Ensemble Convolutional Neural Networks for the Classification and Visualization of Retinal Diseases in Optical Coherence Tomography Images

Jongwoo Kim
Lister Hill National Center for Biomedical Communications
National Library of Medicine, National Institutes of Health
Bethesda, USA
jongkim@mail.nih.gov

Loc Tran
Lister Hill National Center for Biomedical Communications
National Library of Medicine National Institutes of Health
Bethesda, USA
lotran@mail.nih.gov

***Abstract*—Optical Coherence Tomography (OCT) is a non-invasive imaging technique that uses light waves to capture cross-sectional images of patients' retina layers, allowing for the diagnosis of various retinal diseases. Ophthalmologists use OCT images to decide whether to perform anti-Vascular Endothelial Growth Factor therapy. However, it is time-consuming work to analyze the images since OCT provides several images for each patient. This paper proposes an ensemble learning (EL) model, based on three deep learning models, that categorize patients' OCT images into four categories such as Choroidal neovascularization, Diabetic macular edema, Drusen, and Normal. Four different Convolutional Neural networks (CNNs) are adapted to train the images. Among them, three CNNs are selected for the proposed EL model such as VGG19, ResNet152, and DenseNet121. Two different voting methods (soft and hard) are also used in the proposed EL model. The proposed EL model shows 0.9930 accuracy, 0.9930 sensitivity, and 0.9977 specificity. New heatmap algorithm is also proposed, based on positive and negative heatmaps, to analyze activity of CNN models and estimate regions of interest from the OCT images accurately. The proposed EL model and heatmap algorithms shows relatively good performance compared to other CNN models and heatmap algorithms. The proposed EL shows the potential to work as a second reader for ophthalmologist.**

Keywords—Deep Learning (DL), Ensemble Learning (EL), Convolutional Neural Networks (CNNs), Heatmap, Optical coherence tomography (OCT)

I. INTRODUCTION

Optical coherence tomography (OCT) is a non-invasive test to take cross-section pictures of retina layer [1]. Ophthalmologists use it to diagnose retina's distinctive layers and measure their thickness. Therefore, OCT images have been an important modality for the detection and quantification of retinal diseases and retina abnormities such as age-related macular degeneration (AMD), diabetic macular edema (DME), choroidal neovascularization (CNV), and drusen. Macular thickness maps and OCT images can also provide guidance for treatment response to anti-VEGF therapy for DME and AMD patients [2]. Unfortunately, it is a time consuming work for ophthalmologist to analyze the images since OCT provides several images for each patient.

Deep learning (DL) has been adapted in medical data analysis recently and shows robust performance compared with existing image processing and machine learning algorithms. It is also adapted in the segmentation and classification of OCT images. Li *et al*. use VGG16 [3] to train OCT images to categorize them into four classes [4]. Lu *et al.* adapt ResNet101 [5] to classify cystoid macular edema, macular hole, epiretinal membrane, and serous macular detachment from OCT images [6]. Li. *et al.* adapt EL based on ResNet50 [5] to classify the images into the four classes [7]. Islam *et al.* train DenseNet201 [8] and ten other CNNs for the classification of OCT images into four classes [9]. Kermany et al. use Inception V3 [10] to classify OCT images into four classes such as CNV, DME, Drusen, and Normal [11].

DL shows promising performance in the classification and segmentation of medical images. However, there are issues since the DL is a black box and does not provide any interpretable explanations for their decisions. Therefore, there are several studies to resolve this issue. Zeiler *et al.* introduce a visualization technique that reveals the input stimuli that excite individual feature map at any intermediate layer in the model based on Deconvolutional Network [12]. Zhou *et al*. show that convolutional layers have ability to localize objects despite no supervision on the locations of the objects. Later, they introduce Class Activation Map (CAM) to visualize activity of the last convolutional layers for localization/Regions of Interest (ROIs) detection in image classification [13, 14]. It uses last convolutional layer and weights connected between the last convolutional layer and output layer for each class to generate the map. Selvaraju *et.al*. adapt gradient information of an output class to improve CAM and propose Gradient-weighted Class Activation Mapping (Grad-CAM) to produce a localization map highlighting the ROIs in the images accurately in image classification [15]. The method can be applied in several CNN models. Kim *et al.* propose a new map called Class-Selective Relevance Map (CRM) [16]. CRM also uses last convolutional layer and weights connected between the convolutional layer and output layer for each class. It measures the importance of the activation at each location in the feature maps by squaring the difference between whole feature map activation value and whole feature map value without using a value in the location.

Most CNNs include one or more dense layers between the feature extractor and output layer, and understanding the weights connecting the extractor and layer are critical to comprehend the activity of CNNs. However, above heatmap methods do not account for dense layers or do not demonstrate on CNNs with dense layers.

There are positive and negative values in the feature map for each class in image classification case. However, all these

This research/work was supported in part by the Lister Hill National Center for Biomedical Communications of the National Library of Medicine (NLM), National Institutes of Health (NIH).

2372-9198/23/$31.00 ©2023 IEEE
DOI 10.1109/CBMS58004.2023.00031

methods [12-16] do not consider negative values in the map. Negative value area for a class means that the area is not the feature area for the class but can be feature areas for the other classes. Therefore, when we generate separate feature maps for positive and negative values for each class, we can analyze activation of DL algorithms clearer and make better heatmap to estimate ROIs from the map.

In this paper, we propose new approaches to classify OCT images into four different classes (CNV, DME, Drusen, and Normal) using EL, two voting methods, and three CNN classifiers. We also propose a new heatmap, called Class-based Positive Negative Map, to estimate ROIs for each class in CNNs accurately, including those with dense layers.

The remainder of this paper is organized as follows: Section II describes methods that we use, Section III describes our experimental results and discussion in detail, and section IV is our conclusion of findings.

II. METHODS

A. Dataset

An OCT image dataset publicly available is used in our experiment [11]. The dataset is composed of 108,309 training images and 1,000 test images. There are four classes in the dataset: CNV, DME, Drusen, and Normal. Unfortunately, the training dataset is imbalanced. There are 37,205 images for CNV; 11,348 images for DME; 8,616 images for Drusen; and 51,140 images for Normal. The largest class (Normal) is about six times larger than the smallest class (Drusen). The test dataset is composed of 250 images in each of the four classes. Fig. 1 shows four images from each class in the dataset.

There are two boundaries of interest in the OCT image. Inner Limiting Membrane (ILM) is the upper boundary of retina layer (that separates the background from the retina layer). Chorio-Scleral Interface (CSI) is the lower boundary of the retina layer (that separates the retina layer from the background). Retinal Pigment Epithelium (RPE) is the brightest layer located in the middle between the ILM and CSI. In Fig. 1, CNV shows variation in below/above RPE or intra-retinal area (showing macular fluid in black color, bumpy RPE layer, and poorly defined boundaries). DME has intra-retinal fluids between ILM and RPE shown in dark color. Drusen has deformation or thickening in RPE layer instead of a flat shape.

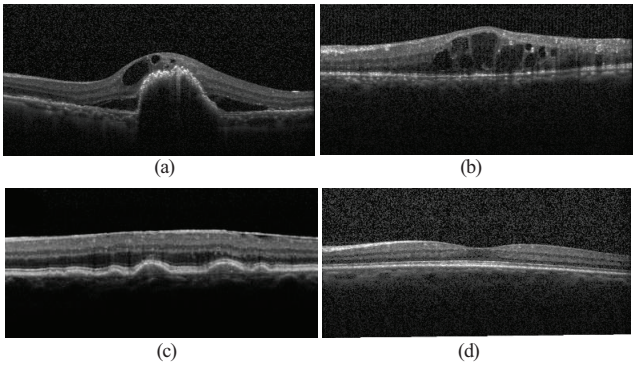


Figure 1. Some examples of OCT images in each class in the dataset (a) CNV, (b) DME, (c) Drusen, and (d) Normal.

B. Image Normalization in the Dataset

There are several different size and quality images in the dataset. Fig. 2 shows examples of images in the dataset for DME class. OCT images usually have a rectangle shape (Fig. 2(a)). However, majority images have a square shape (Fig 2(b)-(d)). Most OCT images have black and white noises, but some images have more serious noises (Fig. 2(b)) than normal images. Several images have tilted retina layers as shown in Figs. 2(c) and (d). Zoom ratio of each image varies and is not consistent with one another. There are white background noises in the several images.

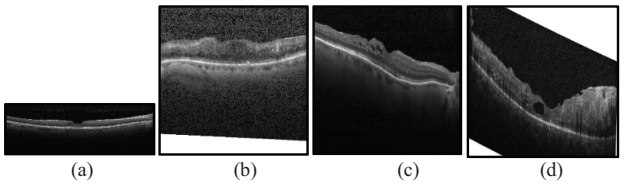


Figure 2. OCT images of the CNV class in the dataset.

Therefore, we perform several preprocessing steps to improve the quality and consistency of the images in the dataset as shown in Fig. 3. First, fill the white background to black color in the image, and then make the image into a square shape using the longest side of either height or width of the image. Second, segment retina layer using the Fully Convolutional Network (FCN) having a U-Nets architecture. Third, remove background noise (pepper and salt noise) using the segmentation results as a mask. Fourth, adjust the tilted retina layer. Fifth, crop the retina layer from the image. Sixth, resize the image into a square shape for the input of DL models for the classification. For the FCN, inputs of the FCN are 224×224×3 color images and output are 224×224×1 gray images. 1,000 images (250 images from each class) are used to train the FCN. A detailed description of the process is described in [17].

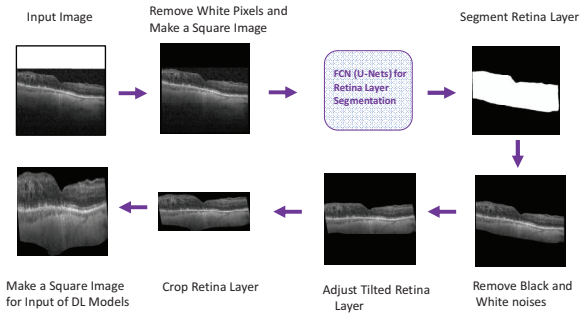


Figure 3. Preprocessing procedure to standardize OCT images.

C. CNN Models Used for Classification of OCT Images

Four DL models from ImageNet [18] are used as feature extractors for the classification such as VGG19, ResNet50, ResNet152, and DenseNet121. Fig. 4 shows the proposed CNN architecture. After a feature extractor, six more layers are added; global average pooling layer, two 1×1,024 fully connected (dense) layers, two dropout layers, and 1×4 Softmax

output layer. Pre-trained weights of the ImageNet are used as initial weights for training.

D. Proposed Ensemble Learning Model Based on Three CNNs and Two Voting Methods

We propose an EL-based model using three CNNs as base classifiers (see Fig. 4) to improve the performance. Fig. 5 shows the architecture of the model. Unlike other EL models, two different voting methods are used for the first time to our knowledge: Soft Voting (SV) and Hard Voting (HV). SV estimates average of the three CNN results (Eq. (1)) and HV estimates average of CNN results after categorizing each CNN result (Eq. (2)). After then, we estimate a weighted average of the two voting results for the final model output as shown in Eq. (3). The HV results can provide more impact for each CNN results and SV compensates winner-take-all issue of HV.

$$SV\ (\mathrm{X}_1, \mathrm{X}_2, \dots, \mathrm{X}_n) = \frac{\sum_{i=0}^{n} \mathrm{X}_i}{n} \quad (1)$$

$$HV(\mathrm{X}_1, \mathrm{X}_2, \dots, \mathrm{X}_n) = \frac{\sum_{i=0}^{n} Categorical\ Output(\mathrm{X}_i)}{n} \quad (2)$$

$$Hybrid\ Voting(\mathrm{X}_1, \mathrm{X}_2, \dots, \mathrm{X}_n) = w \times SV(\mathrm{X}_1, \mathrm{X}_2, \dots, \mathrm{X}_n) + (1-w) \times HV(\mathrm{X}_1, \mathrm{X}_2, \dots, \mathrm{X}_n) \quad (3)$$

In the equations, X_i is output of a CNN model i, n is number of CNNs, and w is a weight parameter (w=0.5 is used in this experiment). X_i contains four classification score values for the four classes in this experiment. *Categorical Output* (X_i) means the maximum value in X_i becomes 1 and the others become 0.

E. Proposed Class-based Positive Negative Map

Heatmap shows an activity of the DL model graphically for an input image. However, all existing heatmaps (CAM, Grad-CAM, and CRM) are implemented for CNNs without having dense layer or do not consider all weights connecting the last convolutional layer and output layer.

There are positive and negative values in the last convolutional layer (last feature maps). The negative values in a class can provide positive or negative impact to other classes based on weights connecting between the convolutional layer and output layer. However, existing methods do not consider the role of negative values that exist between classes. CRM treats the negative values the same as positive values by squaring the values. In addition, the squaring method intentionally increases gaps between high and low values in the heatmap.

Therefore, we propose a new heatmap algorithm, called Class-based Positive Negative Map (CPNM), that generates positive and negative heatmap of each class and uses negative heatmap of a class to improve final heatmaps of the other classes. Fig. 6 shows the workflow of estimating the proposed CPNM.

Our proposed CNN (Fig. 4) is composed of a CNN Feature extractor, two dense layers, and a softmax output layer. The output of last convolutional layer is the input of the first dense layer, the output of the first dense layer is the input of the second dense, and the output of the second dense layer is the input of the softmax output layer.

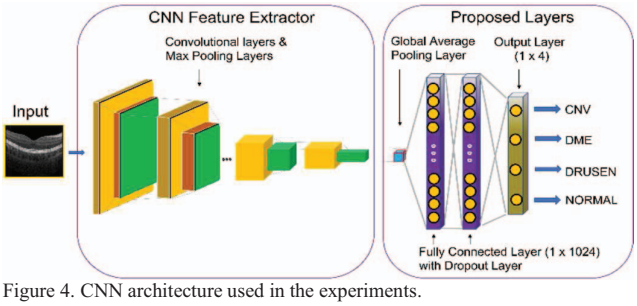


Figure 4. CNN architecture used in the experiments.

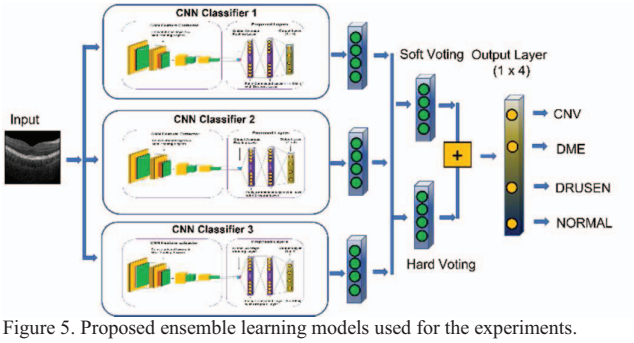


Figure 5. Proposed ensemble learning models used for the experiments.

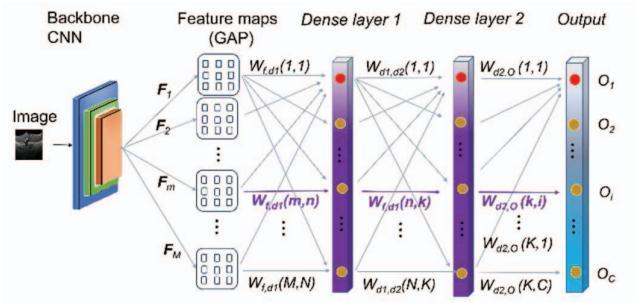


Figure 6. Proposed CNN weights used for the estimation of the proposed Class-based Positive Negative Map (CPNM).

Therefore, a prediction score matrix of class i ($\boldsymbol{P}_i$) is estimated as shown in Eq. (4). $\boldsymbol{P}_i$ has the same row and column of the last convolutional layer. $\boldsymbol{P}_i(x, y)$ is the value at (x, y) in $\boldsymbol{P}_i$. Fig. 6 shows the architecture of the proposed CNN and variables used in the equation.

$$\boldsymbol{P}_i = \sum_{m=1}^{M} F_m \sum_{n=1}^{N} \boldsymbol{W}_{f,d1}(m,n) \sum_{k=1}^{K} \boldsymbol{W}_{d1,d2}(n,k) \times \boldsymbol{W}_{d2,O}(k,i) \quad (4)$$

where $\boldsymbol{F}_m$ is the mth feature map in last convolutional layer, and M is the number of feature maps in the layer. $\boldsymbol{W}_{f,d1}(m,n)$.is the weight between the mth feature map in the last convolutional layer and the nth neuron in the first dense layer, and $\boldsymbol{W}_{f,d1}$ has $M \times N$ dimension. $\boldsymbol{W}_{d1,d2}(n,k)$ is the weight between the nth neuron in the first dense layer and the kth neuron in the second dense layer, and $\boldsymbol{W}_{d1,d2}$ has $N \times K$ dimension. $\boldsymbol{W}_{d2,O}(k,i)$ is the weight between the kth neuron in the second dense layer and

the *i*th neuron (class *i*) in the output layer, and $W_{d2,O}$ has $K \times C$ dimension.

From the Eq. (4), we estimate positive and negative CPNM values of each class as shown in Eqs. (5) and (6). $P_{i,pos}(x,y)$ and $P_{i,neg}(x,y)$ mean positive and negative heatmap values of class *i* at (*x, y*).

$$P_{i,pos}(x,y) = P_i(x,y) \quad if\ P_i(x,y) > 0 \qquad (5)$$
$$P_{i,pos}(x,y) = 0 \quad if\ P_i(x,y) < 0$$

$$P_{i,neg}(x,y) = |P_i(x,y)| \quad if\ P_i(x,y) < 0 \qquad (6)$$
$$P_{i,neg}(x,y) = 0 \quad if\ P_i(x,y) > 0$$

The positive value areas in $P_{i,\,neg}$ mean the areas are not used as feature areas for the class *i*, but the areas can be used as feature areas for the other classes. However, $P_{i,\,neg}$ does not provide information of the classes related to the areas. Therefore, a mask is used to provide positive effect to specific classes.

The mask M is a matrix of 1 and 0 (true and false). If the $P_{i,pos}(x,y)$ is greater than a threshold t (t = 0.25× max ($P_{i,pos}$, $P_{i,neg}$) in this experiment), the mask $M_i(x, y)$ becomes 1, otherwise it becomes 0.

$$M_i(x,y) = 1, \quad if\ P_{i,pos}(x,y) \geq t \qquad (7)$$
$$M_i(x,y) = 0, \quad if\ P_{i,pos}(x,y) < t$$

Based on Eqs. (4)-(7), we propose the following formulas to generate a new heatmap, called CPNM as shown in Eqs. (8) and (9). Eq. (8) shows the proposed heatmap value for class *i* at (*x,y*) and Eq. (9) shows combined values for all classes at (*x,y*).

$$CPNM_i(x,y) = P_{i,pos}(x,y) \times M_i(x,y) \times \prod_{j=0, j\neq i}^{C} P_{j,neg}(x,y) \qquad (8)$$

$$CPNM(x,y) = \sum_{i=1}^{C} CPNM_i(x,y) \qquad (9)$$

where *C* is number of classes. Since $CPNM_i$(x, y) is based on multiplication, $CPNM_i$(x, y) can become zero if one of values in the heatmaps becomes zero. Therefore, value one is added to $P_{i,pos}(x, y)$ and $P_{i,neg}(x, y)$ to resolve the issue.

Fig. 7 shows the examples of CPNM of a CNV class image estimated from the proposed CNN based on DenseNet121 extractor. Each column shows heatmaps for each class. The first row shows positive heatmaps ($P_{i,pos}$), the second row shows negative heatmaps ($P_{i,neg}$), and the last row shows the proposed heatmaps ($CPNM_i$). The image in the first row first column is the positive map ($P_{i,pos}$) of CNV class. The heatmap shows that proposed CNN generates strong activities to CNV class and no activity to DME, Drusen, and Normal classes. The negative heatmap ($P_{i,neg}$) in the second rows show activity to DME, Drusen, and Normal classes. Among them the heatmap of the Drusen class shows the moderate activity. It means that the active area should not be used as features for Drusen class. The last row shows the final proposed heatmap for each class ($CPNM_i$). Only the map of CNV class shows activities. Compared with the positive heatmap (first row first column) and final map (last row first column) of CNV class, the proposed heatmap focuses more on specific area in the image and allow us to crop ROIs for CNV easily.

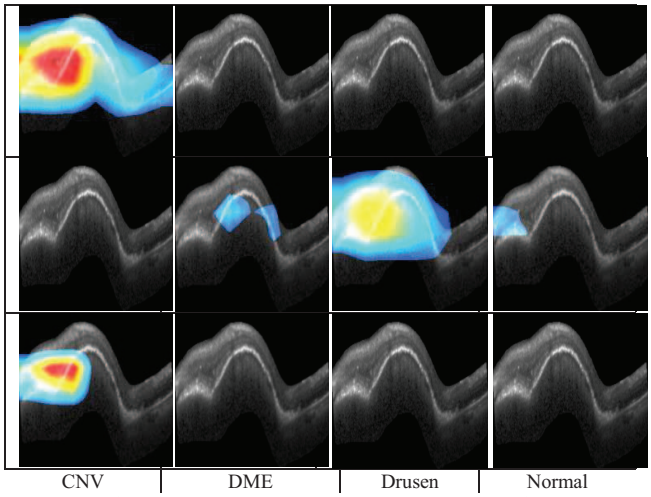


Fig 7. Example of heatmaps from DenseNet121. The first row shows positive heatmaps ($P_{i,pos}$), the second row shows negative heatmaps ($P_{i,neg}$) of each class, and the last row shows the proposed heatmaps ($CPNM_i$).

III. Experimental Results and Discussion

A. Training Dataset Generation and Training Parameters

The training dataset has unbalanced number of images in between classes. Therefore, we generate training sets by sampling five different number of images (2,000, 3,000, 5,000, 6,000, and 8,616) from each class using random sampling method and train CNNs (Fig. 4) using the training sets. The pre-trained weights from ImageNet are used as initial weights and all CNN weights and layers are trained again including the two new dense layers and one output layer.

Python and Keras with TensorFlow [19, 20] is used to implement the CNN models and stochastic gradient descent with learning rate = 0.001, decay=1e-6, momentum = 0.9 and nesterov momentum = True are used for training. We use epochs = 10~100 and choose the weight with the highest accuracy during the training as the final weight. The hardware configuration used for this experiment is 2 × Intel Xeon Gold 5218 processors 2.3 GHz, 64 hyper-thread processors, 8 × RTX 2080 Ti, and Red Hat Enterprise Linux 7.

B. Performance for the Proposed CNNs

Table I shows the performance (accuracy) of each CNN model estimated from the test dataset containing 1,000 images. The first column (Class size) shows number of training images per each class, and the remaining columns show the performance of each CNN model. In the first row, 2,000 means 2,000 images are randomly selected from each class. Since there are four classes in the dataset, total number of images used for training is 8,000. The same rules are applied to all other rows. VGG19 has the best performance at Class size = 2,000 with 0.9860 accuracy. ResNet50 has the best accuracy of 0.9730 at Class size = 6,000. ResNet152 has the best performance at Class size = 8,616 with 0.9810 accuracy.

DenseNet121 also has the best performance of 0.9810 at Class size = 6,000 and 8,616. Overall, VGG19 shows the best performance.

TABLE I. COMPARISON OF THE PROPOSED CNN MODELS

Class size (per each class)	VGG19	ResNet50	ResNet152	DenseNet121
2,000	**0.9860**	0.9620	0.9480	0.9660
3,000	0.9740	0.9700	0.9730	0.9750
5,000	0.9780	0.9710	0.9750	0.9710
6,000	0.9760	**0.9730**	0.9730	**0.9810**
8616	0.9750	0.9700	**0.9810**	**0.9810**

C. Performance of the Proposed Ensemble Learning Model

EL is commonly used in machine learning to improve classification accuracy since it uses the results of multiple machine learning algorithms [21]. Bootstrap aggregating method (or bagging) lets each model to vote or average the results from all models to improve the classification accuracy. Unlike other EL models, that use either voting or averaging, the proposed EL uses the hybrid voting method (SV and HV) as shown in Fig. 5. Table II shows the performance of the proposed EL model. The EL based on the three CNNs in Table II shows the improvement with 0.9930 accuracy, 0.9930 sensitivity, and 0.9977 specificity.

Table III shows the comparison of the proposed EL method with other existing DL methods. Among them, the method using ResNet50 (third row) and our proposed methods use EL methods. The remaining methods use one or more multi-class classifiers. It shows that the proposed EL models shows better performance compared to the other methods.

TABLE II. PERFORMANCE OF THE PROPOSED TREE CNN MODELS AND ENSEMBLE LEARNING METHOD.

Model	Acc.	Sen.	Spec.
VGG19	0.9860	0.9860	0.9954
ResNet152	0.9810	0.9810	0.9936
DenseNet121	0.9810	0.9810	0.9937
Proposed EL using Hybrid Voting (w=0.5)	**0.9930**	**0.9930**	**0.9977**

TABLE III. COMPARISION OF THE PROPOSED ENSEMBLE LEARNING METHOD WITH OTHER ALGORITHMS.

Method	CNN	*Acc.*	*Sen.*	*Spec.*
Lu et al. [6]	ResNet101	0.959	0.942	0.964
Kermany et al [11]	Inception V3	0.966	0.978	0.974
Li et al [7]	EL with ResNet50	0.979	0.968	0.994
Islam et al. [9]	DenseNet101	0.986	0.986	0.995
Li et al. [4]	VGG16	0.986	0.978	0.994
Proposed Ensemble Learning	EL with VGG19 ResNet152 DenseNet121	**0.993**	**0.993**	**0.998**

D. CPNM of the Proposed EL Model

Fig. 8 shows heatmaps ($CPNM_i$) generated by the three CNNs used in the proposed EL model. The first row (a) shows heatmaps of an image of CNV class. All heatmaps show strong activity in similar area, and the heatmap of ResNet121 shows activity in another area also. For the DME class image in the second row (b), all heatmaps show a high level of activity in similar areas.

We compare our heatmaps with CAM and CRM since CRM shows similar or better performance than Grad-CAM [15]. Since CAM and CRM are implemented for CNNs without dense layer, we cannot compare our heatmap with them directly. Therefore, we have modified CAM and CRM to generate heatmaps from CNNs with multiple dense layers based on our proposed algorithm.

Fig. 9 shows heatmaps generated using the DenseNet121-based CNN for input images belonging to the CNV and DME classes. The first and second rows show heatmaps of modified CAM and CRM, and the third row shows the proposed heatmaps. Fig. 9 (a) and (b) shows heatmaps of input images of CNV and DME classes, respectively. In both examples, proposed heatmaps show strong activity only at the right classes in more specific areas and enable us to estimate more accurate ROIs for each disease.

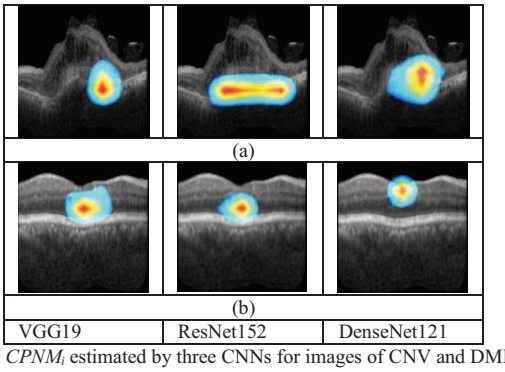
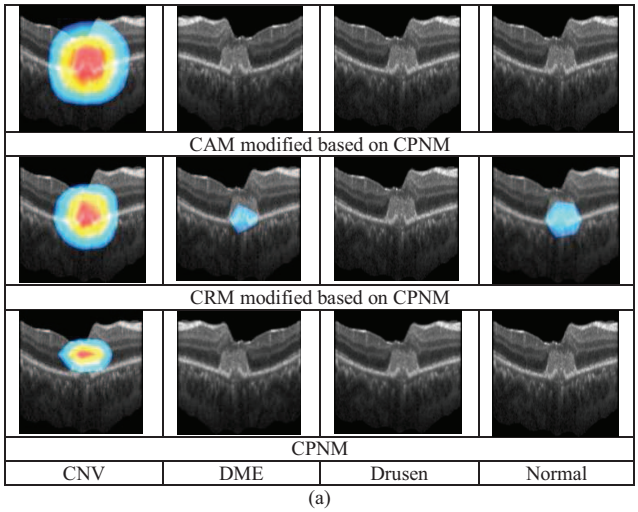
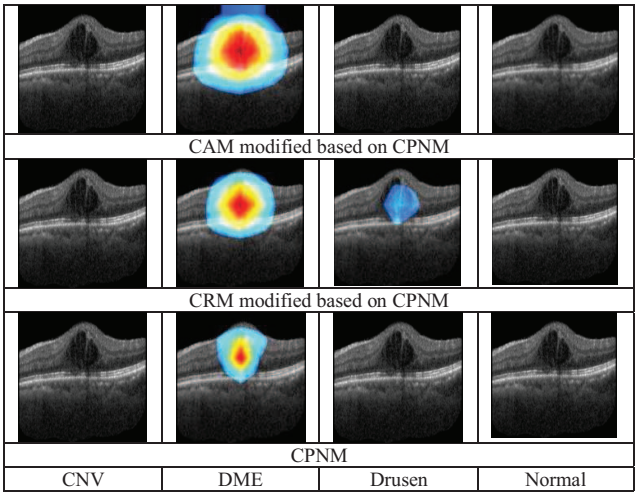


Figure 8. $CPNM_i$ estimated by three CNNs for images of CNV and DME classes. (a) Input image of CNV class. (b) Input image of DME class.



(a)



(b)

Figure 9. Comparison of the proposed heatmap ($CPNM_i$) with CAM and CRM. In (a) and (b), the first and second rows are the heatmaps of CAM and CRM modified based on *CPNM*, and the third row is heatmaps of the proposed $CPNM_i$. The heatmaps are generated using the DenseNet121-based CNN for input images belonging to the CNV class (a) and DME class (b).

IV. CONCLUSIONS

This paper proposes an EL model to classify OCT images into four categories using three CNNs as base classifiers. We use the pre-trained weights on the ImageNet as initial weights and train all weights and layers including two new dense layers and one output layers in our base CNN classifier. Four different CNN models and five different sizes of training datasets are used for training the base classifiers. Among them, VGG19 shows the best performance with 0.986 accuracy, 0.986 sensitivity, and 0.9954 specificity. The proposed EL model, based on the three different CNNs (VGG19, ResNet152, and DenseNet121) and two voting methods, shows more robust results with 0.9930 accuracy, 0.9930 sensitivity, and 0.9977 specificity. We also proposed a new heatmap based on positive and negative heatmap of each class, called CPNM, for the first time to our knowledge and show that CPNM estimates ROIs more accurately compared with other methods. In future work, we plan to further investigate new architectures of the DL model and handcrafted features to improve classification accuracy. In addition, we will investigate heatmap algorithms further to estimate ROIs more accurately.

ACKNOWLEDGMENT

This research was carried out by staff of the National Library of Medicine (NLM), National Institutes of Health (NIH), with support from NLM. We thank Lan Le for generating annotation images to segment retina layers in OCT images. The authors thank, Yolanda L. Jones, NIH Library for editing assistance.

REFERENCES

[1] Americal Academy of Ophthalmology, https://www.aao.org/eye-health/treatments/what-is-optical-coherence-tomography.

[2] S.R. Cohen and T.W. Gardner, "Diabetic Retinopathy and Diabetic Macular Edema", Dev Ophthalmology, Vol 55, pp. 137-146, 2016.

[3] K, Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", International Conference on Learning Representations, 2015.

[4] F. Li, H. Chen, Z Liu, X. Zhang, and Z. Wu, "Fully Automated Detection of Retinal Disorders by Image Based Deep Learning", Gradfe's Archive for Clinical and Experimental Ophthalmology, 257, pp. 495-505, Jaunary 2019

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, Las Vegas, USA, 2016.

[6] W. Lu, Y. Tong, Y. Yu, Y. Xing, C. Chen, and Y. Shen, "Deep Learning-Based Automated Classification of Multi-Categorical Abnormalities From Optical Coherence Tomography Images", Translational Vision Science and Technology (TVST), vol. 7, no. 6, pp.1-10, 2018.

[7] F. Li, H. Chen, Z. Liu, X. Zhang, M. Jiang, Z. Wu, and K. Zhou, "Deep Learning-based Automated Detection of Retinal Diseases Using Ophtical Coherence Tomography Images", Biomedical Optics Express, Vol 10, No 12, pp. 6204-6226, December 2019.

[8] G. Huang, Z Liu, L.V.D. Maaten, K.Q. Weinberger, "Densely Connected Convolutional Networks", CVPR 2017.

[9] K.T. Islam, S. Wijewickrema, and S. O'Leary, "Identifying Diabetic Retinopathy from OCT Images using Deep Transfer Learning with Artificial Neural Networks", 32nd International Symposium on Computer-Based Medical Systems, pp. 281-286, June 2019.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", IWWW Conference on Cimputer Vision and Pattern Recognition, pp. 2818-2826, 2016.

[11] D.S. Kermany, M. Goldbaum, W. Cai, et al., ''Identifying medical diagnoses and treatable diseases by image-based deep learning,'' Cell, vol. 172, no. 5, pp. 1122–1131, 2018.

[12] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks. In Proceedings of the 13th ECCV 2014—European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Lecture Notes in Computer Science. Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer; Volume 8689, pp. 818–833.

[13] B Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2921–2929. 20.

[14] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. International Conference on Learning Representations, 2015.

[15] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the International conference, Computer Vision Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 618–626.

[16] I. Kim, S. Rajaraman, S. Antani, Visual Interpretation of Convolutional Neural Network Predictions in Classifying Medical Image Modalities, Diagnostics, Vol. 9, Issue 2, April 2019.

[17] J Kim, L. Tran, Retinal Disease Classification from OCT Images Using Deep Learning Algorithms. IEEE-CIBCB 2021, pp. 42-47, Melbourne, Australia, October 2021.

[18] ImageNet, http://www.image-net.com.

[19] https://www.tensorflow.org/

[20] https://keras.io/

[21] N. Altman and M. Krzywinski, "Ensemble Methods: Bagging and Random Forest", Nature Methods, 14, pp.933-934, 2017.