Biomedical Optics EXPRESS

Transfer learning based classification of optical coherence tomography images with diabetic macular edema and dry age-related macular degeneration

S. P. K. KARRI,[1,*] DEBJANI CHAKRABORTY,[2] AND JYOTIRMOY CHATTERJEE[1]

[1]School of Medical Science and Technology, IIT Kharagpur, Kharagpur, India
[2]Department of Mathematics, IIT Kharagpur, Kharagpur, India
[*]pkkarri.mm@iitkgp.ac.in

Abstract: We present an algorithm for identifying retinal pathologies given retinal optical coherence tomography (OCT) images. Our approach fine-tunes a pre-trained convolutional neural network (CNN), GoogLeNet, to improve its prediction capability (compared to random initialization training) and identifies salient responses during prediction to understand learned filter characteristics. We considered a data set containing subjects with diabetic macular edema, or dry age-related macular degeneration, or no pathology. The fine-tuned CNN could effectively identify pathologies in comparison to classical learning. Our algorithm aims to demonstrate that models trained on non-medical images can be fine-tuned for classifying OCT images with limited training data.

OCIS codes: (110.4500) Optical coherence tomography; (070.5010) Pattern recognition; (100.2960) Image analysis; (170.1610) Clinical applications; (170.4470) Ophthalmology.

References and links

1. T. Otani, S. Kishi, and Y. Maruyama, “Patterns of diabetic macular edema with optical coherence tomography,” Am. J. Ophthalmol. **127**(6), 688–693 (1999).
2. W. Drexler and J. G. Fujimoto, “State-of-the-art retinal optical coherence tomography,” Prog. Retin. Eye Res. **27**(1), 45–88 (2008).
3. R. R. Bourne, J. B. Jonas, S. R. Flaxman, J. Keeffe, J. Leasher, K. Naidoo, M. B. Parodi, K. Pesudovs, H. Price, R. A. White, T. Y. Wong, S. Resnikoff, and H. R. Taylor; Vision Loss Expert Group of the Global Burden of Disease Study, “Prevalence and causes of vision loss in high-income countries and in Eastern and Central Europe: 1990-2010,” Br. J. Ophthalmol. **98**(5), 629–638 (2014).
4. P. Romero-Aroca, “Current status in diabetic macular edema treatments,” World J. Diabetes **4**(5), 165–169 (2013).
5. C. B. Rickman, S. Farsiu, C. A. Toth, and M. Klingeborn, “Dry age-related macular degeneration: Mechanisms, therapeutic targets, and imaging dry AMD mechanisms, targets, and imaging,” Investigative Ophthalmol. Vis. Sci. **54**, ORSF68 (2013).
6. T. A. Ciulla, A. G. Amador, and B. Zinman, “Diabetic retinopathy and diabetic macular edema: pathophysiology, screening, and novel therapies,” Diabetes Care **26**(9), 2653–2664 (2003).
7. S. Farsiu, S. J. Chiu, R. V. O’Connell, F. A. Folgar, E. Yuan, J. A. Izatt, and C. A. Toth; Age-Related Eye Disease Study 2 Ancillary Spectral Domain Optical Coherence Tomography Study Group, “Quantitative classification of eyes with and without intermediate age-related macular degeneration using optical coherence tomography,” Ophthalmology **121**(1), 162–172 (2014).
8. G. Gregori, F. Wang, P. J. Rosenfeld, Z. Yehoshua, N. Z. Gregori, B. J. Lujan, C. A. Puliafito, and W. J. Feuer, “Spectral domain optical coherence tomography imaging of drusen in nonexudative age-related macular degeneration,” Ophthalmology **118**(7), 1373–1379 (2011).
9. M. A. Mayer, A. Borsdorf, M. Wagner, J. Hornegger, C. Y. Mardin, and R. P. Tornow, “Wavelet denoising of multiframe optical coherence tomography data,” Biomed. Opt. Express **3**(3), 572–589 (2012).
10. K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” IEEE Trans. Image Process. **16**(8), 2080–2095 (2007).
11. S. J. Chiu, J. A. Izatt, R. V. O’Connell, K. P. Winter, C. A. Toth, and S. Farsiu, “Validated automatic segmentation of AMD pathology including drusen and geographic atrophy in SD-OCT images,” Invest. Ophthalmol. Vis. Sci. **53**(1), 53–61 (2012).

#274354 http://dx.doi.org/10.1364/BOE.8.000579
Journal © 2017 Received 1 Sep 2016; revised 12 Dec 2016; accepted 13 Dec 2016; published 4 Jan 2017

12. P. A. Dufour, L. Ceklic, H. Abdillahi, S. Schröder, S. De Dzanet, U. Wolf-Schnurrbusch, and J. Kowal, "Graph-based multi-surface segmentation of OCT data using trained hard and soft constraints," IEEE Trans. Med. Imaging **32**(3), 531–543 (2013).
13. A. Carass, A. Lang, M. Hauser, P. A. Calabresi, H. S. Ying, and J. L. Prince, "Multiple-object geometric deformable model for segmentation of macular OCT," Biomed. Opt. Express **5**(4), 1062–1074 (2014).
14. P. P. Srinivasan, S. J. Heflin, J. A. Izatt, V. Y. Arshavsky, and S. Farsiu, "Automatic segmentation of up to ten layer boundaries in SD-OCT images of the mouse retina with and without missing layers due to pathology," Biomed. Opt. Express **5**(2), 348–365 (2014).
15. S. J. Chiu, M. J. Allingham, P. S. Mettu, S. W. Cousins, J. A. Izatt, and S. Farsiu, "Kernel regression based segmentation of optical coherence tomography images with diabetic macular edema," Biomed. Opt. Express **6**(4), 1172–1194 (2015).
16. R. Kafieh, H. Rabbani, M. D. Abramoff, and M. Sonka, "Intra-retinal layer segmentation of 3D optical coherence tomography using coarse grained diffusion map," Med. Image Anal. **17**(8), 907–928 (2013).
17. J.-C. Mwanza, J. D. Oakley, D. L. Budenz, R. T. Chang, O. J. Knight, and W. J. Feuer, "Macular ganglion cell-inner plexiform layer: automated detection and thickness reproducibility with spectral domain-optical coherence tomography in glaucoma," Invest. Ophthalmol. Vis. Sci. **52**(11), 8323–8329 (2011).
18. F. Rathke, S. Schmidt, and C. Schnörr, "Probabilistic intra-retinal layer segmentation in 3-D OCT images using global shape regularization," Med. Image Anal. **18**(5), 781–794 (2014).
19. A. Kanamori, M. Nakamura, M. F. Escano, R. Seya, H. Maeda, and A. Negi, "Evaluation of the glaucomatous damage on retinal nerve fiber layer thickness measured by optical coherence tomography," Am. J. Ophthalmol. **135**(4), 513–520 (2003).
20. E. Tátrai, S. Ranganathan, M. Ferencz, D. C. DeBuc, and G. M. Somfai, "Comparison of retinal thickness by Fourier-domain optical coherence tomography and OCT retinal image analysis software segmentation analysis derived from Stratus optical coherence tomography images," J. Biomed. Opt. **16**(5), 056004 (2011).
21. A. Albarrak, F. Coenen, and Y. Zheng, "Age-related Macular Degeneration Identification In Volumetric Optical Coherence Tomography Using Decomposition and Local Feature Extraction," in *The 17th Annual Conference in Medical Image Understanding and Analysis (MIUA)*(2013), pp. 59–64.
22. N. Anantrasirichai, A. Achim, J. E. Morgan, I. Erchova, and L. Nicholson, "SVM-based texture classification in Optical Coherence Tomography," in *2013 IEEE 10th International Symposium on Biomedical Imaging (ISBI)*(2013), pp. 1332–1335.
23. Y. Zhang, B. Zhang, F. Coenen, J. Xiao, and W. Lu, "One-class kernel subspace ensemble for medical image classification," EURASIP J. Adv. Signal Process. **2014**, 1–13 (2014).
24. P. P. Srinivasan, L. A. Kim, P. S. Mettu, S. W. Cousins, G. M. Comer, J. A. Izatt, and S. Farsiu, "Fully automated detection of diabetic macular edema and dry age-related macular degeneration from optical coherence tomography images," Biomed. Opt. Express **5**(10), 3568–3577 (2014).
25. G. Lemaître, M. Rastgoo, J. Massich, C. Y. Cheung, T. Y. Wong, E. Lamoureux, D. Milea, F. Mériaudeau, and D. Sidibé, "Classification of SD-OCT volumes using local binary patterns: Experimental Validation for DME detection," J. Ophthalmol. **2016**, 3298606 (2016).
26. Y. Wang, Y. Zhang, Z. Yao, R. Zhao, and F. Zhou, "Machine learning based detection of age-related macular degeneration (AMD) and diabetic macular edema (DME) from optical coherence tomography (OCT) images," Biomed. Opt. Express **7**(12), 4928–4940 (2016).
27. S. Ding, H. Zhu, W. Jia, and C. Su, "A survey on feature extraction for pattern recognition," Artif. Intell. Rev. **37**(3), 169–180 (2012).
28. A. Coates, H. Lee, and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics* (2011).
29. Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013).
30. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**(7553), 436–444 (2015).
31. M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Weakly supervised object recognition with convolutional neural networks," Adv. Neural Inf. Process. Syst. **2014**, 1717–1724 (2014).
32. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE **86**(11), 2278–2324 (1998).
33. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Adv. Neural Inf. Process. Syst. **2012**, 1097–1105 (2012).
34. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.
35. S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010).
36. G. Mesnil, Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. J. Goodfellow, E. Lavoie, X. Muller, G. Desjardins, D. Warde-Farley, and P. Vincent, "Unsupervised and transfer learning challenge: A deep learning approach," ICML Unsupervised and Transfer Learning **27**, 97–110 (2012).
37. Y. L. Boureau and Y. L. Cun, "Sparse feature learning for deep belief networks," Adv. Neural Inf. Process. Syst. **2008**, 1185–1192 (2008).

38. J. Yosinki, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” Adv. Neural Inf. Process. Syst. **2014**, 3320–3328 (2014).
39. P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” In International Conference on Document Analysis, **3**, 958–963 (2003).
40. N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” J. Mach. Learn. Res. **15**(1), 1929–1958 (2014).
41. X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” *Proc. AISTATS* **9,** 249–256 (2010).
42. S. P. K. Karri, D. Chakraborthi, and J. Chatterjee, “Learning layer-specific edges for segmenting retinal layers with large deformations,” Biomed. Opt. Express **7**(7), 2888–2901 (2016).
43. J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” J. Mach. Learn. Res. **12**, 2121–2159 (2011).

1. Introduction

Spectral domain optical coherence tomography (SD-OCT) has been an integral imaging instrument in ophthalmology for the early diagnosis of pathologies through textural and morphological variations [1, 2]. The current application is aimed at automated classification (with limited training data) of retinal pathologies diabetic macular edema (DME) and age-related macular degeneration (AMD) as they account for the majority of irreversible vision loss subjects in developed and developing countries [3–5]. In the case of DME fluid is accumulated [6] and in the case of dry AMD drusen is deposited resulting in geographic atrophy, thereby incurring the deformation of retinal layers [7, 8]. From the onset of the disease as it progresses, retinal layers deform and detach from subsequent layers resulting in vision loss. Early diagnosis through inspection for layer deformations, the presence of fluids, geographic atrophy, and drusen, is crucial for subduing vision loss. Retinal images of subjects with dry AMD in Fig. 1(b) and DME in Fig. 1(c) with geographic atrophy and fluid, respectively, are shown in Fig. 1 in comparison to an image of the retina of a normal subject in Fig. 1(a). Such visual inspection process has been fragmented into solvable image analysis blocks to reduce the workload of the ophthalmologist and for remote clinical applications. These blocks are aimed at noise removal, layer segmentation, feature quantification (heuristically and intuitive) and classification.

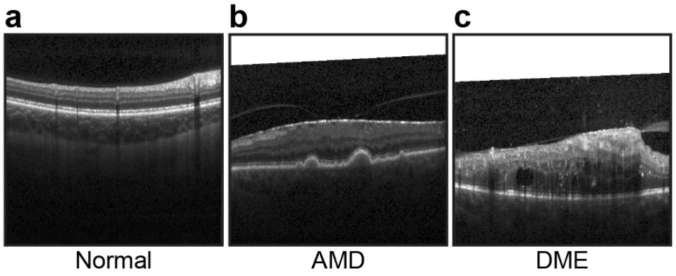

Fig. 1. Layer deformation due to drusen (geographic atrophy) and fluid in dry AMD and DME in comparison to a normal subject‘s retina (the white regions are introduced when exporting from OCT machines to maintain the image size). (a) Normal. (b) AMD. (c) DME.

The noise removal block commonly employs techniques such as wavelets [9] and BM3D [10]. Retinal segmentation algorithms vary from traditional image processing to machine-learning [11–18] of which graph-based approaches implemented over machine-learning predictions have been successful [12, 15, 18]. Information about the segmented layer is utilized to quantify layer deformations to identify the pathology [2,7, 19, 20]. Another stream of algorithms includes the quantification of textural and morphological features to train a classifier to identify the pathology [21–26]. The evolution of salient feature quantification has reduced the dependency of image classification approaches on retinal segmentation; for example, in recent OCT image classification techniques [24] utilizes only segmented retinal pigment epithelium (RPE) layer information for retinal flattening and this has also been considered as a common practice. The proposed approach also utilizes only filtered and flattened images for labeling an unknown retinal OCT image as AMD or DME or normal.

In machine learning the introduction of an additional module of feature extraction [27] between feature quantification and the classifier transforms quantified features to improve the separation of the feature space. Principal Component Analysis (PCA) and neural networks (auto-encoders) are frequently used as feature extractors for enriching the quantified features. With the impact brought by such modules, more effort has been devoted to learning (identifying) data-driven feature quantifiers (from raw data) [28] rather than feature quantifiers that are limited to heuristics (clinical knowledge) and intuition. When multiple of these modules i.e., learnable feature quantifiers (filters) are stacked between raw data (OCT image) and class data (normal, AMD, or DME) it leads to deep learning [29, 30]. Deep learning has had a phenomenal impact on artificial intelligence and machine-learning communities particularly for image processing, language processing, and voice processing applications [30]. Deep learning has emerged as deep neural networks (DNNs), deep belief networks (DBNs), and convolutional neural networks (CNNs), out of which CNNs are widely employed in image analysis applications [31]. In recent years, variants of CNN architectures namely LeNet [32], AlexNet [33], and GoogLeNet [34] have been developed. We selected GoogLeNet for our current application as it established state of the art performance.

A common assumption in machine learning and deep learning is that a large amount of data is required to increase the generalizability of the learnt model. The construction of a model with data-driven feature quantifiers (filters), i.e., a higher order model, on an inadequate amount of data leads to overfitting, which has a negative impact on performance during testing. As this is also the case with classical machine learning, transfer learning [35] has been introduced in which the model is trained on a relative task for which ample data is available and the model parameters are fine-tuned for the required target task [36–38]. Fine-tuning a model trained on general images for retinal optical coherence tomography image classification has not been explored. The proposed approach involves fine-tuning GoogLeNet, which is pre-trained on an ImageNet database, to identify OCT images with pathology. It also includes a preprocessing step for filtering and flattening OCT images, which is not general practice in the deep learning community. The proposed approach has been compared with GoogLeNet trained with random initialization and classical machine learning approaches. Recent training approaches induce randomness or noise to avoid overfitting and improve generalization. Upon introduction of noise the model performance changes from experiment to experiment. Therefore, a proper understanding of the model performance requires repeated experimentation to illustrate the bounds of performance of the trained model.

Collectively proposed approach illustrates the adaptation of GoogLeNet (trained on object classification) for OCT image classification, the employment of less data for training, and finally, automated identification of the potential filter response at each layer.

2. Preamble to GoogLeNet

Classical machine-learning algorithms construct a function (h), which transforms given quantified features (X) to a target space (y) where features representing different classes (DME, AMD, and normal) are easily distinguishable and a rule or boundary is identified to distinguish the classes in the target space. An alternative community has been working on automated feature learning where the feature quantification and transformation function is more data-driven instead of using heuristics (feature engineering) to accommodate more complex functions to minimize intra-class variance and maximize inter-class variance. Rather than learning a complex function in one transformation (layer), deep learning has taken the approach of concatenating multiple transformations in a sequential process such that more abstract features (edges in the first layer, corners and curves in the second layer and so on) are learned as information is processed (learning progresses) from layer to layer (learning progresses). In other words, instead of one complex transformation t(D) (where D represents raw data), deep learning adopts multiple transformations in sequence (t(D) = g(f(k(D)))) and decision boundaries (a single boundary in the case of a binary class problem) as the final layer. Many network architectures have been proposed, e.g., DNN, DBN, and CNN, of which CNN is widely applied in computer vision and image processing applications where the filters at each layer are responsible for transformation at each layer.

Convolutional neural networks (CNN) are more suitable for image applications due to their receptive field property (transformation at any particular pixel is based on information from neighboring pixels). A typical CNN has two phases: forward pass and backward pass. Forward pass involves predicting the output and computing loss (error between the predicted output and actual output). Backward pass involves correcting the learnable filters and weights based on the computed error through a back-propagation algorithm. Backward pass is only employed during the training phase.

CNN architectures for image classification applications involve combinations of the convolution block with learnable filters [39], activation block [33], crossmap local response normalization (LRN) block [33], pooling block [32], fully connected (traditional neural network) block [32] with learnable weights, dropout [40], and loss block [32,33]. Given an image or the responses, the convolution block convolves with a set of filters (learnable), the activation block alters each element in the input based on a user-defined function, the local response normalization acts as regularizer for unbounded activation functions, the pooling block replaces each element based on a statistical operation (maximum or mean) of neighboring elements (image dilation or mean filtering), the fully connected layer is a traditional artificial neural network (ANN), and the loss block computes the error between the predicted class and actual class. The error computed in the loss layer is utilized to correct the filter weights of each convolution block with gradients computed through error back-propagation.

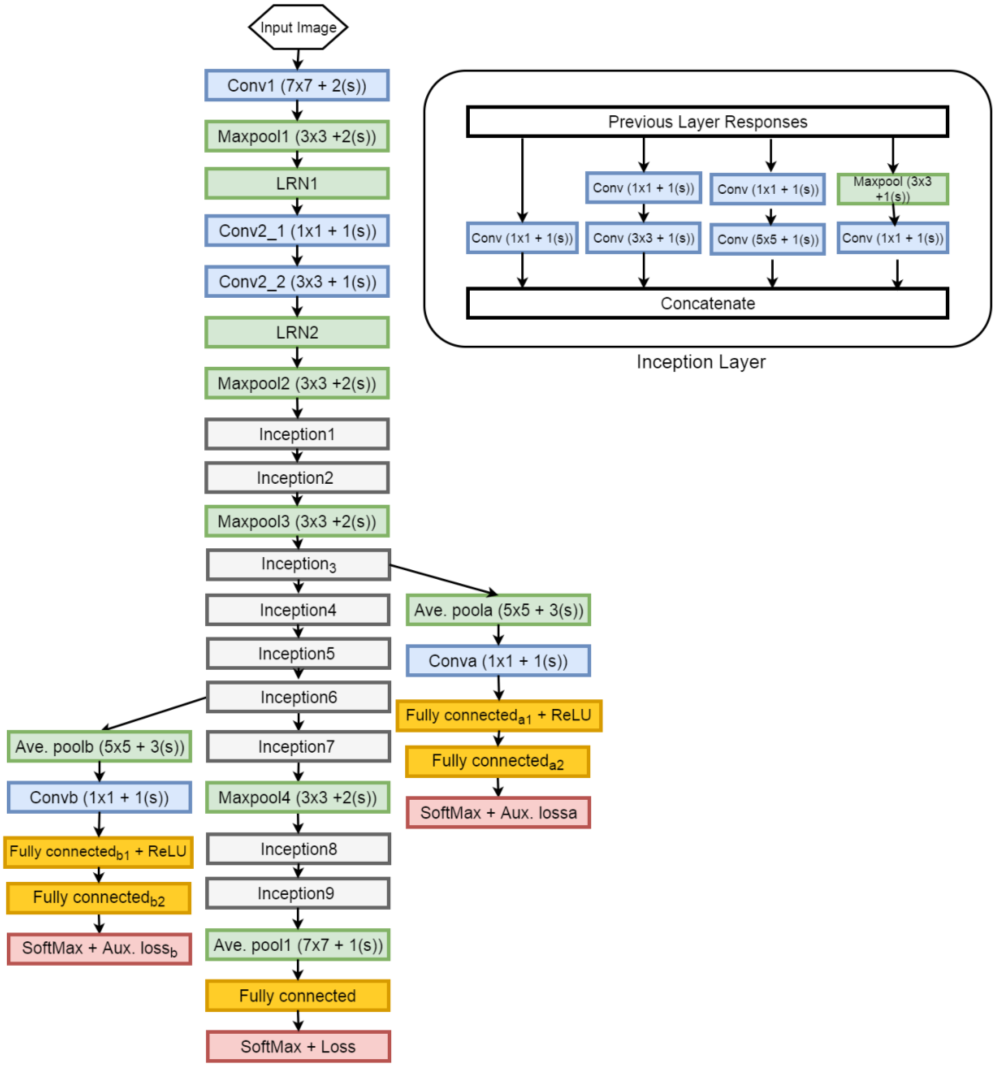

Fig. 2. GoogLeNet with filter dimensions and illustration of inception layer.

The LeNet, GoogLeNet, and ResidualNet neural networks are commonly employed for image classification applications, of which GoogLeNet and ResidualNet have established state-of-the-art performance on the ImageNet challenge (image classification challenge with 1000 classes) in 2014 and 2015. In current applications, GoogLeNet architecture is employed instead of ResidualNet with a 150-layer deep architecture (computationally expensive). Traditional CNN architectures comprise a repetition of convolution blocks (Conv1, Conv2_1, etc.), activation blocks, and the maxpooling blocks (Maxpool1, Maxpool2, etc.), which are appended with multilayer perceptron architecture (ANN) at the end. The salient property of GoogLeNet is an inception layer (Appendix) that introduces sparsity and multiscale (convolving with different filter sizes) information in one block. Functionally it is equivalent to a small network inside a large network. The filters and weights of GoogLeNet are identified iteratively through error back-propagation. The error back-propagation approach converts a loss into gradients to correct the last layer and computes “correcting gradients” of any layer based on (multiplication) 'correcting gradients' of subsequent layers. In the case of deeper architectures, the filters of the initial layers (near the input layer) are not corrected as effectively as the last layer (near the loss layer) and this is termed as vanishing gradient. One additional novelty of GoogLeNet is pacifying the vanishing gradient through multi-headed loss, in contrast to the single loss block in traditional CNN image classification architectures. Thus, auxiliary branches also contribute to the correction of filters in the initial layers. For the current architecture of GoogLeNet, the input image should be of size 224 × 224 with three input channels, as shown in Fig. 2, and on forward pass results in an array of 1000 (1000 classes) after each softmax block. Here it should be noted that all convolution blocks are appended with a Rectified Linear Unit (ReLU) activation block and the dropout layer is appended after full connected, full connection$_{a1}$, and full connection$_{b1}$, but this is not represented in Fig. 2.

3. Method

3.1 Formal definition to the solution

A set of OCT images ($I \in \mathbb{R}^{a{\times}b}$) and corresponding labels ($l \subset \{normal, AMD, DME\}$) are needed as training data. As the adopted GoogLeNet is designed for 224 × 224 images with three channels (color images), all OCT images are preprocessed ($I \in \mathbb{R}^{224\times224}$) and the same plane is concatenated thrice ($I \in \mathbb{R}^{3\times224\times224}$). A mean image ($I_{mean} \in R^{3 \times 224 \times 224}$) is computed based on OCT images from training data and subtracted from the individual image (for data normalization). The GoogLeNet architecture is modified such that the final fully connected layer of both the main and auxiliary branches has three outputs. Before training there are various approaches for initialization of the filter weights: Random initialization, selective random initialization [41], auto-encoder, and transfer learning (initialization based on the weights of a CNN trained on another data set). The first approaches need a large amount of supervised training data (>10,000 images). The second approach needs a large amount of image data (unsupervised) but not labels. The third approach, which needs GoogLeNet trained on an image data set, is best suited to small training data sets (the current case), faster convergence, and high accuracy. To develop an understanding of information processing at each layer, a potential response is identified. This visualization assists a non-expert to understand information processing at each layer and an expert to identify layers that need to be modified in the case of critical errors.

3.2 Preprocessing

Initially, the saturated pixels (i.e., those with an intensity value of 255) in Fig. 3(a) are replaced with an intensity of 10 as shown in Fig. 3(b). The RPE is estimated as proposed in [42], the estimated RPE is smoothened, retinal flattening is performed, and the RPE lower contour is shifted to a fixed position (70% of the height) as shown in Fig. 3(c). The image is resized (bicubic kernel) to 224 × 224 as shown in Fig. 3(d) and filtered through BM3D filter as shown in Fig. 3(e). Figure 3 shows the intermediate results obtained for the preprocessing step with the BM3D result as the output. The BM3D result is replicated three times and each result is treated as channel information.

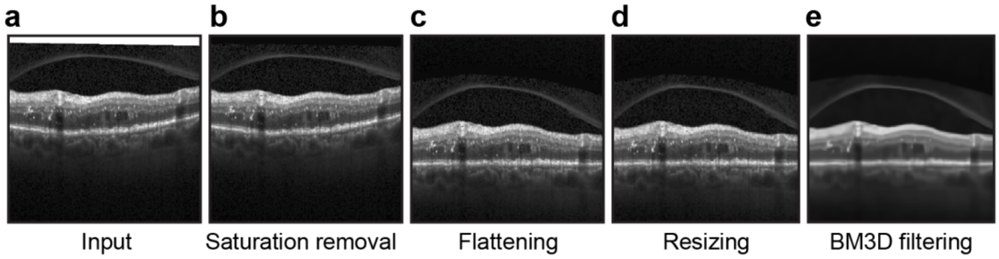

Fig. 3. Illustrating intermediate results of preprocessing. (a) Input image. (b) Saturated pixel removal. (c) Flattening. (d) Resizing. (e) BM3D filtering.

3.3 Modifications to GoogLeNet trained on ImageNet

- Fully connected layer, fully connected layer$_{a2}$, fully connected layer$_{b2}$ have three outputs
- Weights of all fully connected layers and convolutional layers in auxiliary branches are randomly initialized to learn the feature space shifts between ImageNet data and OCT data
- All the last layers have been changed from soft-max to logsoft-max as the soft-max probabilities are low and avoid underflow of gradients.
- The modified GoogLeNet is trained with the adagrad [43] optimizer rather than adam or stochastic gradient descent (SGD), as adam results in impulsive step lengths and SGD needs expertly introduced learning rates at different epochs.

3.4 Identification of potential response

The capability of the fine-tuned CNN is evaluated by considering a test set and computing the prediction error. In a few cases, an expert interprets the filters and responses to identify repetitive filters, missing patterns, etc. This response visualization also helps a non-expert to understand information processing at each layer. Conventional practice involves supervised identification of the potential responses but each layer of a fine-tuned CNN has numerous filters resulting in a large number of responses. The current approach is automated and draws more inspiration from the basic definition of pathology, i.e., deviation from normality. Thus, given a test image and fine-tuned GoogLeNet on prediction of the test image being abnormal (AMD or DME)

- The Pseudo–ground truth for test image is considered normal.
- Error back-propagation computes gradients for correcting each filter in every convolution block.
- In a convolution block, the magnitude of the 'correcting gradients' is proportional to the filter contribution to the prediction.
- In every convolution block all filters are ranked according to the magnitude, and the response of the top filter is treated as the potential response

Based on the capability of human experts, the top-k potential responses can be identified. For simplicity only one response per convolutional layer is identified.

4. Experimentation and results

The Duke OCT data set [24] for classification of ophthalmic images consists of data from 45 subjects (15 AMD, 15 DME, and 15 normal) and multiple images exist for each subject. MATLAB 2015 b is employed to code the benchmark [24] and Torch (http://torch.ch/) is used for training and testing the CNN architectures. Training was conducted on a workstation with 48 GB RAM and a Nvidia tesla k40 GPU. The impact of transfer learning is illustrated by establishing a baseline where weights are randomly initialized. The code for the benchmark, proposed method, and base lines was released (https://github.com/ultrai/Chap_3) for reproducibility.

The training phase involves the following:

- Select 24 subjects (first eight subjects from each class i.e., AMD, DME, and normal) and corresponding images are treated as training images.
- Each training image is preprocessed, self-replicated three times, and concatenated to generate the image to be 3 × 224 × 224.

- Assign images from the same class (AMD, DME, or normal) a single unique label.
- Compute the mean image (3 × 224 × 224) of the training set and subtract the mean image from each image in the training set.
- Divide the training set into batches where each batch contains ≤ 65 images.
- For each batch on a forward pass, CNN predicts 65 labels (one per image) and the negative log-likelihood computes the loss
- On a backward pass, loss is translated to gradients at each layer through a chain rule in error back-propagation to correct the filters and weights.
- The amount of correction required is determined by the AdaGrad optimizer.
- A single iterative cycle involves selecting a subsequent batch, forward pass, backward pass, and correction of filters and weights.

The current setup is trained by dividing the training set into 28 batches, and a collection of 28 iterative cycles is treated as an epoch and training is performed for 50 epochs, i.e., the training data proceeds though the CNN 50 times.

The testing set involves:

- Select the remaining 21 subjects (seven subjects from each class i.e., AMD, DME, and normal), followed by preprocessing each image
- Each image is self-replicated three times and concatenated, thus the resulting test image becomes 3 × 224 × 224.
- Subtract the mean image (3 × 224 × 224) of the training set from each test image.

Fine-tuning GoogLeNet on a defined training set for 50 epochs and computing prediction accuracy on a defined test set (training set and testing sets have no overlap) at every epoch is treated as an experiment. Prediction (test) accuracy computation involves GoogLeNet predicting labels of all test set images and ratio between number of correctly predicted labels and total number of test images. For an experiment, CNN convergence and predictions are subjective to initialization. As fully connected layers are randomly initialized the performance changes from experiment to experiment. Similarly, the dropout block also randomly nullifies (setting to zero) the percentage of responses. This also results in the training model performance changing from experiment to experiment. The randomness that is introduced during model fine-tuning produces variable results for each experimentation; thus, in practice, multiple (10) experiments are repeated and the best model (96%) is stored. The test accuracy of each experiment for every epoch has been illustrated in Fig. 4. Keeping the repeatability in mind, the model with 94% accuracy is considered for evaluations (Table 1).

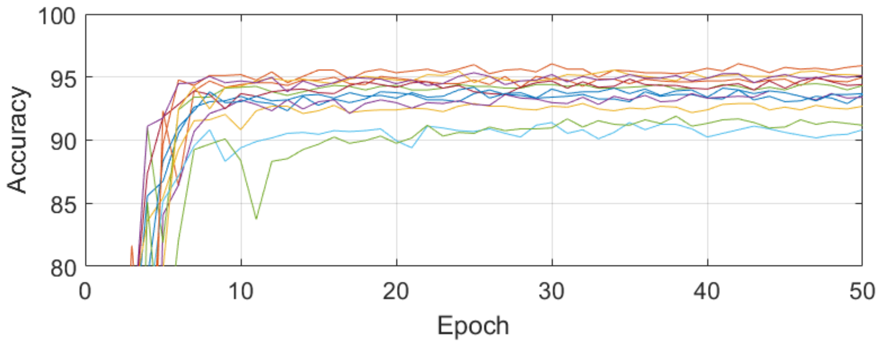

Fig. 4. Repeatability of experimentation in terms of test accuracy also illustrates test accuracy divergence.

The impact of the transfer learning is illustrated by comparison with GoogLeNet trained with random initialization (rather than selective initialization) at all filters. The plots in Fig. 5 present the prediction accuracy on the complete test set (remaining seven subjects) images and illustrates faster convergence with improved accuracy for transfer-learning approach.

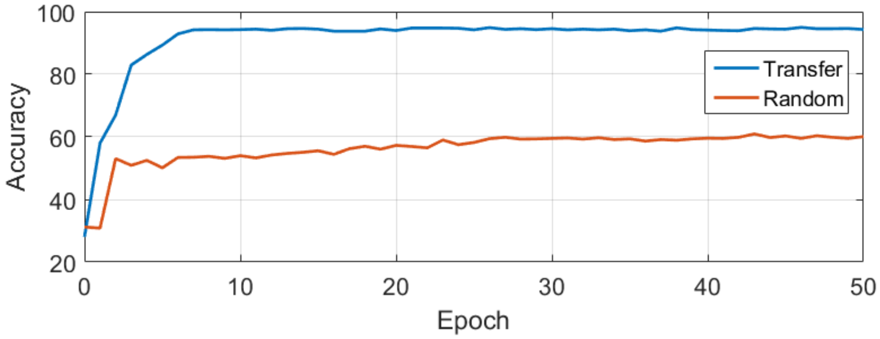

Fig. 5. Test accuracy of fine-tuned GoogLeNet (transfer) in comparison to random initialization.

The benchmark method [24] involves: image resizing, BM3D filtering, RPE estimation, retinal flattening, cropping to remove the background and choroid, and image pyramid construction. The training feature vector is constructed by concatenating the histogram of oriented gradients on each plane of the pyramid and three support vector machines (binary classifier) are trained in a one vs. one approach to accommodate multiclass classification through a binary classifier (SVM). However, to maintain an equal ground for comparison, the resizing, filtering, and RPE flattening are swapped according to the proposed preprocessing protocol.

Given a test set, images of each subject are separated and fine-tuned GoogLeNet predicts the label of each image. Majority voting of all predicted labels for each subject is treated as decision for the corresponding subject [24]. If the predicted decision class and actual patient class are the same then the decision pooling for the subject is '1'. Decision pooling per subject for the proposed method and benchmark is tabulated in Table 1. As the initial eight subjects are considered for training, the test set contains subjects 9 to 15; hence, the corresponding decision poolings are tabulated. It can be observed from the decision pooling that subject decisions are correct but the model is not 100% accurate. The benchmark employs an SVM, which has improved generalization capability with a smaller number of training samples. This has resulted in moderate performance. The fine-tuned model performs more accurately, particularly in comparison with the decisions relating to subjects 9 and 10.

Table 1. Decision pooling evaluation across proposed, random initialization, and benchmark

	Proposed			Random			Benchmark		
	Nor.	AMD	DME	Nor.	AMD	DME	Nor.	AMD	DME
Subject 9	1	1	1	1	1	0	0	1	0
Subject 10	1	1	1	1	1	1	1	1	0
Subject 11	1	1	1	1	0	1	1	1	1
Subject 12	1	1	1	1	0	1	1	1	1
Subject 13	1	1	1	1	0	0	1	1	1
Subject 14	1	1	1	1	0	1	1	1	1
Subject 15	1	1	1	0	1	0	1	1	1

“0” and “1” imply incorrect and correct decisions, respectively, during testing.

Cross validation involves division of a data set into 15 folds, with each fold containing three subjects (one from each class). Each validation involves eight folds for training and seven folds for testing. Folds are sequentially chosen instead of randomly. The first validation includes folds 1-8 as training set and folds 9-15 as the testing test, the second validation includes fold 2-9 as the training set and folds 10-15 along with fold 1 as the test set and so on. This results in 15 validations. The mean of decision pooling across all validations for normal, AMD, and DME are 0.99, 0.89, and 0.86, respectively, for the proposed method.

5. Discussion

Deep learning has been successful in automatically crafting the feature quantifiers subjective to an image analysis application. As such, algorithms require an ample amount of data and higher number of epochs for convergence. Barriers such as these are avoided by fine-tuning models trained on similar data sets for various applications and this approach has proven to be successful. Transfer-learning-based image classification or segmentation for medical images mainly involves microscopic and CT images of which the underlying imaging distributions are related to natural images. A current application thereof involves fine-tuning models for OCT images of which the imaging physics incorporates speckles that are not similar to natural images.

In terms of space complexity there is no difference between fine-tuning an architecture and training an architecture from scratch (random initialization). However, the constraint for fine-tuning an architecture is on changing the filter size or number of filters of a layer in which case subsequent layers need to be trained from scratch. In pre-trained models, the n^{th} layer filters might be responsible for a transformation (h) and subsequent layers are intended for transforming the information transformed by h. Changing the layer configuration changes the transformation (i.e., k instead of h), which enforces subsequent filters to learn a new transformation that can utilize k transformed information. In terms of decision pooling, fine-tuned architectures have outperformed their corresponding architectures trained from scratch. A common rule of thumb exists in deep learning that the deeper the architecture the more accurate the abstractions it can learn; however, no rule has been established regarding the depth of a CNN. Thus, the choice of depth is always user defined and is subjective to accuracy. The depth not only negatively impacts upon space complexity but also on time complexity. Our evaluation clearly illustrated that transfer learning has a definite advantage over learning from scratch. Because CNN is subjective to initialization, the performance varies with different protocols of selective initialization.

Future scope includes incubating the decision pooling inside the architecture; thus, given a stack of images from a subject rather than predicting each image class, the algorithm is expected to predict the subject class. This approach is commonly termed multiple instance learning.

It is common practice to identify a potential filter and corresponding response to visually interpret information processing at each layer. Even though an individual filter cannot quantify the pathology completely, it provides an approximation of a highly contributing filter. This facilitates identification of the areas of congestion of information processing in CNN. Given a 3 × 224 × 224 OCT image with AMD, the first convolution layer alone produces 64 responses; therefore, the potential response at each layer is identified autonomously and

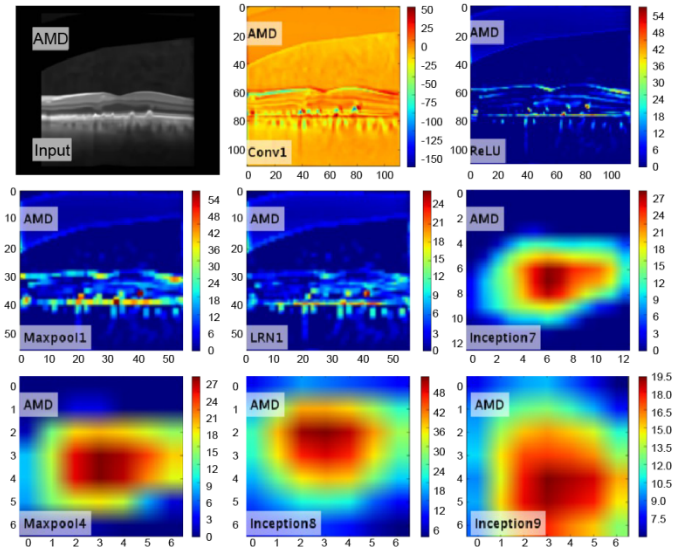

Fig. 6. Identified potential response of conv1, ReLU, Maxpool1, LRN1, Inception7, Maxpool4, Inception8 and Inception9 layers given AMD test image.

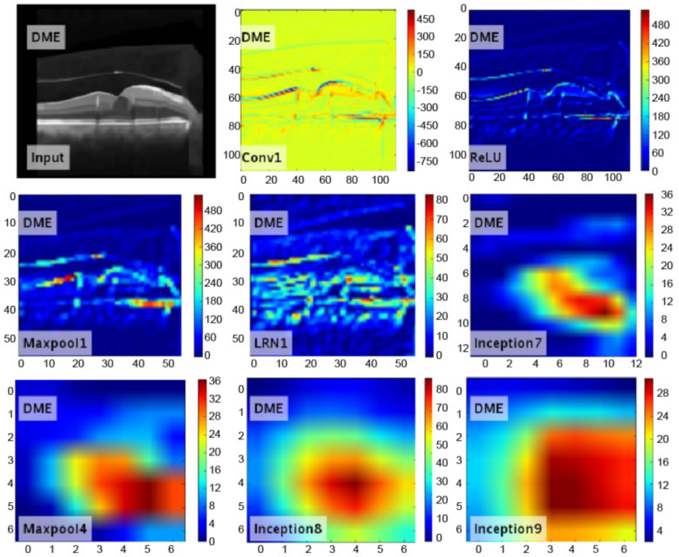

Fig. 7. Identified potential response of conv1, ReLU, Maxpool1, LRN1, Inception7, Maxpool4, Inception8 and Inception9 layers given DME test image.

are shown in Fig. 6. The Conv1 layer (potential filter) identifies the filter that quantizes curves, which incidentally happens to identify the deformations in the RPE layer along with other subtle information. The ReLU only retains positive responses. Maxpooling induces spatial invariance (local) by allotting the maximum response to all neighboring pixels and stride results to reduce the image size. The LRN1 layer is renormalized to boost the responses. The potential responses of the last layers are illustrated as they are crucial for classification. The potential responses of inceptions 7, 8, and 9 include abstract patterns, which are difficult to interpret visually but distinct from the DME case. Given a 3 × 224 × 224 OCT image with DME, the autonomously identified potential response at each layer is shown in Fig. 7. In contrast to those of the AMD subjects, the conv1 potential response cannot illustrate salient markers to identify fluid deposits in the retina. However, as it moves on to the final layers, the potential responses of inception 7, 8, and 9 encode the responses of the previous layers to an abstract pattern that is distinguishable from AMD. One reason for identifying prominent retinal layer deformation instead of fluid deposits by CNN could be that "Layer deformation (Geographical Atrophy)" is a common factor in both pathologies. In the case of AMD, the drusen deposits beneath the RPE, which is hyperreflective, and Conv1 is responsive to RPE, but not at drusen. In the case of DME, the contrast between the intraretinal layers is low; hence, the top retinal layer is more responsive due to its sharper contrast. Inception7 of DME illustrates the prominence as salient feature, which could be fluid deposition but needs further validation by experts. All the potential responses have different dynamic ranges and are depicted in the corresponding sub-figures. All the responses are not transformed into common dynamic responses as few of the responses have a low dynamic range and transformation leads to the loss of visualization contrast. Future work involves the combination of responses into one response for visualization rather than identifying one single representative response.

6. Conclusion

An OCT-based retinal pathological image identification model has been learned for prescreening and remote clinical applications. The conventional approach requires heuristic feature quantifiers and large amounts of data, but the proposed approach learns application subjective feature quantifiers from data. Additionally, another advantage of the proposed approach is the employment of pre-trained models for faster convergence with less data. It has been illustrated that models trained on general images can be fine-tuned to classify OCT images. The method also has the ability to identify the potential response at each layer to understand information processing at each layer.

Appendix building blocks of GoogLeNet

A convolutional block consists of a set of filters and corresponding biases (scalar values) whose values are learned during training. Each filter ($F \in \mathbb{R}^{d \times h \times w}$) height (h) and width (w) are user defined, but the depth (d) is constrained to the output depth of the previous layer. The convolution block involves standard 3D convolution with a pixel shift of one. Given a 3D tuple ($I \in \mathbb{R}^{d \times m \times n}$) concatenation of d number of 2D planes, 3D convolution ($O_i = b_i + I * F_i$) with an i$^{\text{th}}$ filter (F_i) and corresponding bias (b_i) results in 2D response ($O_i \in \mathbb{R}^{floor(m-h) \times floor(n-w)}$); thus, a convolution block with a set of filters results in the same number of 2D responses. These filters and bias act as feature quantifiers within a neighborhood of d×m×n. The filter values are corrected during backward pass. An activation block considers the user-defined function (f ,Rectified Linear Unit in the current case) and transforms each element of the receiving tuple responses. A Rectified Linear Unit ($f(I) = \max(0, I)$) is an unbounded activation function, which means LRN acts as regularizer. On a 2D response plane LRN aims at identifying high responses in comparison to neighboring responses and making it more sensitive. It also pacifies uniform or multiple large responses within a neighborhood. The neighborhood need not be limited to 2D and can be extended to depth (across response planes). A pooling block replaces each element based on a statistical operation (maximum or mean) within a neighborhood for each 2D response plane (image dilation or mean filtering). A fully connected layer is a traditional artificial neural network (ANN) where weights and bias are learned during training. Dropout is an approach where a percentage of responses are set to zero to improve the generalization of the model. The loss block computes the error between the predicted class and actual class. The error computed in the loss layer is utilized to correct each of the convolutional block filter values and fully connected layer weights with gradients computed through error back-propagation.

Stride or shift(s) is a user-defined factor in the convolution and pooling blocks, i.e., after computing convolution or pooling at the (x,y) element, the next convolution or pooling computation is at (x+s,y) or (x,y+s). Thus, given an image or response of size m,n a stride s operation results in approximately m/s,n/s.

The inception layer processes the given information in four independent streams of information processing and concatenates along the depth at the output. The first stream involves compression through a filter with size 1×1 and where the number of filters are less than (data compression) or equivalent to the depth of input. This situation results in a reduced or equal depth of output compared to the input depth but no change along the height and width of the response. The second stream involves two convolution blocks of filter height × width, the size being 1×1 and 3×3, respectively. The initial block performs data compression and the subsequent block is responsible for quantifying the features within a neighborhood with a depth of 3×3. The third stream involves two convolution blocks of filter height × width, the size being 1×1 and 5×5, respectively. As appending convolution blocks of the second and third streams have different neighborhoods they act in a way similar to multi-scale signal processing. The fourth stream involves a maxpooling block and convolution block of filter height × width, the size being 3×3 and 1×1, respectively. A high response at a location on a 2D plane at the convolution block output implies the presence of an important feature at the same location in the input within a neighborhood of convolutional block filters with height and width (other than data compressing convolutional blocks). This implies that the high response is contributed by neighboring elements of the location corresponding to the input. To incorporate this intuition, maxpooling is performed in the fourth stream and this is followed by the convolution block for data compression.

Funding

This study did not receive any funding.

Acknowledgments

The authors would like to express their sincere gratitude to the group of Prof. Sina Farsiu at Duke University for making the OCT classification data set available for research purposes. The authors would also like to thank P. Srinivasan, a former student of the group, for his help with identifying ambiguous images during training.