IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 27, 2021, accepted July 4, 2021, date of publication July 8, 2021, date of current version July 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3095655

Epiretinal Membrane Detection in Optical Coherence Tomography Retinal Images Using Deep Learning

ESTHER PARRA-MORA[1,2], ALEX CAZAÑAS-GORDON[1,2], RUI PROENÇA[3,4], AND LUÍS A. DA SILVA CRUZ[1,2], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
[2]Instituto de Telecomunicações, University of Coimbra, 3030-290 Coimbra, Portugal
[3]Faculty of Medicine, University of Coimbra, 3000-370 Coimbra, Portugal
[4]Centro Cirúrgico de Coimbra, 3045-089 Coimbra, Portugal

Corresponding authors: Esther Parra-Mora (eparra@deec.uc.pt) and Luís A. da Silva Cruz (lcruz@deec.uc.pt)

This work was supported in part by the Secretariat of Higher Education, Science, Technology and Innovation of the Republic of Ecuador, and in part by the Portuguese research funding agency Fundação para a Ciência e a Tecnologia (FCT) under Project UIDB/EEA/50008/2020.

ABSTRACT Epiretinal membrane (ERM) is an eye disease that affects 7% of the world population, with a higher incidence in people over 75 years old. If left untreated, it can lead to complications in the central vision, resulting in severe vision loss. Early detection is important for progress follow-up, treatment monitoring, and to avoid total vision loss. Optical coherence tomography, a non-invasive retina imaging technique, can be used for effective detection and monitoring of this condition. To date, automatic methods to detect ERM have received little attention in the research literature. This article describes the application of deep learning to the automatic detection of ERM. The proposed solution is based on four widely used convolutional neural network architectures adapted to the task using transfer learning, and fine-tuned with a proprietary dataset. The architectures were specialized by optimizing the network hyperparameters and two loss functions, cross-entropy and focal loss. A detailed description of the methods is provided, complemented with an exhaustive evaluation of their performance. Overall, the methods reached an accuracy of 99.7%, with sensitivity and specificity of 99.47% and 99.93%, respectively. The results showed that transfer learning enabled a successful use of deep learning to detect ERM in optical coherence tomography retinal images, even when only relatively small training datasets are available.

INDEX TERMS Artificial intelligence, deep learning, epiretinal membrane, macular puker, neural networks, optical coherence tomography, transfer learning.

I. INTRODUCTION

Epiretinal Membrane (ERM) is an ophthalmic condition that burdens 7% of the total human population, being a major problem in people older than 75 years [1]–[3]. Even though most cases are asymptomatic during the initial stages of the disease, its early detection and treatment can help to delay or avoid a negative evolution. Unfortunately, due to the increasing numbers of elderly people in modern societies, and the relative scarcity of medical ophthalmologists, thorough screening for ERM is not always possible. A solution to this logistic problem involves the use of automatic image analysis techniques in computer-aided diagnosis (CAD) systems tools, to aid non-medical ophthalmology technicians perform pre-clinical screenings.

In recent years, convolutional neural networks (CNNs), an important part of deep learning (DL) techniques, have transformed the landscape of image-based CAD systems. CNNs revolutionized the design of systems based on image classification by learning features directly from sample data, reaching higher levels of classification performance than those of hand-crafted solutions. However, one of the main requirements for these architectures to produce outstanding results is the availability of large number of labeled samples to use in the learning process. This requirement is a major obstacle to the application of CNNs to ophthalmic problems, since the availability of large datasets of retinal images, annotated by experts is limited. To overcome this limitation,

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

transfer learning techniques have become the best option to access the benefits of DL when small amounts of data are available.

Different modalities of retinal images and DL techniques have been extensively used in the literature, mostly on the detection of diabetic retinopathy [4]–[6], detection and assessment of macular edema and fluid accumulation [7]–[9], segmentation of retinal layers [10]–[12] and blood vessels [13]–[15], among others. In contrast, there are few works concerned with automatic ERM detection. A literature review identified some research works that address this problem, [16]–[20]. A common factor to these studies is the use optical coherence tomography (OCT) retinal images as the source of information for the detection algorithm. Regarding methodology, two of the studies apply traditional machine learning approaches by extracting hand-crafted features, two use CNNs, and one presents a comparison between traditional machine learning and deep learning.

While these works advanced the state-of-the-art in ERM detection, some questions or design options were not fully addressed or explored like: (a) the effect of different CNN models and different hyperparameters on the performance of the methods, (b) the use of datasets containing OCT images captured at widely separated times, by different makes and models of OCT scanners, (c) the detection of ERM in images from patients with other eye diseases, (d) the detection of ERM in images previously identified by experienced medical ophthalmologist as difficult to assess, (e) the detection of ERM in images of early-stage ERM cases. Compounding these gaps it was observed that some of the works reviewed provided little or no information about the training process, hardware and software used, as well as hyperparameters and model choice methodologies.

The present work contributes to cover these gaps while furthering the knowledge about the problem at hand and providing a ready to use solution. Trained models are available in a github repository, including documentation to predict whether an OCT B-scan shows sign of ERM. These goals are fulfilled by providing a detailed description of all the steps followed to develop and test the solution proposed.

The major contributions and differentiating aspects of this research work are as follows:

- Use dataset with image mix close to real-life clinical data: Inclusion of different stages of ERM, from early to advanced stage. Inclusion of many images from patients with more than one condition in the same eye.
- Explore different CNN architectures: four widely used CNN architectures (AlexNet, SqueezeNet, ResNet and VGGNet) were evaluated following transfer-learning and fine-tuning methodologies.
- Thorough architecture optimizations and tests: the CNN based ERM detectors were optimized and their performance assessed following well documented steps with exhaustive data presented and analyzed.

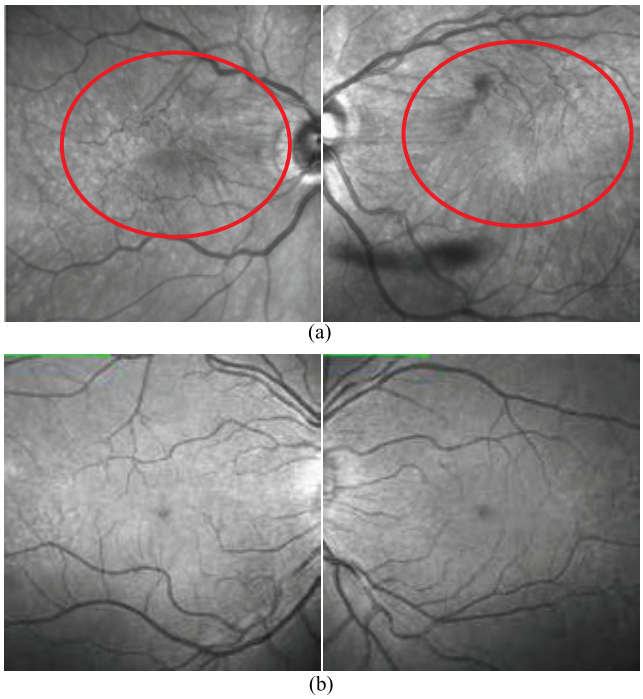

FIGURE 1. (a) Retina with ERM: retinal surface and vasculature is affected by ERM. Red circles indicate the areas where the abnormality can be seen. (b) Normal fundus images.

- Use diversified dataset: dataset images were obtained at different times with different Heidelberg Spectralis and Zeiss Cirrus devices operated by different technicians.
- Evaluated alternative loss functions: besides the commonly used cross-entropy loss function, a weighted loss function designed to down-weigh the impact of the dataset majority class was tested.
- Propose fully evaluated high performance ERM detection algorithms: the central major contribution of this work are the state-of-the-art ERM detection algorithms that were trained and evaluated as described in the next sections.

The remainder of this article is broken down into the following sections. In Section 2, we provide some medical and epidemiological information about ERM, review the techniques used in our experiments and describe the related works found in the literature. Section 3 provides an in-depth explanation of the methods proposed for ERM detection. In section 4 we present and analyze the results of the experiments and discuss the more important findings. Finally, Section 5 concludes the article by summing-up the work and outlining future research explorations.

II. BACKGROUND INFORMATION

This section first provides information about the context of this work describing the underlying problem of ERM disease early detection. It then introduces the concepts to be used in the development of this work and describes summarily the few recent studies found in the literature, that propose ERM

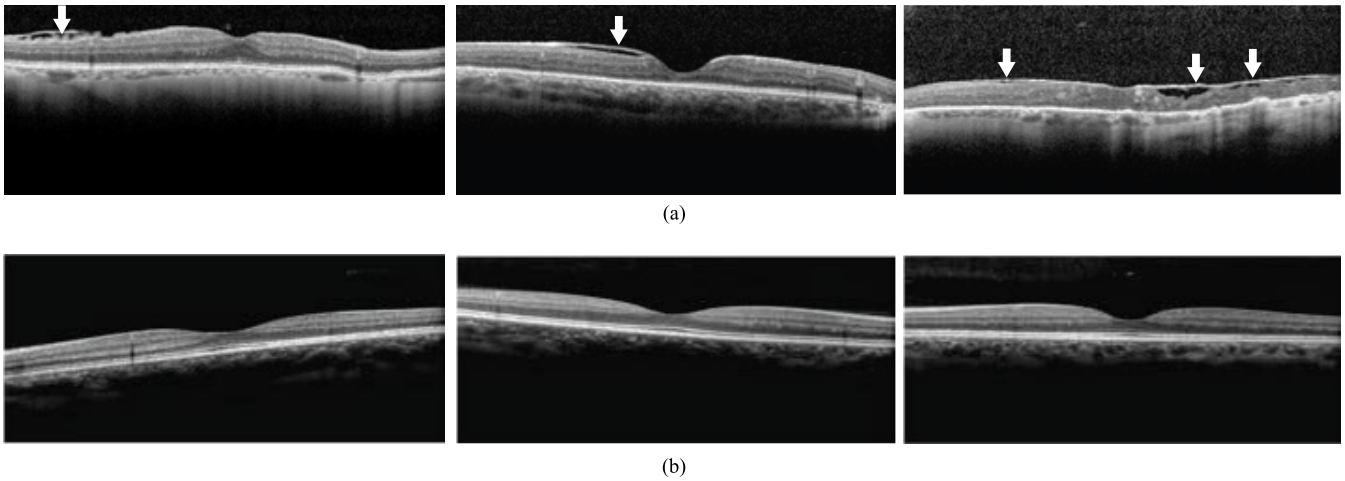

FIGURE 2. Training set retinal OCT B-scans (a) ERM and (b) Normal. Arrows indicate the location of epiretinal membranes.

automatic detection methods designed to operate on OCT images.

A. EPIRETINAL MEMBRANE

Epiretinal Membrane (ERM), also known as macular pucker, consists on the development of a thin layer of glial cells over the central retina or macula. When this membrane grows it can cause retinal surface wrinkling or traction and vascular changes, with subsequent vision alterations [21]. The disease can be caused by trauma or illnesses such as diabetic retinopathy, retinal vein thrombosis, retinal tear, retinal detachment, and posterior vitreous detachment, among others. Fig. 1 presents four fundus retinal images, two of them showing the effects of ERM on the retinal surface.

ERM is a common cause of visual acuity loss and image distortion, a condition known as metamorphopsia. ERM affects 7% of the general population [1], mostly individuals over 50 years old [2]. It is most common in patients above 75, with 20% of patients in this group showing evidence of ERM [3], but when it appears in people over 60, it is usually idiopathic [1].

The symptoms of ERM depend on the location, opacity, contraction magnitude, and degree of distortion that the membrane induces in the retina. Most patients with ERM are asymptomatic, but for some of them, symptoms such as double or blurry central vision, distortion of objects size (macropsia or micropsia), and waves in straight forms may appear. The treatment for ERM is known as vitrectomy with membrane peeling, a surgical intervention that peels off the membrane. After membrane removal, most patients recover the lost vision [3], [22]. Early-stage detection of ERM usually has a good prognosis because it enables timely treatment [23]. An imaging modality that is useful to assess, diagnose, and monitor the development and treatment of ERM is OCT, a non-invasive technology that uses light to generate cross-sectional images of the retina. OCT images allow ophthalmologists to analyze the retinal structure and detect morphological alterations, like changes in thickness and form of the retina and its layers. In an OCT B-scan, ERM is visible as a hyperreflective layer over the internal limiting membrane (ILM) of the retina. This reflective layer usually appears as a non-smooth line and in many cases, the underlying retinal layers present physical distortions. Additionally, cystoid spaces between the membrane and the ILM can be found [3], [22]. Fig. 2a illustrates three examples of retinal OCT images with ERM, where the arrows indicate the sections where the membrane is visible. Fig. 2b shows examples of OCT images with no signs of ERM. These images belong to the dataset used in this research work.

B. CONVOLUTIONAL NEURAL NETWORKS ARCHITECTURES

Convolutional neural networks are signal processing architectures based on convolution operations and neural networks, which can be trained using machine learning (ML) techniques. CNNs are designed to compute color, spatial, and temporal domain features of image and video signals, which are then fed to additional processing layers to classify the inputs or detect certain conditions. CNNs have proven to be effective for visual recognition tasks [24]–[26], autonomous driving [27], among many others. CNNs are organized as stacks of layers, usually an input layer, multiple hidden layers, an output layer. Hidden layers are several convolutional layers followed by end-of-chain fully connected layers for the classification or regression computation. Fig. 3 shows a basic structure of a CNN, with two convolutional layers, one pooling layer, and one fully connected layer.

The convolutional layers is where most of the processing takes place. These layers extract features by using filters of different sizes which are convolved with the output of a previous layer, as shown in Fig. 4.

In general, the final section of a CNN is a fully connected network, which performs the classification by processing the feature maps produced by the last convolutional layer.

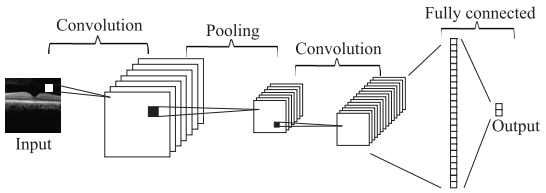

FIGURE 3. Schematic representation of a basic convolutional network.

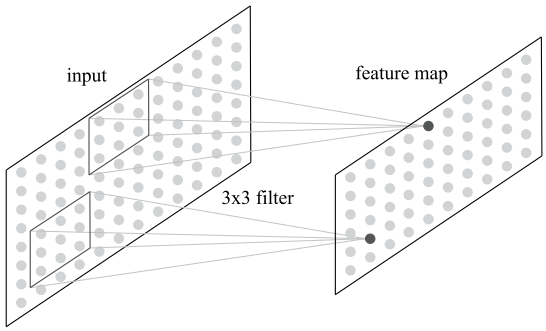

FIGURE 4. Feature computation through 2D convolution.

In between convolutional layers, there can be other types of layers such as pooling and dropout layers. Pooling layers reduce the amount of data to be processed by the network and introduce scale invariance by subsampling intermediate feature maps, while dropout layers avoid overfitting of the network.

Before CNNs, the features used as inputs by classifiers were designed by experts, usually image processing specialists knowledgeable in the classification problem to be solved. In contrast, when using CNNs, during training the network weights and biases of each layer are gradually adjusted to ensure that the collective action of the filters, and other processing operations, extract features that optimize the performance of the classifier. Since the number of network parameters and coefficients is large, it is necessary to use large sets of training data (classification examples in the form of image and class label pairs) to have the training algorithm (usually backpropagation) do its job well. The chain of convolutional layers can be seen as performing a transformation that produces features at different levels of abstraction. As an example, the first layers can detect edges and lines, middle layers round edges and corners that are part of objects, and the upper layers can detect larger parts or complete objects.

The state-of-the-art models for image classification that won the ImageNet Large Scale Visual Recognition Challenges (ILSVRC) from 2012 to 2017 were all based on CNN architectures, a fact that clearly shows the advantages of this type of image classifiers. These models were trained and evaluated on ImageNet, a very large image dataset containing around fourteen million samples manually sorted into about 1000 classes [28], [29].

In this work, we use some of these CNN architectures, AlexNet, SqueezeNet, VGGNet, and ResNet, that due to their good performance have contributed to the popularization of deep learning. AlexNet was introduced by Krizhevsky *et al.* and won the ILSRVC in 2011. The version used in the challenge consisted of 8 layers, 5 of which were convolutional layers followed by the Rectified Linear Unit (ReLU) as activation function, and the last 3 layers were fully connected. In total the network comprises about 60 million parameters. The innovation brought by AlexNet was the use of the non-linear ReLU function as the activation function, instead of the common tanh or sigmoid functions, resulting in faster training [30]. This model marked a milestone in the development of CNNs and was influential to the advances of subsequent deeper networks with better performance.

SqueezeNet is based on AlexNet, but it uses fifty times fewer parameters, a reduction that does not affect performance while speeding up computation. This network is based on *Fire Modules*, which consist of two types of convolutional layers: (i) squeeze layer and (ii) expand layer. The squeeze layer uses 1×1 filters, and the expand layer 1×1 and 3×3 filters. The entire architecture is structured as a convolutional layer at the beginning, followed by 8 Fire modules, ending with convolutional layers instead of fully connected layers [31].

VGGNet is a deeper network architecture presented at the 2014 ILSRVC by the Visual Geometry Group at Oxford. It was the 1st runner-up in that year's competition for classification. The main novelty of this network is the use of small 3×3 filters instead of the 11×11 and 5×5 used in AlexNet. This change decreased the number of parameters in the individual convolutional layers. The two versions, VGG-16 and VGG-19 consisted of 16 and 19 layers, respectively. In both cases, the 3 last layers were fully connected, and the first 13 and 16 layers, respectively, were convolutional [32].

One problem that affects the previous CNNs models is the loss of the first layers' information as we go deeper in the layer stack. To overcome this problem, in 2015, He *et al.* presented ResNet, a very deep network that introduced a cross-layer connection that performs identity mapping. This connection consists in adding the output of a previous layer after one or more (two in the paper) weight layers, instead of having a sequential input/output model. Fig. 5 shows a building block with this type of connection. The model of the entire network consists of several such building blocks interconnected. Depending on the number of convolutional layers, several versions of ResNet have been defined, for instance ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152 which won the ILSVRC in 2015 [33] competition.

Table 1 lists top-5 performing image classification accuracies for some of the most prominent convolutional neural network architectures and respective parameter counts.

C. TRANSFER LEARNING

One of the requirements to get outstanding results from CNNs is the availability of large quantities of data properly labeled. In some cases like medical applications, for a variety of reasons it is not easy to obtain large datasets labeled by

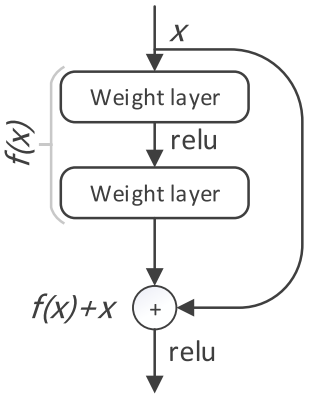

FIGURE 5. Building block [33] of ResNet networks.

TABLE 1. CNNs models top-5 accuracy (adapted from [34]).

	Number of parameters	Top-5 accuracy
AlexNet	60 M	84.70%
ZFNet	–	85.30%
VGG-19	138 M	92.70%
Inception V1 (GoogLeNet)	5M	93.30%
ResNet-101	44.6M	96.40%
ResNet-152	60.3M	93.80%
SqueezeNet	1.2M	80.30%
GoogLeNet V4	–	96.90%

specialists. Transfer Learning is a network adaptation technique that helps to overcome these limitations by using models that were pre-trained with large generic datasets, which are then specialized by fine-tuning to solve a problem from another (related) domain. Besides easing the data availability constraints, transfer learning also reduces the computational cost of training a model. Usually, fine-tuning is applied only to the classifier stages, adapting the models to get the number of classes that the application requires, but it is also possible to adjust the weights and biases in one or more intermediate layers. During this limited retraining process, the model is adapted to the characteristics of the dataset of interest [35].

Several research reports describe the successful application of transfer learning to solve medical diagnosis problems involving automated image analysis, with the most common reason for using transfer learning being the limited access to labeled data [36]–[39]. The advantages of this approach are also reported in some of the works reviewed by Zou *et al.* [40]. This article presents a classification of the methods used in CAD systems for automatic breast cancer diagnosis, and transfer learning has its own subcategory because of its numerous applications. Another example of transfer learning is the work by Liang and Zheng [36], that developed a method to diagnose pneumonia in children, using a self-designed CNN model based in residual blocks. The model was pre-trained in-house with a large dataset comprising 112,120 chest X-ray images belonging to 14 classes. After the pre-training process, it was fine-tuned using 5,856 children chest X-ray scans. The recall value reported was 96.7%.

Hon *et al.* also use transfer learning to classify magnetic resonance imaging (MRI) images to detect Alzheimer's Disease (AD). Two CNN architectures were used, VGG-16 and Inception V4, pre-trained with ImageNet dataset and fine-tuned with the author's own 6,400 specialized image dataset. The accuracy improved from 74.12% when training the VGG-16 model from zero state, to 92.3% when using transfer learning [37]. These and other works show that transfer learning is an important technique that could successfully solve the labeled data availability problems that affect many real-world application areas.

D. CLASS IMBALANCE PROBLEM

In classification problems, it is not uncommon for datasets to have very skewed class distributions. In the medical field and for image-based diagnosis, it is easier to find normal images, or where the disease to be detected is not present than images showing pathology signs. Image datasets, where the number of samples of one class is greater than the number of samples of another class, can be heavily imbalanced. If used in training deep learning models, this imbalance can cause low classification or detection performance. This problem arises because the training process updates the network weights to improve the classification of the majority class, down weighing the minority class classification errors. This problem can have severe negative consequences in the case of medical applications as it can lead to classifiers with high false-negative rates.

The *Focal Loss* is a type of weighted loss function that can be used to reduce the problem of dataset imbalance during the training/fine-tuning process of the models. The operating principle of the focal loss is minimizing the contribution of correctly classified and majority class samples and maximizing that of those erroneously classified and minority class to the loss function value. This loss function was introduced by Lin *et al.* [41], and it is based on the cross-entropy loss function for binary classification. Provided that for binary classification, we defined as p the probability that a sample belongs to the positive class or the class with label equals to 1, the cross-entropy (CE) for binary classification is defined by (1).

$$CE = -log(p_t)$$
$$p_t = \begin{cases} p & \text{if ground truth is 1} \\ 1-p & \text{otherwise} \end{cases} \quad (1)$$

The focal loss adds two factors to the loss function: (i) a modulating factor $(1-p_t)^\gamma$ to reduce the impact of well (easy) classified samples, and (ii) a balancing factor α to increase the contribution of the minority class. The Focal loss is calculated according to (2).

$$FL = \alpha_t(1-p_t)^\gamma CE$$

$$where: \quad \alpha_t = \begin{cases} \alpha & \text{if ground truth is 1} \\ 1-\alpha & \text{otherwise} \end{cases} \tag{2}$$

In [41], the focal loss was used for dense object detection, with good results reported for γ between 0.5 and 5 and α between 0.25 and 0.75, with the best result obtained for $\gamma = 2$ and $\alpha = 0.25$. Other studies have reported improvements in performance when using this weighted loss function. Tran *et al.* [42] describe a classifier of lung nodules using deep learning, reporting an accuracy improvement from 95.6% when using cross-entropy loss to 97.2% when using the focal loss. Similarly, the work published by Al Rahhal *et al.* [43] which evaluated the cross-entropy, over-sampling of the minority class, and the focal loss when using an imbalanced dataset demonstrated that focal loss produced better results.

E. STATE OF THE ART ON EPIRETINAL MEMBRANE AUTOMATIC DETECTION METHODS

Successful application of recent artificial intelligence methods for classification, detection, and segmentation of image and video, has allowed researchers to develop systems for health care applications. Image classification, segmentation, and detection are the most common tasks when analyzing medical images, with the most common application being deciding about the presence or absence of a specific disease. In the context of eye diseases expressed as retinal anomalies, researchers have demonstrated the efficacy of deep learning for the detection of macular edema [39], diabetic retinopathy [4], retinal detachments [44], retinopathy of prematurity [45], among others. These computer-aided systems help health care professionals diagnose and decide starting treatment of these diseases in a timely and efficient manner.

There are not many automatic methods for the detection of ERM in OCT retinal images described in the literature. The relatively few studies available are based on two approaches for feature extraction, traditional and involving deep learning. Baamonde *et al.* [16], [18] and Fang *et al.* [46] reported methods based on conventional feature extraction and machine learning. Lo *et al.* [20] and Lu *et al.* [17] proposed deep learning methods to identify the presence of ERM. Finally, Sonobe *et al.* [19] compared support vector machines (SVM) and deep learning techniques using the reconstructed surface of the retina from OCT images to detect ERM.

Baamonde *et al.* in their two works [16] and [18], formulated the detection of ERM as a classification problem based on manually pre-defined features and machine learning methods. In both studies, the process consisted of three main steps: (i) pre-processing, (ii) feature definition and extraction, and (iii) classification. Image pre-processing involved the definition of the region of interest. The algorithm extracted features reflecting image characteristics like luminosity, texture, contrast, among others, by processing sets of 17×17 [18] or 13×13 [16] pixels centered on ILM pixels. For the classification stage, they used classifiers such as Multilayer Perceptron, Naïve Bayes, K-nearest neighbors, and Random Forests. To train the classifiers of [16] 129 OCT B-scans were used whereas in [18] 285 were used. Reported disease detection accuracies were 91.25% and 89.35%, respectively.

The study by Lu *et al.* [17] applied deep learning techniques to detect four pathologies: cystoid macular edema, serous detachment, ERM, and macular hole. For ERM disease, the dataset consisted of 20,458 OCT images, from which 2,393 images had ERM. The authors trained a convolutional neural network using transfer learning and reported 95.7% of accuracy for ERM detection. However, the experiment is limited to only single-disease images which are rare in clinical practice as most patients are elderly and quite often have other ophthalmic problems. The ERM detection performance on images with multiple diseases is not documented with only the global value of sensitivity being reported.

Similarly, Lo *et al.* [20] proposed a method to detect ERM at a medical specialist level. A ResNet-101 architecture was employed with a dataset comprising 3,618 OCT images (2,171 normal and 1,447 ERM). The authors reported 98.1% accuracy and 0.99 AUROC. The study did not consider the different stages of ERM, and the images that presented inconsistent labeling across annotators were discarded. The authors reported that the classifier of this study failed with images showing early stage ERM signs.

Sonobe *et al.* [19] presented a comparison between the performance of support vector machines and deep learning in the detection of ERM using 3D surfaces reconstructed from OCTs. The inputs for both classifiers were 529 3D reconstructions of the retinal surface. The AUROC of the deep learning model was 0.993 surpassing the AUROC value of 0.988 obtained when using SVM. Unfortunately the authors did not provide any details about hardware, software, and training hyperparameters used in the study.

Taken together, all these studies indicate that the use of DL techniques is appropriate to detect ERM. However, most have some shortcomings: (i) most use images sourced from a single device, (ii) some of them did not perform cross-validation, (iii) none looked into the class imbalance problem, and (iv) all evaluated only one network architecture with one set of hyperparameters, and in some cases, did not report the hyperparameter values. With our work we aimed at improving this state of affairs, by providing more information about the performance of CNNs when applied to ERM detection, fully documenting all the steps followed and analyzing the results obtained.

III. MATERIALS AND METHODS

We propose to detect ERM on OCT B-scans using CNNs constructed using a transfer-learning approach. Our method fine-tunes CNNs pre-trained in the ImageNet dataset to train a classifier with an imbalanced proprietary dataset. The detection problem is modeled as a binary classification task, where the positive and negative classes denote the presence or absence of ERM, respectively. Four

TABLE 2. Dataset composition and size.

	Training set		*Testing set*	
Size	**ERM**	**Not ERM**	**ERM**	**Not ERM**
1024x496 pixels	360	1200	300	300

different pre-trained CNN architectures were tested: AlexNet, SqueezeNet, ResNet, and VGGNet, with different sets of hyperparameters. The next sub-sections describe the dataset, the algorithm's development, testing methodology, and the experiments conducted to evaluate the final classifiers.

A. DATASET

For the development of the present work, anonymized macula-centered spectral-domain OCT (SD-OCT) B-scans images were acquired at a local clinic. All OCT images in the dataset were captured by experienced operators using several Heidelberg Spectralis or Zeiss Cirrus devices over a large period of years.

We used a total of 2,160 B-scans from 608 patients. The dataset was divided into a training subset consisting of 1,560 B-scans, and a testing subset with 600 B-scans. The patients in the training set are different from the ones in the test set. The retinal images were labeled by medical ophthalmology specialists as showing signs of membrane or not and assigned to ERM and not-ERM classes, respectively. Table 2 shows the breakdown of the dataset into the two classes, showing that 23% of the images in the training dataset are labeled as ERM and the remaining 77% as not-ERM, meaning that the dataset is mildly imbalanced.

Train and test images were chosen to cover different cases of ERM, based on the characterization provided by [21], in which ERM cases are divided into four stages as shown in Table 3. Moreover, the dataset included images with ERM and additional abnormalities such as macular edema, and intraretinal or subretinal fluid, and images that are difficult to classify even by human specialists due to noise, lack of image definition, or simply because membrane development is in its very early stages. Fig. 6 shows samples with ERM taken from the dataset at the different development stages, and Fig.7 illustrates images difficult to classify.

All the images were provided by Centro Cirúrgico de Coimbra (CCC) and their use in this work was approved by the Ethics Committee of CCC contingent on the use of anonymity and secure storage measures.

B. CLASSIFIER ARCHITECTURE

After a thorough and careful study of the performance of the many CNN architectures described in the literature, AlexNet, SqueezeNet, ResNet, and VGGNet were evaluated as the possible basis for designing a classifier able to assign OCT B-Scan retinal images into either the ERM or the not-ERM class. All deep learning and ancillary processing were performed using python-based PyTorch (v1.5) framework [47]. The four architectures provided by PyTorch were originally

TABLE 3. ERM stages according to [21].

Stage	Characteristics
1	Mild ERM Foveal depression preserved Retinal layers well defined
2	Wide outer nuclear layer Foveal depression absent Retinal layers well defined
3	Ectopic inner foveal layers Foveal depression absent Retinal layers well defined
4	Ectopic inner foveal layers Foveal depression absent Thick and undefined retinal layers

trained to classify input images into a set of 1,000 different classes. Since in our application we have two classes, ERM and not-ERM, the last layer of each network was modified to have two outputs, corresponding to the desired two classes.

Pytorch implementation of the architectures follows the specifications briefly mentioned in section II-B. AlexNet consists of five convolutional layers for feature extraction and three fully connected layers for classification, as illustrated in Fig. 8. For this and the following architectures, we specify the kernel size in case of convolutional layers; and the input size, and the output size in the case of fully connected layers. SqueezeNet starts with a convolutional layer, followed by 8 fire modules for feature extraction, and at the end a convolutional layer for classification. Fig. 9 shows the fire module and the complete architecture of SqueezeNet. For VGGNet, we used the 19 layers implementation, from which 16 are convolutional for feature extraction and 3 fully connected for classification (Fig. 10). Lastly, we used the 101 version of ResNet, with 100 convolutional layers. The basic building block (BB) for this network is shown in Fig. 11. ResNet101 starts with a convolutional layer with a 7×7 kernel, the next layers are divided into 4 stages: (i) stage 1 with 3 BB, (ii) stage 2 with 4 BB, (iii) stage 3 with 23 BB, and (iv) stage with 3 BB. At the end of the network, a fully connected layer receives 2048 features and outputs 2 values for later use by a softmax function.

All these models were pre-trained using the dataset ImageNet. Since all these CNNs expect the input images to have resolutions equal to 224 by 224 pixels, and as mentioned in Table 2, our images are 1024 x 496 pixels, the dataset images were resized to the target resolution using bilinear interpolation as implemented in the Image.BILINEAR function from the *pillow* library [48]. The general process of adapting the pre-trained models to our dataset is illustrated in Fig. 12.

The main two steps are: (i) create a new instance of the pre-trained model, and (ii) after resizing the images, fine-tune the model using our training dataset. As a result, we have a model adjusted to classify OCT images into one of two classes: ERM or not-ERM. For all cases, the probabilities that

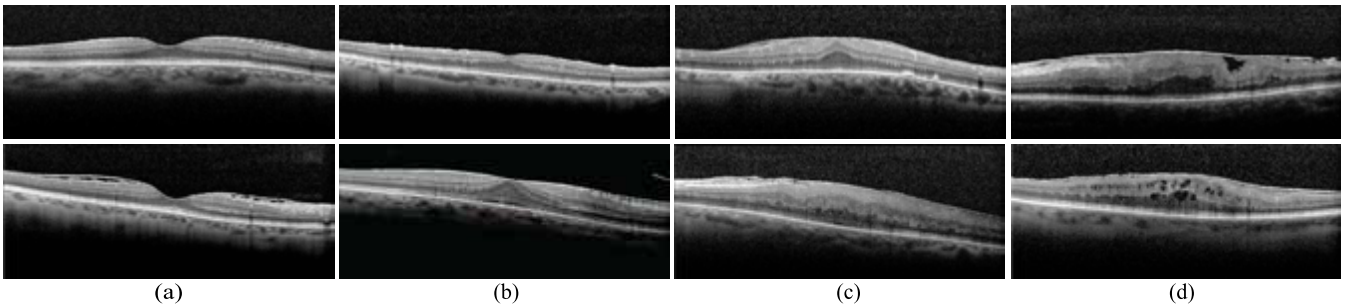

FIGURE 6. Training set B-Scans at different stages of ERM included in the dataset: (a) Stage 1, (b) Stage 2, (c) Stage 3, and (d) Stage 4. Based on the characterization provided by [21].

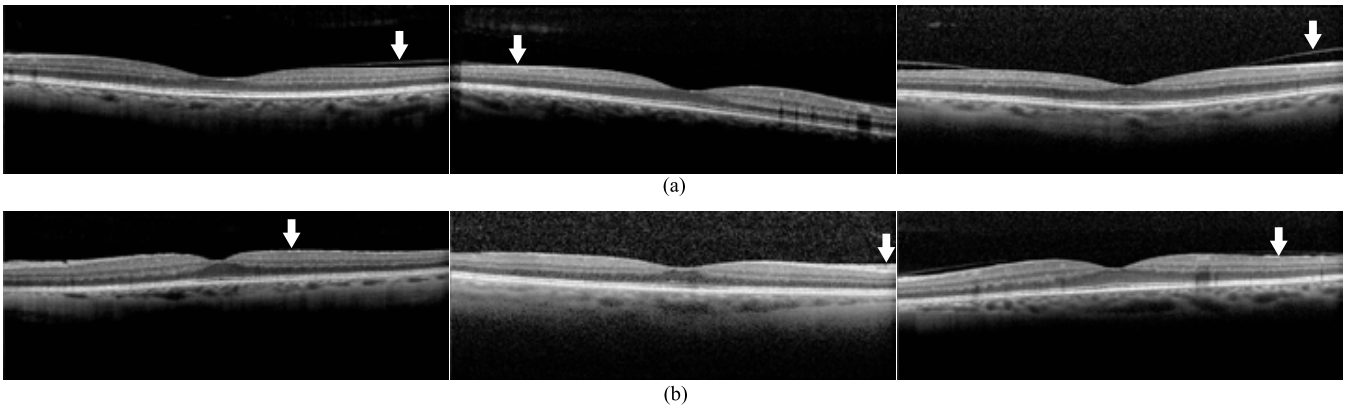

FIGURE 7. Examples of hard to classify OCT retinal images: (a) Normal, arrows indicate the areas that can be confusing for the algorithms, and (b) ERM, arrows indicate where the ERM is located.

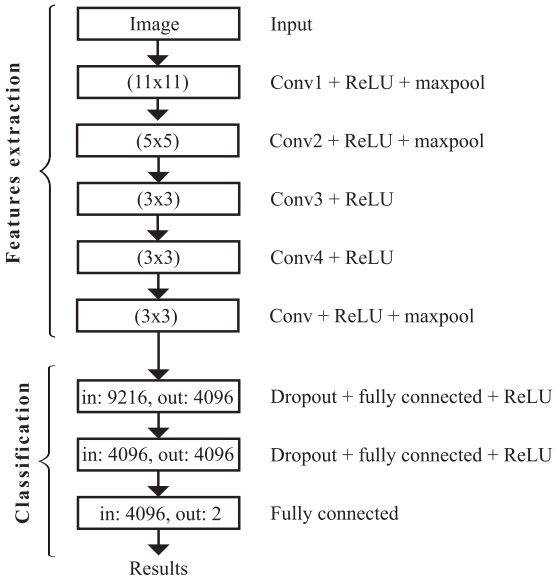

FIGURE 8. AlexNet architecture.

a sample belongs to one class or the other are calculated by applying softmax to the output of the networks.

C. EXPERIMENTAL SETTINGS

1) HYPERPARAMETER OPTIMIZATION

The pre-trained CNNs were fine-tuned with the stochastic gradient descent with momentum (SGDM) algorithm. This optimization algorithm has been demonstrated to be more stable and converge faster than other stochastic-gradient-descent methods [49]. Upon selecting the optimization algorithm, we determined the optimal set of hyperparameters for each network by grid search. The search space included two values of learning rate: 0.001 and 0.0001, two values of momentum: 0.8 and 0.9, and four values of mini-batch size: 8, 16, 32, and 64. Every model was trained for a maximum of 100 epochs with each of the 16 combinations in the hyperparameter grid. The number of epochs was determined empirically in preliminary trials upon recording the number of training steps the models needed to converge. The network parameters were updated to minimize the binary cross-entropy loss function. Table 4 list the hyperparameter values used in the grid search.

2) HANDLING OF THE CLASS IMBALANCE

As it can be seen in Table 2, the distribution of our dataset was skewed towards the negative class. According to the relevant literature, handling the class imbalance has been reported to improve the classification performance [41]–[43]. To verify this assertion, we conducted an experiment in which the pre-trained networks were fine-tuned to optimized a weighted loss function – the focal loss. Besides the hyperparameters of the SGDM algorithm, the focal loss requires setting two more hyperparameters: alpha (α) and gamma (γ). This hyperparameters were also determined by a grid search with four combinations of the following values: (i) α: 0.25 and 0.35, and (ii) γ: 1.5 and 2.0. To find the best set of hyperparameters

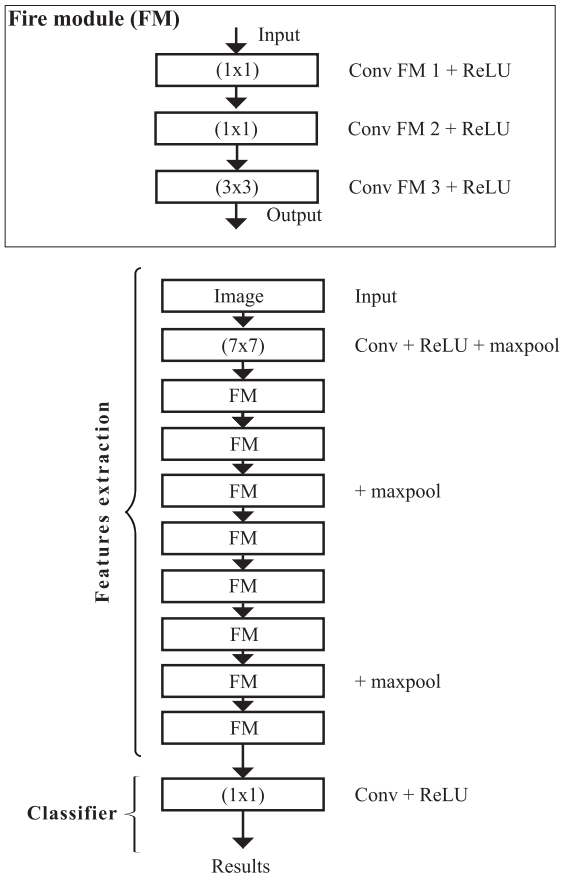

FIGURE 9. SqueezeNet architecture.

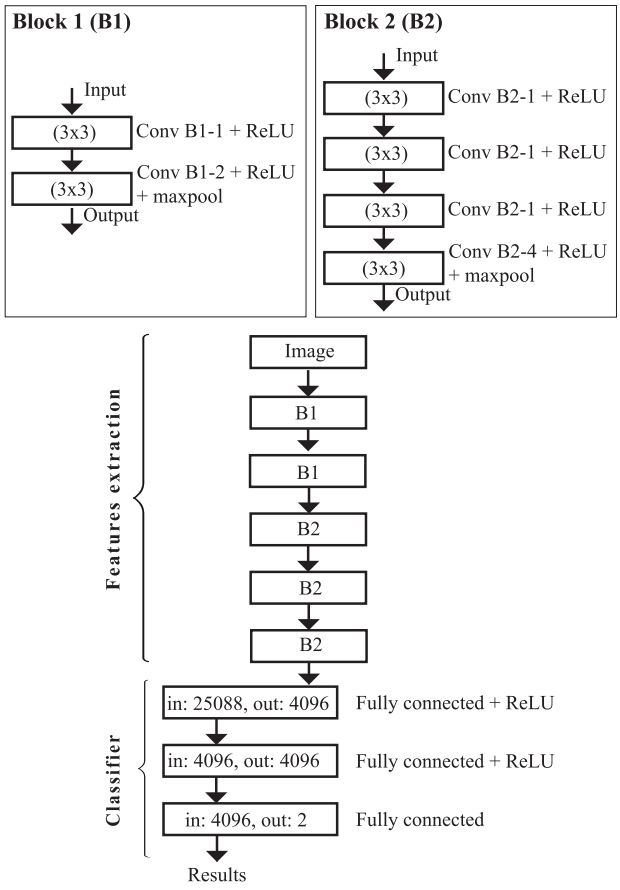

FIGURE 10. VGG-19 architecture.

we fine-tuned each of the four pre-trained CNNs with the best set of hyperparameters of the experiment with the binary-cross-entropy loss and all the combinations of focal-loss hyperparameters.

3) FULLY TRAINING THE MODELS

In this study, we took advantage of pre-trained models to train classifiers for detecting ERM in retinal OCT images using a relatively small dataset. We chose this approach based on the increasing empirical evidence about the use of transfer learning to train deep learning models with limited annotated data. To objectively assess the impact of transfer learning on the classification performance, we initialized the pre-trained models with random weights and trained them with the best set of hyperparameters found in the hyperparameter-optimization experiments. To initialized the network weights we used the Glorot initializer [50]. The network parameters were updated with the SGDM algorithm and the loss function was the binary cross-entropy.

D. HARDWARE AND SOFTWARE

For the development of the classifiers in this work, we used two desktop workstations with the following configurations:

- GPU: NVIDIA GeForce GTX 1070 8GB, CPU: Intel Core i7-8700K, RAM: 32GB, OS: Windows 10

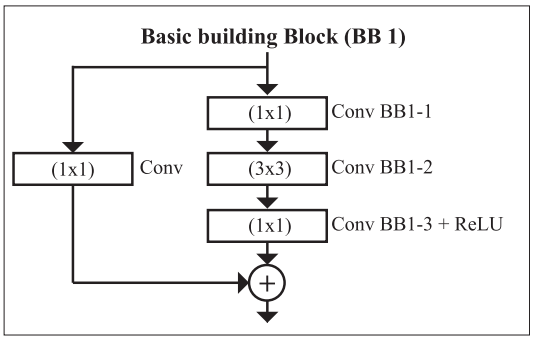

FIGURE 11. ResNet basic building block.

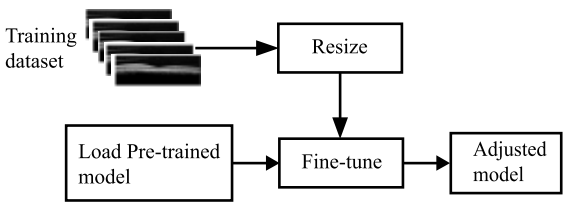

FIGURE 12. Transfer learning general process.

- GPU: NVIDIA GeForce GTX 1080 Ti 16GB, CPU: Intel Core i7-8700K, RAM: 32GB, OS: Windows 10.

TABLE 4. List of hyperparameters values used in the experiments.

Hyperparameter	Value
optimizer	stochastic gradient descent (SGD)
loss function	cross entropy focal loss
mini-batch size	8 16 32 64
learning rate	0.001 0.0001
SGD momentum	0.8 0.9
maximum number of epochs	100

The development framework was based on python version 3.7 and the machine learning package pytorch version 1.5 as well as the helper libraries NumPy version 1.18.3, xlsxwriter version 1.2.8, SciPy version 1.4.1, scikit-learn version 0.22.2.post1, and pandas version 1.1.4.

E. PERFORMANCE EVALUATION

The classification performance of every CNN was evaluated using k-fold cross-validation with k=10. Before fine-tuning the network, the training dataset was divided into 10 partitions each of them with the same number of ERM images (36) and not-ERM images (120). For each of the ten folds, one data partition was designated as the validation set, whereas the other nine were joined into one single training set. During training, we observed the validation loss and saved the model with the lowest loss value. Upon completing all training runs, we obtained a set of ten models $M = \{M_1 \dots M_{10}\}$ which were then applied to the test set to obtain the class-label predictions. Lastly, with the model predictions, we computed the mean classification performance of the CNN.

1) PERFORMANCE METRICS

After fine-tuning, the models were used to classify the images in the test set to evaluate their performance. For a greater insight into the predictive skill of the models, we used several performance metrics, namely: accuracy, sensitivity, specificity. The formal definitions of these metrics is presented below.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{4}$$

$$Specificity = \frac{TN}{TN + FP} \tag{5}$$

where TP is the number true positives, TN is the number of true negatives, FP is the number of false positives, FN is the number of false negatives. The number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) cases were computed based on the predicted and target labels.

To further evaluate the robustness of the classification performance we computed the area under the receiver operating characteristic curve (AUROC). The receiver operating characteristic curve (ROC) is a probability curve that plots the true positive rate (TPR) against the false positive rate (FPR) for different values of threshold, this statistical tool is often used in the analysis of the discriminant capacity of a classification model. AUROC values were calculated using the module metrics of Scikit-learn package [51], which receives the labels and the probability that a sample belongs to the positive class, the built-in function defines the thresholds according to the inputs. In this work we defined the images without signs of ERM as a negative class and the images with ERM as a positive class.

IV. RESULTS AND DISCUSSION

Four CNN architectures and 16 hyperparameter configurations were considered to train a classifier for the task at hand. Every CNN was trained and evaluated using 10-fold cross-validation to estimate the classification performance on unseen data. In total 160 models per architecture were fine-tuned, ten models per combination of hyperparameters. The results of the hyperparameter optimization were summarized by averaging the performance across the ten folds of each set of hyperparameters. Mean and standard deviation values of classification accuracy, sensitivity, specificity, and the AUROC are presented along with corresponding hyperparameters for architectures AlexNet (Table 6), SqueezeNet (Table 7), ResNet (Table 8), and VGGNet (Table 9).

To further evaluate the discriminative capacity of the obtained classifiers, ROC plots were computed using the average of the predicted probabilities of the positive class, i.e. the probability that a given OCT B-scan belongs to the class ERM. Fig. 13 shows the ROC plots of the mean performance for the best configuration of each CNN architecture.

The complexity of classifier training was estimated by measuring the total training time per fold. Inference-time computational complexity was also estimated, by measuring the computation time, per image, during testing. Model complexity was estimated by the amount of memory used. As stated previously, we set to 100 the number of epochs for each fold. Table 5 presents the (average) time in minutes it took to complete one fold training, as well as the time to process a single image from the input to the final decision. The table also includes the size of the models for each network architecture.

Looking at the influence of the hyperparameters on the classification performance, we observed an inverse correlation between the mini-batch size and the classification performance. The larger the mini-batch size the lower the performance, and vice versa. This observation is consistent with prior work regarding the optimization of this hyperparameter [39], [52], and suggests that setting the batch size

TABLE 5. Training complexity (mean training time per fold), inference complexity (testing time for a single image), trained model size (model file size).

	Training time per fold [minutes]	Testing time per image [miliseconds]	Model size [MB]	# of parameters [M]	Accuracy %	Sensitivity %
AlexNet	44	2	217	60.0	98.42	96.87
SqueezeNet	35	4	3	1.2	99.23	98.47
VGGNet	37	11	532	138.0	99.47	98.93
ResNet	95	20	162	44.6	99.70	99.47

to the largest value that fits in memory might constraint the model to settle at a sub-optimal solution. On the other hand, we also verified that the mini-batch size is directly related to the training duration, with smaller mini-batches leading to longer training times. Thus, there is a trade-off controlled by the mini-batch size between the classification performance and the time complexity, where small values of this hyperparameter result in high classification performance but longer training, whereas large mini-batch sizes result in the opposite.

As per the remaining hyperparameters, we observed that in three of the four fine-tuned CNNs, the highest values of accuracy were obtained with the hyperparameters: mini-batch size 8, learning rate 0.001, and momentum 0.9. The exception was VGGNet, for which the best results were obtained using mini-batch size 16 and the same values for the other two parameters. Additionally, for all models, the configuration with the highest accuracy also produced the highest value of sensitivity.

The training process requires large amounts of computational resources and time. To contribute to future works and allow easy performance comparisons with other methods yet to appear, we have published the best models for each architecture in a github repository (`https://github.com/Esther-ParraMora/ERM-detection.git`), along with a detailed explanation of the use of these models.

A. ERM DETECTION PERFORMANCE

Overall, the models developed achieved high classification accuracy, sensitivity, and specificity. The best performing architecture was ResNet with a mean accuracy of 99.7%, mean sensitivity of 99.47%, mean specificity 99.93%, and AUROC value of 1.0. Conversely, the lowest performance was obtained by an AlexNet network with a mean accuracy of 98.42%, mean sensitivity of 96.87%, mean specificity 99.97%, and AUROC value of 1.0. The robustness of the classifiers is further demonstrated by the AUROC values. As it can be seen from tables 6 to 9, the AUROC was consistently high across all hyperparameter combinations, with a minimum of 0.998 and a maximum of 1.00. Similarly, the ROC plots in Fig. 13 show that the models have a good measure of separability between the two classes.

To understand which regions of the images had a strong influence on the decision made by the CNN, gradient-weighted class activation maps (Grad-CAM) [53] were computed and plotted. Fig. 14 shows three examples of images belonging to the ERM class but from cases at different ERM development stages that were correctly classified. Besides the original OCTs, the figure shows their Grad-CAM maps. The regions where ERM is visible were successfully identified in the early-stage case (Fig. 14a), as well as in the more advanced stage case (Fig. 14b). Additionally, we present an image with ERM and macular edema to demonstrate that the decision is heavily based on the part of the image where the membrane is visible (Fig. 14c).

According to the accuracy results and the number of test images, approximately 10 images were misclassified by the network with the lower accuracy (AlexNet), and only one by the best performance network (ResNet). Upon visual inspection of the misclassified images, we found that classification errors correspond to images showing very early stages of ERM which were predicted as not-ERM. Fig. 15 shows three of those images that stand out because they were wrongly classified by more than one network. Fig. 15a was labeled as not-ERM in three of the four architectures, and it is the only ERM sample that was misclassified by the ResNet best model. The other two images, Figures 15b and 15c, failed in two of the four architectures. A possible explanation for this might be that less than 5% of the images in the training dataset show a very early stage ERM. Classifying such images is challenging as they are very similar to images of healthy retinas. Being a problem caused by the under-representation of this type of image, we expect the failure rate to drop by adding more samples of very early stage ERM to the training dataset.

B. WEIGHTED LOSS FUNCTION

Upon determining the optimal set of hyperparameters for fine-tuning the pre-trained CNNs, we sought to improve the classification performance by tuning the models with a weighted loss function. To find the optimal pair of hyperparameters of the weighted loss function, four sets of values were evaluated with each network. Table 10 presents mean and standard deviation values of accuracy, sensitivity, and specificity along with corresponding hyperparameter values in the grid search.

As with the experiment with the binary cross-entropy, we observed a high performance across all hyperparameter sets. However, we did not observe gains in performance due to the weighted loss function. Compared to fine-tuning with the cross-entropy loss, the performance obtained with focal loss

TABLE 6. AlexNet: test set performance metrics. The best result for each mini-batch size value is highlighted with boldface, and the overall best result is underlined.

Mini-batch size	Learning rate	Momentum	AUROC	Accuracy	Sensitivity	Specificity	Epoch
8	0.001	0.8	1.000	0.9822 ± 0.0048	0.9647 ± 0.0093	0.9997 ± 0.0011	29
		0.9	1.000	**0.9842 ± 0.0035**	**0.9687 ± 0.0063**	**0.9997 ± 0.0011**	26
	0.0001	0.8	0.999	0.9798 ± 0.0034	0.9617 ± 0.0059	0.9980 ± 0.0028	52
		0.9	0.999	0.9800 ± 0.0053	0.9653 ± 0.012	0.9947 ± 0.0134	45
16	0.001	0.8	0.999	**0.9838 ± 0.0037**	**0.9677 ± 0.0074**	**1.0 ± 0.0**	28
		0.9	1.000	0.9837 ± 0.0026	0.9673 ± 0.0052	1.0 ± 0.0	27
	0.0001	0.8	0.999	0.9778 ± 0.0061	0.9583 ± 0.0124	0.9973 ± 0.0021	65
		0.9	0.999	0.9790 ± 0.0041	0.9597 ± 0.0076	0.9983 ± 0.0018	50
32	0.001	0.8	0.999	**0.9823 ± 0.0042**	**0.9647 ± 0.0085**	**1.0 ± 0.0**	48
		0.9	0.999	0.9120 ± 0.0139	0.8955 ± 0.0165	1.0 ± 0.0	37
	0.0001	0.8	0.998	0.9740 ± 0.0036	0.9497 ± 0.0071	0.9983 ± 0.0018	75
		0.9	0.998	0.9772 ± 0.0044	0.9580 ± 0.0082	0.9963 ± 0.0019	60
64	0.001	0.8	0.999	0.9802 ± 0.0036	0.9623 ± 0.0072	0.9980 ± 0.0023	49
		0.9	0.999	**0.9828 ± 0.0022**	**0.9663 ± 0.0033**	**0.9993 ± 0.0014**	38
	0.0001	0.8	0.998	0.9738 ± 0.0047	0.9500 ± 0.0102	0.9977 ± 0.0022	91
		0.9	0.998	0.9755 ± 0.004	0.9543 ± 0.0069	0.9967 ± 0.0027	72

TABLE 7. SqueezeNet: test set performance metrics. The best result for each mini-batch size value is highlighted with boldface, and the overall best result is underlined.

Mini-batch size	Learning rate	Momentum	AUROC	Accuracy	Sensitivity	Specificity	Epoch
8	0.001	0.8	1.000	0.9918 ± 0.0036	0.9837 ± 0.0073	1.0 ± 0.0	42
		0.9	1.000	**0.9923 ± 0.0038**	**0.9847 ± 0.0076**	**1.0 ± 0.0**	49
	0.0001	0.8	0.999	0.9828 ± 0.0048	0.9677 ± 0.0119	0.9980 ± 0.0028	55
		0.9	0.999	0.9858 ± 0.0021	0.9733 ± 0.0038	0.9983 ± 0.0053	54
16	0.001	0.8	1.000	0.9878 ± 0.0022	0.9763 ± 0.0055	0.9993 ± 0.0021	39
		0.9	1.000	**0.9905 ± 0.0034**	**0.981 ± 0.0069**	**1.0 ± 0.0**	64
	0.0001	0.8	0.998	0.9812 ± 0.0039	0.9673 ± 0.0081	0.9950 ± 0.0097	69
		0.9	0.999	0.9838 ± 0.0044	0.969 ± 0.011	0.9987 ± 0.0042	61
32	0.001	0.8	1.000	**0.9862 ± 0.0024**	**0.9723 ± 0.0047**	**1.0 ± 0.0**	68
		0.9	0.999	0.9833 ± 0.0035	0.9690 ± 0.0039	0.9977 ± 0.0045	68
	0.0001	0.8	0.998	0.9767 ± 0.0048	0.9593 ± 0.0115	0.9940 ± 0.0058	88
		0.9	0.999	0.9815 ± 0.009	0.9677 ± 0.0166	0.9953 ± 0.0125	72
64	0.001	0.8	1.000	0.9842 ± 0.0029	0.971 ± 0.0097	0.9973 ± 0.0052	72
		0.9	1.000	**0.9867 ± 0.0019**	**0.9753 ± 0.0067**	**0.9980 ± 0.0036**	55
	0.0001	0.8	0.998	0.9823 ± 0.0039	0.969 ± 0.0101	0.9957 ± 0.0055	81
		0.9	0.999	0.9798 ± 0.0061	0.9613 ± 0.0119	0.9983 ± 0.0024	83

was lower overall. This does not discount prior work in class imbalance using weighted loss but suggests that penalizing trivial predictions could be detrimental in mild imbalanced datasets like the one used in this work.

C. FULLY-TRAINING VS. TRANSFER LEARNING

To validate the transfer learning approach used in this work, we fully trained the four architectures, using the same values of hyperparameters that produced the best classifiers described earlier. Table 11 shows a performance comparison of the two approaches. The accuracy improvements when using transfer learning ranged from 3.89% for SqueezeNet to 8.33% for ResNet. The most evident gain is in terms of sensitivity, which ranges between 7.4% for SqueezeNet and 13.4% for ResNet. Transfer learning made possible to improve the probability of correctly classifying positive cases of ERM.

Fig. 16 visualizes some instances of the training process progress for both transfer learning and full training, showing the number of epochs that were necessary to obtain the best results, and the evolution of the key performance indicators with the epoch number. The performance indicators presented are the estimated mean validation accuracy and the validation loss. These plots show that even the performance of the initial network with pre-trained weights is higher than the best performance of the fully trained network. Furthermore, using transfer learning, the architectures converged faster and reached better performances than those obtained when fully training the networks.

TABLE 8. ResNet: test set performance metrics. The best result for each mini-batch size value is highlighted with boldface, and the overall best result is underlined.

Mini-batch size	Learning rate	Momentum	AUROC	Accuracy	Sensitivity	Specificity	Epoch
8	0.001	0.8	1.000	0.9915 ± 0.0076	0.9847 ± 0.0163	0.9983 ± 0.0032	20
		0.9	1.000	**0.9970 ± 0.0041**	**0.9947 ± 0.0086**	**0.9993 ± 0.0014**	50
	0.0001	0.8	1.000	0.9878 ± 0.0054	0.9930 ± 0.0081	0.9827 ± 0.0093	97
		0.9	1.000	0.9908 ± 0.0056	0.9937 ± 0.0078	0.9880 ± 0.008	98
16	0.001	0.8	1.000	**0.9920 ± 0.0071**	**0.9900 ± 0.0137**	**0.9940 ± 0.0052**	62
		0.9	1.000	0.9917 ± 0.0069	0.9893 ± 0.0126	0.9940 ± 0.01	48
	0.0001	0.8	1.000	0.9792 ± 0.0115	0.9653 ± 0.0191	0.9930 ± 0.0067	97
		0.9	1.000	0.9862 ± 0.0052	0.9787 ± 0.0097	0.9937 ± 0.0051	98
32	0.001	0.8	1.000	0.9838 ± 0.0089	0.9737 ± 0.0188	0.9940 ± 0.0049	62
		0.9	1.000	**0.9883 ± 0.0043**	**0.9800 ± 0.0079**	**0.9967 ± 0.0022**	63
	0.0001	0.8	1.000	0.9738 ± 0.0099	0.9567 ± 0.0194	0.9910 ± 0.0039	98
		0.9	1.000	0.9767 ± 0.0082	0.9597 ± 0.0158	0.9937 ± 0.0053	98
64	0.001	0.8	1.000	0.9782 ± 0.0078	0.9627 ± 0.0151	0.9937 ± 0.0037	53
		0.9	1.000	**0.9775 ± 0.0082**	**0.9593 ± 0.0141**	**0.9957 ± 0.0045**	51
	0.0001	0.8	0.999	0.9620 ± 0.0088	0.9323 ± 0.017	0.9917 ± 0.0042	91
		0.9	0.999	0.9683 ± 0.0107	0.9443 ± 0.0212	0.9923 ± 0.0039	84

TABLE 9. VGGNet: test set performance metrics. The best result for each mini-batch size value is highlighted with boldface, and the overall best result is underlined.

Mini-batch size	Learning rate	Momentum	AUROC	Accuracy	Sensitivity	Specificity	Epoch
8	0.001	0.8	1.000	0.9930 ± 0.0038	0.9860 ± 0.0075	1.0 ± 0.0	60
		0.9	1.000	**0.9937 ± 0.0035**	**0.9873 ± 0.007**	**1.0 ± 0.0**	36
	0.0001	0.8	1.000	0.9933 ± 0.0039	0.9867 ± 0.0079	1.0 ± 0.0	65
		0.9	0.999	0.9843 ± 0.0039	0.9733 ± 0.0077	0.9953 ± 0.0126	34
16	0.001	0.8	1.000	0.9878 ± 0.0095	0.9757 ± 0.0189	1.0 ± 0.0	54
		0.9	1.000	**0.9947 ± 0.0026**	**0.9893 ± 0.0052**	**1.0 ± 0.0**	35
	0.0001	0.8	1.000	0.9893 ± 0.0039	0.9787 ± 0.0079	1.0 ± 0.0	33
		0.9	1.000	0.9927 ± 0.0044	0.9853 ± 0.0088	1.0 ± 0.0	55
32	0.001	0.8	0.999	0.9852 ± 0.0061	0.9713 ± 0.0116	0.999 ± 0.0022	35
		0.9	1.000	**0.9893 ± 0.0027**	**0.9787 ± 0.0055**	**1.0 ± 0.0**	51
	0.0001	0.8	0.999	0.9862 ± 0.0053	0.9743 ± 0.0107	0.9980 ± 0.0036	34
		0.9	1.000	0.9887 ± 0.0044	0.9777 ± 0.0082	0.9997 ± 0.0011	44
64	0.001	0.8	0.999	0.9813 ± 0.0116	0.964 ± 0.0238	0.9987 ± 0.0028	37
		0.9	0.999	0.9852 ± 0.004	0.9717 ± 0.0089	0.9987 ± 0.0042	41
	0.0001	0.8	0.999	0.9835 ± 0.003	0.9683 ± 0.005	0.9987 ± 0.0032	36
		0.9	1.000	**0.9853 ± 0.0035**	**0.971 ± 0.0072**	**0.9997 ± 0.0011**	23

D. COMPARISON WITH RELATED WORKS

The development of fully automatic methods for ERM detection in OCT images is a relatively new research problem, and as such only a few studies on this subject have been published in recent years. The common denominator in the state-of-the-art methods is the application of machine learning algorithms including multilayer perceptrons, random forests [16], [18], support vector machines and convolutional neural networks [17], [19], [20]. All these methods were developed and evaluated on private datasets, acquired with different OCT devices, annotated by a variable number of clinicial experts, and gathered following diverse exclusion criteria.

Table 12 presents a comparison of the classification performance of these methods and of the proposed approach. As mentioned, it is necessary to consider that all the competing methods were tested with different datasets and under different experimental settings. Taking this observation into account, we can observe that our method attains performance levels at least as good as that of the state-of-the-art approaches [16]–[20].

Regarding the differentiating aspects of the methodology and novelty of this study, we highlight that, to the best of our knowledge, there is no other work that investigates the computational costs associated to train deep learning models in the context of ERM detection. From the comparative analysis of representative CNN architectures we observed that the incremental computational cost associated with more complex networks does not necessarily result in considerable gains in performance. In fact, in some cases larger computational expenditures might return no performance

TABLE 10. Test set performance metrics using focal loss. The best result for each network architecture is highlighted with boldface, and the overall best result is underlined.

Model	α	γ	epoch	Accuracy	Sensitivity	Specificity
ResNet	0.25	1.5	72	0.9915 ± 0.0051	0.995 ± 0.0063	0.988 ± 0.0082
		2.0	50	0.9895 ± 0.0039	0.993 ± 0.0051	0.986 ± 0.0083
	0.35	1.5	53	**0.9928 ± 0.0043**	**0.9937 ± 0.0088**	**0.992 ± 0.0082**
		2.0	51	0.9885 ± 0.005	0.9883 ± 0.0116	0.9887 ± 0.0059
SqueezeNet	0.25	1.5	33	0.989 ± 0.0045	0.988 ± 0.0079	0.99 ± 0.008
		2.0	29	**0.9895 ± 0.0049**	**0.9887 ± 0.0061**	**0.9903 ± 0.0092**
	0.35	1.5	30	0.9885 ± 0.0034	0.985 ± 0.0067	0.992 ± 0.0055
		2.0	23	0.9867 ± 0.0054	0.9803 ± 0.0102	0.993 ± 0.0048
VGGNet	0.25	1.5	56	0.9828 ± 0.0016	0.968 ± 0.0036	0.9977 ± 0.0032
		2.0	20	**0.9855 ± 0.0044**	**0.973 ± 0.0073**	**0.998 ± 0.0028**
	0.35	1.5	36	0.9855 ± 0.0024	0.971 ± 0.0047	1.0 ± 0.0
		2.0	29	0.985 ± 0.0033	0.9707 ± 0.0066	0.9993 ± 0.0014
AlexNet	0.25	1.5	45	**0.9842 ± 0.004**	**0.969 ± 0.0074**	**0.9993 ± 0.0014**
		2.0	49	0.9828 ± 0.0048	0.968 ± 0.0097	0.9977 ± 0.0032
	0.35	1.5	41	0.9837 ± 0.0028	0.9673 ± 0.0056	1.0 ± 0.0
		2.0	43	0.9828 ± 0.0045	0.9663 ± 0.0094	0.9993 ± 0.0014

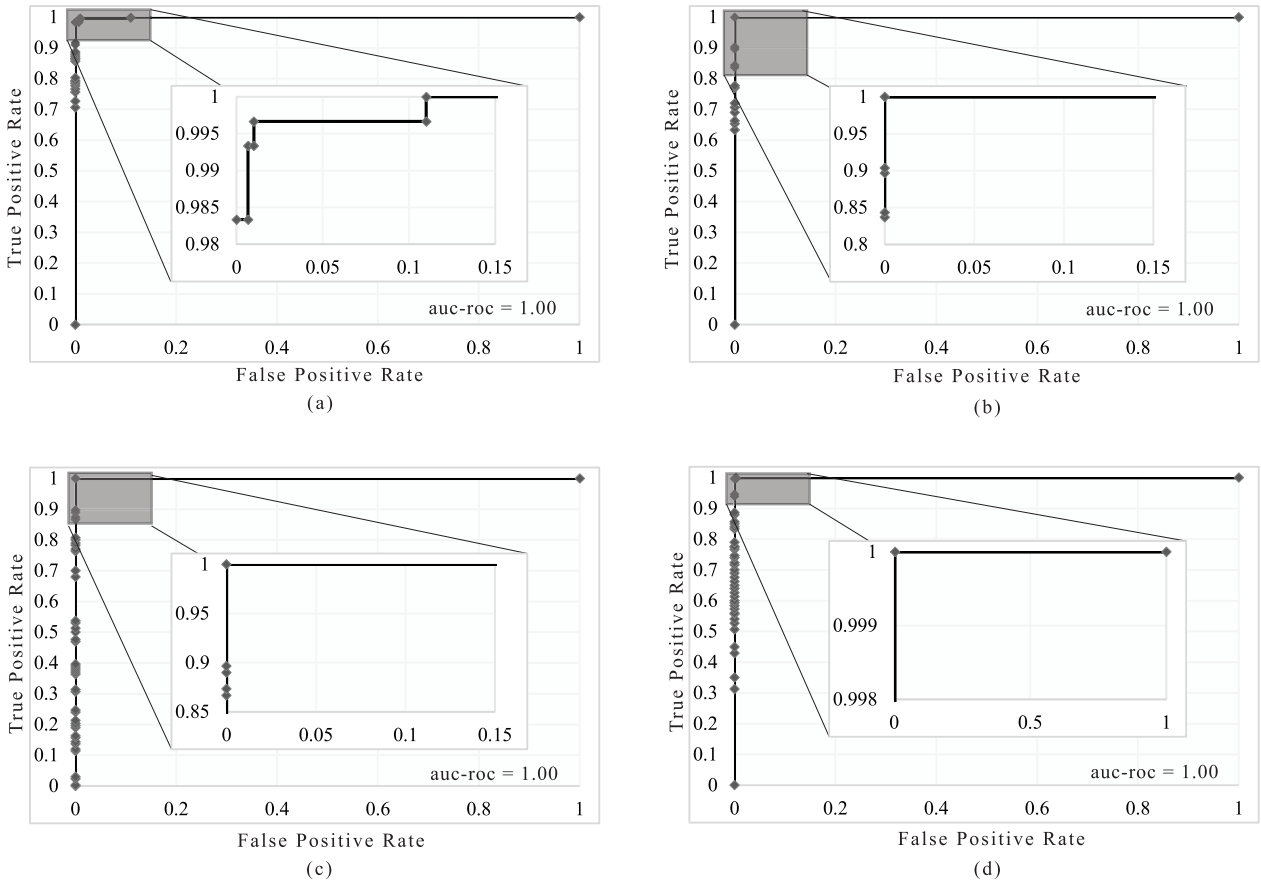

FIGURE 13. Best results ROC-AUC curves: (a) AlexNet, (b) SqueezeNet, (c) ResNet, and (d) VGGNet.

gains whatsoever. This observation is exemplified by the comparison between the top performer in our evaluation (ResNet-101) and the architecture with the lower training time (SqueezeNet). Although the ResNet-101 network is 0.5% more accurate that SqueezeNet, this marginal gain comes at the cost of 50 times more space in disk, and 30 times more memory. On the other hand, chosing SqueezeNet over VGGNet saves almost 100 times memory, and 150 times space in disk, with only a 0.2% loss in classification accuracy (see Table 5).

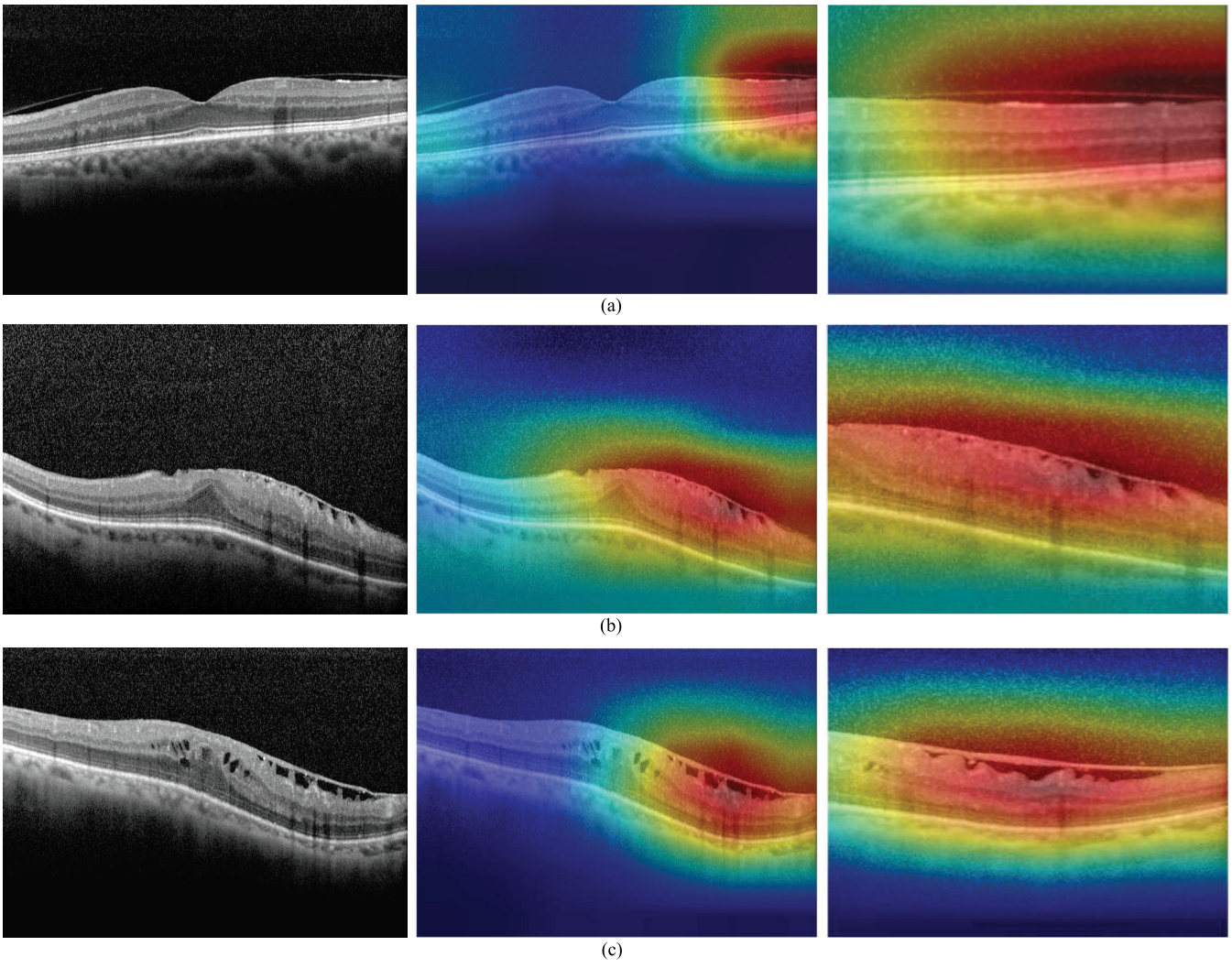

FIGURE 14. Grad-CAM for correctly classified ERM images: (a) early ERM, (b) advanced stages of ERM, and (c) image with macular edema and ERM. The columns represent: first, the original image. Middle column the complete heatmap, and in the last column the zoom of the region of interest.

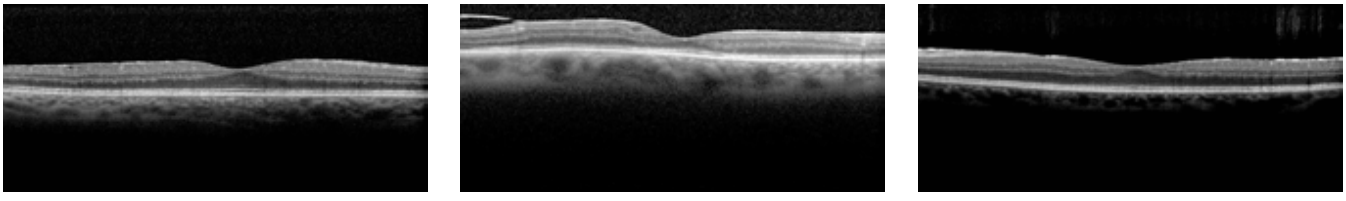

FIGURE 15. Examples of wrongly classified images: (a) 3 out of 4 classifiers failed, (b) and (c) 2 out of 4 classifiers failed.

Besides the impact on the computational cost of the learning process we also investigated and compared the inference time of the classifiers. Once again comparing ResNet-101 and SqueezeNet, as reported in the Table 5, fine tuning the earlier took thrice the time required to train the latter, and the inference time per image of the ResNet-101 was four times longer than that of the SquezeeNet. These observations are very relevant for the development and deployment of CAD systems incorporating deep learning classifiers. Moreover, these findings provide further insight to inform design choices, particularly in low-cost solutions developed for first points of care, or mass screening where the logistics relies in commodity hardware which commonly have modest computational and storage capabilities.

Regarding the robustness of the proposed method, we remark that our approach was evaluated with test data

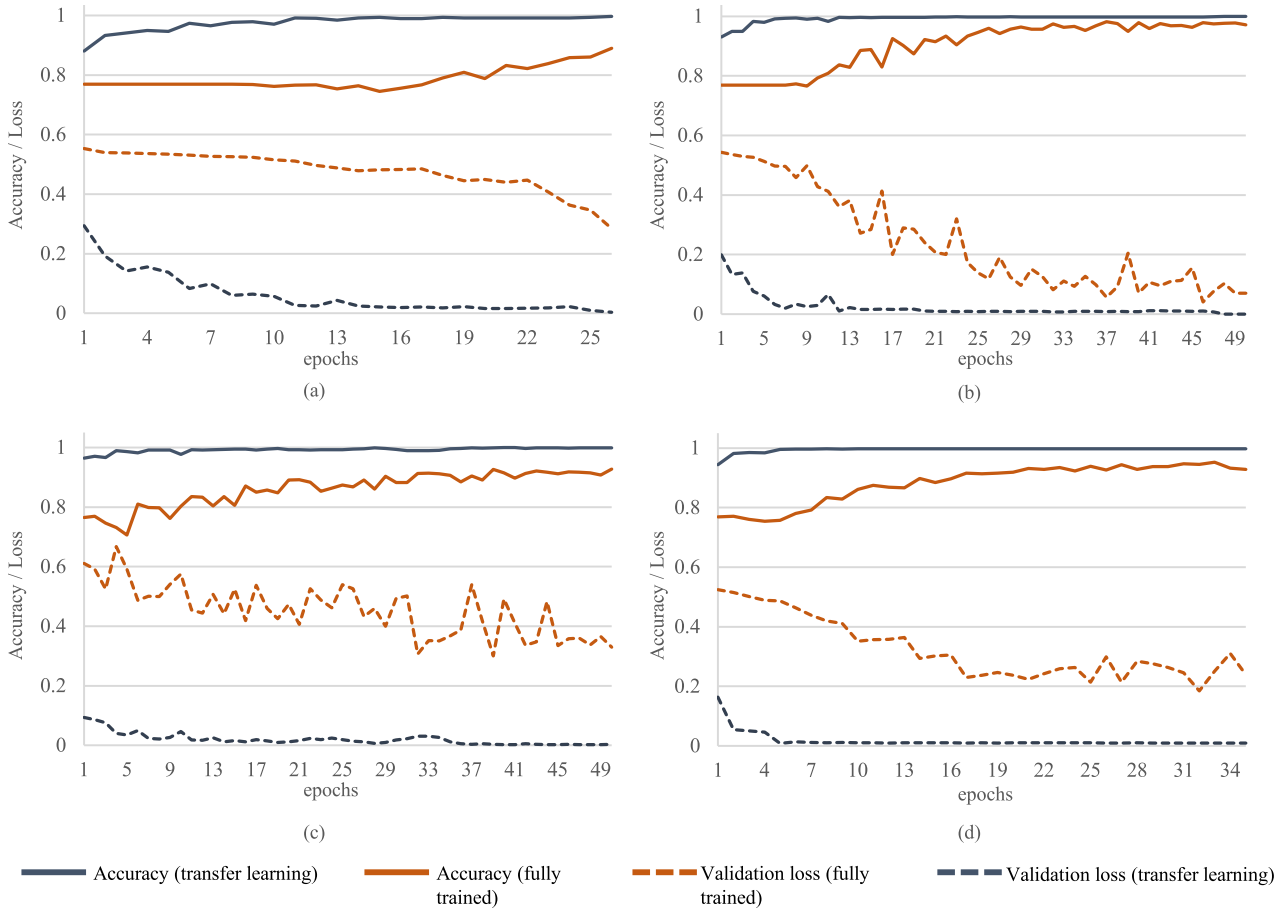

FIGURE 16. Mean validation accuracy and loss value of fully and transfer learning training process. (a) AlexNet, (b) SqueezeNet, (c) ResNet, and (d) VGGNet.

TABLE 11. Comparison between transfer learning (TL) and fully trained (FT) networks.

		Accuracy	Sensitivity	Specificity	epoch
AlexNet	FT	92.11%	87.74%	96.48%	41
	TL	98.42%	96.87%	99.97%	26
SqueezeNet	FT	95.34%	91.07%	99.62%	61
	TL	99.23%	98.47%	100%	49
ResNet	FT	91.37%	86.07%	96.67%	64
	TL	99.7%	99.47%	99.93%	50
VGGNet	FT	93.28%	88.22%	98.33%	42
	TL	99.47%	98.93%	100%	35

TABLE 12. Previous works results for automatic ERM detection methods.

Method	accuracy	sensitivity	specificity
Lu *et al.* [17]	95.7%	–	–
Lo *et al.* [20]	98.1%	98.7%	98.0%
Sonobe *et al.* [19]	97.8% *	97.6%	98.0%
Baamonde *et al.* [16]	91.25%	–	–
Baamonde *et al.* [18]	89.35%	–	–
Ours	99.70%	99.47%	99.93%

* Estimated using the sensitivity, specificity, and number of samples provided in the article

that fits more closely the real-life conditions that clinicians face in practice. We deliberately decided to include hard-to-classify examples, including very early-stage ERM cases and images showing multiple abnormalities. In addition, our dataset includes images acquired using several OCT scanners from two manufactures, Zeiss and Heidelberg. By contrast, related works reported having used exclusion criteria that left out images showing multiple abnormalities [17] and images with disagreements in the manual annotation stage [20]. Despite the heterogeneous nature of the test data, the proposed approach showed consistent high perfomance regardless of the ERM stage (Figures 14a, and 14b) or the presence of multiple lesions (Fig. 14c). Prior works reported low accuracy in detecting early-stage ERM and pointed out that classification errors occurred as a result of class imbalance which was presumably exacerbated by the exclusion criteria of the data collection protocol [17], [20].

The learning approach used in this research work is similar to that of related works in that the classifiers were trained in a supervised fashion. Ideally, supervised learning is conducted with large sets of annotated data to prevent classifiers to

overfit the training data. However, the cost of annotating large sets of data is prohibitive in the medical domain, not only because it is labor-intensive but also because it requires expert knowledge which is in short supply. Consistent with the availability of labeled data in clinical practice, we trained our classifiers with a smaller set of labeled data than related works, but used the weights of pre-trained models to initialize our classifiers thus avoiding the problem of data scarcity. This transfer-learning approach proved to be effective in training deep CNNs without overfitting the training data. Furthermore, a comparative analysis revealed that transfer learning led to faster convergence and higher classification performance as opposed to fully training (see Table 11).

V. CONCLUSION

In this work, we designed a fully automatic method for ERM detection in OCT scans based on CNNs. The proposed algorithm achieved 99.7% detection accuracy when evaluated on a heterogeneous test dataset that includes various stages of ERM and images showing other retinal abnormalities besides the target class. The proposed approach showed high discriminative performance at separating the positive and negative classes. The best classifier from amongst all that were trained and tested attained 99.47% sensitivity, 99.9% specificity, and a AUROC of 1.0. Analysis using Grad-CAM revealed that the proposed algorithm discriminates correctly ERM diagnostic patterns in the OCT scans, such as wrinkling or traction of the retinal surface, or retinal swelling. Taken together, these results show that the proposed method provide a reliable alternative to manual OCT interpretation.

The approach followed builds upon transfer learning and fine-tuning with limited data. Compared to similar works in the literature, our method requires much less annotated data to achieve state-of-the-art performance. It was observed that fine-tuning pre-trained networks as opposed to training from zero state brought significant gains in performance and faster convergence. These findings show that by using available pre-trained CNNs and high quality small datasets one can avoid the need for large volumes of costly annotated data while reaching high classification performance.

Comparative analysis of four CNN architectures showed that in general the more complex the network the higher the classification performance. However, we also observed that the increase in the model complexity does not warrant significant gains in performance. Relative to the architecture with the lowest training time, the best architecture was 0.5% more accurate, but required as much as 40 times more memory, and 50 times more space in disk. Conversely, training the least performing network required 60% less time than training the best performer.

To sum up, the results reported in this article provide complete information of CNN architectures and configurations that produce the best results for ERM detection. Furthermore, we demonstrated that transfer learning makes it possible the use of deep learning techniques even when the size of the dataset is limited.

ACKNOWLEDGMENT

The authors would like to thank Institute for Telecommunications for providing the computational and physical resources needed to develop this work. The authors also wish to acknowledge the support of Centro Cirúrgico de Coimbra and its role in the gathering of the original image dataset used in this work.

REFERENCES

[1] S. Fraser-Bell, M. Guzowski, E. Rochtchina, J. J. Wang, and P. Mitchell, "Five-year cumulative incidence and progression of epiretinal membranes: The blue mountains eye study," *Ophthalmology*, vol. 110, no. 1, pp. 34–40, 2003, doi: 10.1016/s0161-6420(02)01443-4.

[2] S. Mehta. *Epiretinal Membrane—Eye Disorders*. Accessed: Jan. 15, 2021. [Online]. Available: https://www.msdmanuals.com/en-pt/professional/eye-disorders/retinal-dis%orders/epiretinal-membrane

[3] *Epiretinal Membranes—The American Society of Retina Specialists*. Accessed: Jan. 15, 2021. [Online]. Available: https://www.asrs.org/patients/retinal-diseases/19/epiretinal-membranes

[4] R. Gargeya and T. Leng, "Automated identification of diabetic retinopathy using deep learning," *Ophthalmology*, vol. 124, no. 7, pp. 962–969, 2017, doi: 10.1016/j.ophtha.2017.02.008.

[5] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, and R. Kim, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *J. Amer. Med. Assoc.*, vol. 316, no. 22, pp. 2402–2410, 2016, doi: 10.1001/jama.2016.17216.

[6] C. Lam, D. Yi, M. Guo, and T. Lindsey, "Automated detection of diabetic retinopathy using deep learning," *AMIA Summits Transl. Sci.*, vol. 2017, pp. 147–155, May 2018. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/29888061

[7] S. Kaymak and A. Serener, "Automated age-related macular degeneration and diabetic macular edema detection on OCT images using deep learning," in *Proc. IEEE 14th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2018, pp. 265–269, doi: 10.1109/ICCP.2018.8516635.

[8] T. Schlegl, S. M. Waldstein, H. Bogunovic, F. Endstraßer, A. Sadeghipour, A. M. Philip, D. Podkowinski, B. S. Gerendas, G. Langs, and U. Schmidt-Erfurth, "Fully automated detection and quantification of macular fluid in OCT using deep learning," *Ophthalmology*, vol. 125, no. 4, pp. 549–558, 2017, doi: 10.1016/j.ophtha.2017.10.031.

[9] C. S. Lee, A. J. Tyring, N. P. Deruyter, Y. Wu, A. Rokem, and A. Y. Lee, "Deep-learning based, automated segmentation of macular edema in optical coherence tomography," *Biomed. Opt. Exp.*, vol. 8, no. 7, pp. 3440–3448, Jul. 2017, doi: 10.1364/BOE.8.003440.

[10] M. Pekala, N. Joshi, T. Y. A. Liu, N. M. Bressler, D. C. DeBuc, and P. Burlina, "Deep learning based retinal OCT segmentation," *Comput. Biol. Med.*, vol. 114, Nov. 2019, Art. no. 103445, doi: 10.1016/j.compbiomed.2019.103445.

[11] L. Fang, D. Cunefare, C. Wang, R. H. Guymer, S. Li, and S. Farsiu, "Automatic segmentation of nine retinal layer boundaries in OCT images of non-exudative AMD patients using deep learning and graph search," *Biomed. Opt. Exp.*, vol. 8, no. 5, pp. 2732–2744, 2017, doi: 10.1364/BOE.8.002732.

[12] A. Cazanas-Gordon, E. Parra-Mora, and L. A. D. S. Cruz, "Ensemble learning approach to retinal thickness assessment in optical coherence tomography," *IEEE Access*, vol. 9, pp. 67349–67363, 2021, doi: 10.1109/access.2021.3076427.

[13] A. Dasgupta and S. Singh, "A fully convolutional neural network based structured prediction approach towards the retinal vessel segmentation," in *Proc. IEEE 14th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2017, pp. 248–251, doi: 10.1109/ISBI.2017.7950512.

[14] M. Li, Q. Yin, and M. Lu, "Retinal blood vessel segmentation based on multi-scale deep learning," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2018, pp. 1–7.

[15] İ. Atli and O. S. Gedik, "Sine-Net: A fully convolutional deep learning architecture for retinal blood vessel segmentation," *Eng. Sci. Technol., Int. J.*, vol. 24, no. 2, pp. 271–283, Apr. 2021, doi: 10.1016/j.jestch.2020.07.008.

[16] S. Baamonde, J. de Moura, J. Novo, and M. Ortega, "Automatic detection of epiretinal membrane in OCT images by means of local luminosity patterns," in *Advances in Computational Intelligence*. Cham, Switzerland: Springer, 2017, pp. 222–235, doi: 10.1007/978-3-319-59153-7_20.

[17] 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," *Transl. Vis. Sci. Technol.*, vol. 7, no. 6, p. 41, Dec. 2018, doi: 10.1167/tvst.7.6.41.
[18] S. Baamonde, J. de Moura, J. Novo, P. Charlón, and M. Ortega, "Automatic identification and characterization of the epiretinal membrane in OCT images," *Biomed. Opt. Exp.*, vol. 10, no. 8, pp. 4018–4033, 2019, doi: 10.1364/BOE.10.004018.
[19] T. Sonobe, H. Tabuchi, H. Ohsugi, H. Masumoto, N. Ishitobi, S. Morita, H. Enno, and D. Nagasato, "Comparison between support vector machine and deep learning, machine-learning technologies for detecting epiretinal membrane using 3D-OCT," *Int. Ophthalmology*, vol. 39, no. 8, pp. 1871–1877, Aug. 2019, doi: 10.1007/s10792-018-1016-x.
[20] Y.-C. Lo, K.-H. Lin, H. Bair, W. H.-H. Sheu, C.-S. Chang, Y.-C. Shen, and C.-L. Hung, "Epiretinal membrane detection at the ophthalmologist level using deep learning of optical coherence tomography," *Sci. Rep.*, vol. 10, no. 1, p. 8424, Dec. 2020, doi: 10.1038/s41598-020-65405-2.
[21] A. Govetto, R. A. Lalane, D. Sarraf, M. S. Figueroa, and J. P. Hubschman, "Insights into epiretinal membranes: Presence of ectopic inner foveal layers and a new optical coherence tomography staging scheme," *Amer. J. Ophthalmology*, vol. 175, pp. 99–113, Mar. 2017, doi: 10.1016/j.ajo.2016.12.006.
[22] K. T. Oh. *Epiretinal Membrane—Medscape*. Accessed: Jan. 15, 2021. [Online]. Available: https://emedicine.medscape.com/article/1223882
[23] W. Stevenson, C. M. P. Ponce, D. R. Agarwal, R. Gelman, and J. B. Christoforidis, "Epiretinal membrane: Optical coherence tomography-based diagnosis and classification," *Clin. Ophthalmology*, vol. 10, pp. 527–534, Mar. 2016, doi: 10.2147/opth.S97722.
[24] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 1085–1088, doi: 10.1145/2647868.2654970.
[25] Z. Yi, "Evaluation and implementation of convolutional neural networks in image recognition," *J. Phys., Conf. Ser.*, vol. 1087, Sep. 2018, Art. no. 062018, doi: 10.1088/1742-6596/1087/6/062018.
[26] S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," *J. Big Data*, vol. 6, no. 1, pp. 1–18, Dec. 2019, doi: 10.1186/s40537-019-0276-2.
[27] B. Wu, A. Wan, F. Iandola, P. H. Jin, and K. Keutzer, "SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 129–137, doi: 10.1109/CVPRW.2017.60.
[28] *ImageNet*. Accessed: Jan. 15, 2021. [Online]. Available: http://www.image-net.org/
[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and A. C. Berg, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.
[30] 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. [Online]. Available: https://proceedings.neurips.cc/paper/4824-imagenet-classification-with-%deep-convolutional-neural-networks.pdf
[31] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and & MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: http://arxiv.org/abs/1602.07360
[32] 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
[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*. [Online]. Available: http://arxiv.org/abs/1512.03385
[34] *Evolution of CNN Architectures: LeNet, AlexNet, ZFNet, GoogleNet, VGG and ResNet*. Accessed: Feb. 15, 2021. [Online]. Available: https://iq.opengenus.org/evolution-of-cnn-architectures/
[35] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1717–1724, doi: 10.1109/CVPR.2014.222.
[36] G. Liang and L. Zheng, "A transfer learning method with deep residual network for pediatric pneumonia diagnosis," *Comput. Methods Programs Biomed.*, vol. 187, Apr. 2020, Art. no. 104964, doi: 10.1016/j.cmpb.2019.06.023.
[37] M. Hon and N. M. Khan, "Towards Alzheimer's disease classification through transfer learning," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Nov. 2017, pp. 1166–1169, doi: 10.1109/BIBM.2017.8217822.
[38] B. Cheng, M. Liu, D. Shen, Z. Li, and D. Zhang, "Multi-domain transfer learning for early diagnosis of Alzheimer's disease," *Neuroinformatics*, vol. 15, no. 2, pp. 115–132, Apr. 2017, doi: 10.1007/s12021-016-9318-5.
[39] A. Cazanas-Gordon, E. Parra-Mora, and L. A. da Silva Cruz, "Evaluating transfer learning for macular fluid detection with limited data," in *Proc. 28th Eur. Signal Process. Conf. (EUSIPCO)*, Jan. 2021, pp. 1348–1352, doi: 10.23919/Eusipco47968.2020.9287859.
[40] L. Zou, S. Yu, T. Meng, Z. Zhang, X. Liang, and Y. Xie, "A technical review of convolutional neural network-based mammographic breast cancer diagnosis," *Comput. Math. Methods Med.*, vol. 2019, pp. 1–16, Mar. 2019, doi: 10.1155/2019/6509357.
[41] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988. [Online]. Available: https://arxiv.org/pdf/1708.02002v2.pdf
[42] G. S. Tran, T. P. Nghiem, V. T. Nguyen, C. M. Luong, and J.-C. Burie, "Improving accuracy of lung nodule classification using deep learning with focal loss," *J. Healthcare Eng.*, vol. 2019, Feb. 2019, Art. no. 5156416, doi: 10.1155/2019/5156416.
[43] M. M. Al Rahhal, Y. Bazi, H. Almubarak, N. Alajlan, and M. Al Zuair, "Dense convolutional networks with focal loss and image generation for electrocardiogram classification," *IEEE Access*, vol. 7, pp. 182225–182237, 2019, doi: 10.1109/ACCESS.2019.2960116.
[44] H. Ohsugi, H. Tabuchi, H. Enno, and N. Ishitobi, "Accuracy of deep learning, a machine-learning technology, using ultra–wide-field fundus ophthalmoscopy for detecting rhegmatogenous retinal detachment," *Sci. Rep.*, vol. 7, no. 1, pp. 1–4, Dec. 2017, doi: 10.1038/s41598-017-09891-x.
[45] J. M. Brown, J. P. Campbell, and A. Beers, "Automated diagnosis of plus disease in retinopathy of prematurity using deep convolutional neural networks," *JAMA Ophthalmol.*, vol. 136, no. 7, pp. 803–810, Jul. 2018, doi: 10.1001/jamaophthalmol.2018.1934.
[46] L. Fang, L. Yang, S. Li, H. Rabbani, Z. Liu, Q. Peng, and X. Chen, "Automatic detection and recognition of multiple macular lesions in retinal optical coherence tomography images with multi-instance multilabel learning," *J. Biomed. Opt.*, vol. 22, no. 6, Jun. 2017, Art. no. 066014, doi: 10.1117/1.Jbo.22.6.066014.
[47] *PyTorch*. Accessed: Oct. 15, 2020. [Online]. Available: https://pytorch.org/
[48] *Pillow (Pil Fork)*. Accessed: Feb. 15, 2020. [Online]. Available: https://pillow.readthedocs.io/en/stable/reference/ImageEnhance.html
[49] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*. [Online]. Available: http://arxiv.org/abs/1609.04747
[50] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a
[51] *Scikit-Learn Machine Learning in PyThon*. Accessed: Oct. 15, 2020. [Online]. Available: https://scikit-learn.org/stable/
[52] J. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, G. Montavon, G. B. Orr, and K. R. Müller, Eds. Berlin, Germany: Springer, 2012, pp. 437–478.
[53] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626, doi: 10.1109/iccv.2017.74.

ESTHER PARRA-MORA received the bachelor's degree in electronics and information networks from National Polytechnic School, Ecuador, in 2007, and the master's degree in computer science from The University of Queensland, Australia, in 2015. She is currently pursuing the Ph.D. degree with the University of Coimbra, Portugal. Since October 2017, she has been a Researcher with the University of Coimbra. Her research interests include automatic diagnosis of retinal diseases using deep learning techniques and different modalities of retinal images.

ALEX CAZAÑAS-GORDON received the B.E. degree in electrical engineering from National Polytechnic School, Quito, Ecuador, in 2003, and the M.Sc. degree in information technology from The University of Queensland, Brisbane, QLD, Australia, in 2015. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Coimbra, Coimbra, Portugal. Since 2018, he has been a Researcher with the Multimedia Signal Processing Laboratory, Department of Electrical and Computer Engineering, University of Coimbra. His research interests include signal processing, deep learning, optical coherence tomography, scanning laser ophthalmoscopy, and fundus photography.

RUI PROENÇA received the M.D. degree, in 1984, the ophthalmology specialist degree, in 1991, and the Ph.D. degree in ophthalmology, in 1999. He is currently a Senior Ophthalmologist with the Centro Hospitalar e Universitário de Coimbra and a Professor of ophthalmology with the Faculty of Medicine, University of Coimbra, with particular research interest in retina and pathology. Formerly, he was the President of the Portuguese Society of Ophthalmology, the Portuguese Board of Ophthalmology, and the European Ophthalmic Pathology Society.

LUÍS A. DA SILVA CRUZ (Senior Member, IEEE) received the Licenciado and M.Sc. degrees in electrical engineering from the University of Coimbra, Portugal, in 1989 and 1993, respectively, and the M.Sc. degree in mathematics and the Ph.D. degree in electrical computer and systems engineering from Rensselaer Polytechnic Institute (RPI), Troy, NY, USA, in 1997 and 2000, respectively. He has been with the Department of Electrical and Computer Engineering, University of Coimbra, since 1990, first as a Teaching Assistant, and as an Assistant Professor, since 2000. He is currently a Researcher with the Institute for Telecommunications, Coimbra, where he works on image and video processing, coding, and medical image processing. He is a member of the EURASIP, SPIE, and IEEE Technical Societies.