- [2] U. R. Acharya, S. Dua, X. Du, V. S. Sree, and C. K. Chua, "Automated diagnosis of glaucoma using texture and higher order spectra features," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 3, pp. 449–455, May 2011.
- [3] R. Bock, J. Meier, G. Michelson, L. G. Nyúl, and J. Hornegger, "Classifying glaucoma with image-based features from fundus photographs," in *Proc. Joint Pattern Recognit. Symp.* Berlin, Germany: Springer, 2007, pp. 355–364.

- [4] N. R. de Sales Carvalho, M. da Conceição Leal Carvalho Rodrigues, A. O. de Carvalho Filho, and M. J. Mathew, "Automatic method for glaucoma diagnosis using a three-dimensional convoluted neural network," *Neurocomputing*, vol. 438, pp. 72–83, May 2021.
- [5] X. Chen, Y. Xu, D. W. K. Wong, T. Y. Wong, and J. Liu, "Glaucoma detection based on deep convolutional neural network," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2015, Art. no. 201537.
- [6] A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 476–480, May 1999.
- [7] H. Fu, J. Cheng, Y. Xu, C. Zhang, D. W. K. Wong, J. Liu, and X. Cao, "Disc-aware ensemble network for glaucoma screening from fundus image," *IEEE Trans. Med. Imag.*, vol. 37, no. 11, pp. 2493–2501, Nov. 2018.
- [8] M. Al Ghamdi, M. Li, M. Abdel-Mottaleb, and M. A. Shousha, "Semi-supervised transfer learning for convolutional neural networks for glaucoma detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3812–3816.
- [9] S. Gheisari, S. Shariflou, J. Phu, J. P. Kennedy, A. Agar, M. Kalloniatis, and M. Golzan, "A combined convolutional and recurrent neural network for enhanced glaucoma detection," *Sci. Rep.*, vol. 11, Jan. 2021, doi: 10.1038/s41598-021-81554-4.
- [10] A. Ghosh, A. Sarkar, A. S. Ashour, D. Balas-Timar, N. Dey, and V. E. Balas, "Grid color moment features in glaucoma classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 9, pp. 1–14, 2015.
- [11] J. Gilles, "Empirical wavelet transform," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 3999–4010, Aug. 2013.
- [12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [13] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.
- [14] M.-L. Huang, H.-Y. Chen, and J.-J. Huang, "Glaucoma detection using adaptive neuro-fuzzy inference system," *Expert Syst. Appl.*, vol. 32, no. 2, pp. 458–468, Feb. 2007.
- [15] A. Issac, M. P. Sarathi, and M. K. Dutta, "An adaptive threshold based image processing technique for improved glaucoma detection and classification," *Comput. Methods Programs Biomed.*, vol. 122, no. 2, pp. 229–244, Nov. 2015.
- [16] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.
- [17] I. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer, 2011.
- [18] F. Khan, S. A. Khan, U. U. Yasin, I. U. Haq, and U. Qamar, "Detection of glaucoma using retinal fundus images," in *Proc. 6th Biomed. Eng. Int. Conf. (BMEiCON)*, Oct. 2013, pp. 1–5.
- [19] S. J. Kim, K. J. Cho, and S. Oh, "Development of machine learning models for diagnosis of glaucoma," *PLoS ONE*, vol. 12, no. 5, May 2017, Art. no. e0177726.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [21] E. H. Land and J. J. McCann, "Lightness and Retinex theory," *J. Opt. Soc. Amer.*, vol. 61, no. 1, pp. 1–11, 1971.
- [22] J.-F. Le Gall, "Random trees and applications," *Probab. Surv.*, vol. 2, pp. 311–345, Jan. 2005.
- [23] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.
- [24] S. Maheshwari, R. B. Pachori, and U. R. Acharya, "Automated diagnosis of glaucoma using empirical wavelet transform and correntropy features extracted from fundus images," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 3, pp. 803–813, May 2017.
- [25] J. Pruthi and S. Mukherjee, "Computer based early diagnosis of glaucoma in biomedical data using image processing and automated early nerve fiber layer defects detection using feature extraction in retinal colored stereo fundus images," *Int. J. Sci. Eng. Res.*, vol. 4, no. 4, pp. 1822–1828, 2013.
- [26] U. Raghavendra, H. Fujita, S. V. Bhandary, A. Gudigar, J. H. Tan, and U. R. Acharya, "Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images," *Inf. Sci.*, vol. 441, pp. 41–49, May 2018.
- [27] R. G. Ramani, S. Sugirtharani, and B. Lakshmi, "Automatic detection of glaucoma in retinal fundus images through image processing and data mining techniques," *Int. J. Comput. Appl.*, vol. 166, no. 8, pp. 38–43, May 2017.
- [28] S. Samanta, S. S. Ahmed, M. A.-M. M. Salem, S. S. Nath, N. Dey, and S. S. Chowdhury, "Haralick features based automated glaucoma classification using back propagation neural network," in *Proc. 3rd Int. Conf. Frontiers Intell. Comput., Theory Appl. (FICTA)*. Cham, Switzerland: Springer, 2015, pp. 351–358.
- [29] S. Serte and A. Serener, "A generalized deep learning model for glaucoma detection," in *Proc. 3rd Int. Symp. Multidisciplinary Stud. Innov. Technol. (ISMSIT)*, Oct. 2019, pp. 1–5.
- [30] S. Serte and A. Serener, "Graph-based saliency and ensembles of convolutional neural networks for glaucoma detection," *IET Image Process.*, vol. 15, no. 3, pp. 797–804, Feb. 2021.
- [31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [32] S. Sreng, N. Maneerat, K. Hamamoto, and K. YadanarWin, "Deep learning for optic disc segmentation and glaucoma diagnosis on retinal images," *Appl. Sci.*, vol. 10, no. 14, p. 4916, 2020, doi: 10.3390/app10144916.
- [33] W. T. Song and B. W. Schmeiser, "Omitting meaningless digits in point estimates: The probability guarantee of leading-digit rules," *Oper. Res.*, vol. 57, no. 1, pp. 109–117, Feb. 2009.
- [34] W. T. Song and B. W. Schmeiser, "Displaying statistical point estimates using the leading-digit rule," *IIE Trans.*, vol. 43, no. 12, pp. 851–862, Dec. 2011.
- [35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [36] S. Targ, D. Almeida, and K. Lyman, "Resnet in resnet: Generalizing residual architectures," 2016, *arXiv:1603.08029*. [Online]. Available: <http://arxiv.org/abs/1603.08029>
- [37] Y. Yu, X. Si, C. Hu, and Z. Jianxun, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, 2019.



WHEYMING TINA SONG received the Ph.D. degree from the School of Industrial Engineering, Purdue University. She is currently a Distinguished Professor with the Department of Industrial Engineering and Engineering Management (IEEM), National Tsing Hua University (NTHU), Taiwan. Her research interests include methodologies and applications on probability and statistics, simulation, and machine learning. The applications include glaucoma detection, arteriovenous-fistula-stenosis detection, and silicon-carbide-wafers defect detection. She won the 1988 IIE Doctoral Dissertation Award in USA.



ING-CHOU LAI received the M.D. degree from Kaohsiung Medical University, Taiwan. He was the Chief of glaucoma section in ophthalmology with Kaohsiung Chang Gung Memorial Hospital, Taiwan, from 2002 to 2013. He was a Lecturer with Chang Gung University. He is currently a Consultant Doctor of ophthalmology with Chiayi Chang Gung Memorial Hospital, Taiwan. His research interests include general ophthalmology and glaucoma.



YI-ZHU SU received the M.S. degree from the Department of Industrial Engineering and Engineering Management, National Tsing Hua University (NTHU), Taiwan, in 2017, where she is currently pursuing the Ph.D. degree. Her research interests include network resilience analysis, machine learning, and statistical applications.

• • •