IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 October 2023, accepted 26 October 2023, date of publication 30 October 2023, date of current version 13 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3328386

RESEARCH ARTICLE

Exudate Detection: Integrating Retinal-Based Affine Mapping and Design Flow Mechanism to Develop Lightweight Architectures

MUHAMMAD HUSSAIN
School of Computing and Engineering, University of Huddersfield, HD1 3DH Huddersfield, U.K.
e-mail: M.Hussain@hud.ac.uk

ABSTRACT This article introduces an innovative solution to critical challenges in the automated detection of Diabetic Retinopathy using fundus images. To combat data scarcity, we investigate the process of modeling fundus images by proposing the 'Retina-Based Affine Mapping' mechanism. This facilitated the generation of representative augmentations to model occurrences influenced by various internal and external factors during fundus image acquisition, diverging away from the concept of previous works focusing on generic augmentations focused primarily on data scaling rather than increased representation. Additionally, we propose a 'Design Flow Mechanism' to streamline custom Convolutional Neural Network architecture development, via an internal parameter comparison table, resulting in a highly efficient model with 99.51% validation accuracy using only 1.40 million parameters, outperforming state-of-the-art alternatives such as ResNet-18, consisting of 11.69 million learnable parameters. These contributions enhance the field of automated DR detection, promising significant advancements in medical image analysis and early disease diagnosis.

INDEX TERMS Defect detection, computer vision, lightweight, image classification, timely disease detection.

I. INTRODUCTION

Advancements in retinal imaging are necessary to assist optometrists with timely and effective interventions to suppress the progression of Diabetic Retinopathy (DR), a major cause of visual impairment in the working population of developed countries [1]. The World-vision report 2019 approximated around 2.2 billion people were visually impaired, of which 1 billion cases could have been prevented via timely and effective intervention [2]. One of the early signs of DR is known as Exudates. Characterized as 'yellowish' flecks appearing on the retinal tissue, when observed via a fundus image. Exudates are a result of protein leakage due to weak retinal capillaries, and are haphazard in their shape, size, and locality [3]. The timely detection of Exudates can enable medical practitioners to prescribe suitable prevention strategies or enable them to delay the progression of the disease.

Convolutional Neural Networks (CNNs) belong to the deep learning facet of Artificial Intelligence (AI) aimed at processing image data for automated feature engineering/extraction and classification. CNNs have been implemented for automating inspection processes across various domains such as renewable energy [4], food inspection [5] and security [6]. The use of deep learning for the automated detection of exudates via the fundus image has the potential to provide a timely diagnosis of this early stage of DR belonging to the non-proliferative DR phase, enabling targeted prescription for the suppression of the progression of the disease to the proliferative DR stage.

Although it is appreciated that several works on automating DR detection are available as presented in the literature section, there is, however, no streamlined CNN development strategy for DR CNN development. As a result, architectures

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

may provide acceptable or in some cases high degree of accuracies but are computationally demanding and require high-end computing resources for on-site deployment at clinician sites. We aim to address this issue by proposing the Design Flow Mechanism for facilitating the development of CNN's by keeping a check on the number of integrated convolutional blocks.

Additionally, to address the issue of data scarcity we proposed the retinal-based affine mapping pipeline, aimed at scaling DR datasets in a representative manner to assist with better network generalization during the training phase. Furthermore, several regularization strategies are deployed once a base line accuracy has been achieved to suppress overfitting and enhance the overall generalization capacity of the proposed architecture.

A. LITERATURE

The implementation of conventional image processing and utilization of convolutional neural networks (CNN) for automated Diabetic Retinopathy (DR) is an active area of research. Focus is given to early-stage DR detection such as Microaneurysms, Exudates, and Cotton Wool, to provide timely medical intervention for suppressing the progression of the diseae into the proliferative stage.

Abdullah et al. [7] proposed a framework for the detection of the retinal optic disk, by coupling conventional image processing with a more advanced segmentation approach. The optic disk is a fundamental component within a fundus image hence the authors focus on its proliferation using morphological operations. This enabled the authors to further manifest the optic disk region within the fundus image whilst eliminating surrounding vasculature. A subsequent stage involved estimating the center of the optic disk through an implementation of the Hough transformation. The final step involved implementation of the 'Grow-cut' algorithm for precise segmentation of the region of interest (optic disk). This framework was tested on 5 open-source datasets, achieving 100% accuracy on three, with 99.09% and 99.25% on the others.

Habib et al. [8] proposed a framework for the detection of Microaneurysms, one of the earliest signs of DR. The proposed methodology employed 'Gaussian Matched Filters' for feature extraction. The extracted features were used as inputs for an 'ensemble' architecture. The performance of the classifier as reported by the authors was not very effective, achieving a ROC (receiver operating curve) of 41.5%. This was a clear indication of the architecture's inability to generalize the extracted features obtained via the Gaussian approach. Furthermore, when observing the visual appearance of Microaneurysms, we see that they can be described as very tiny red dots, in some cases difficult to detect with the human eye. Hence, we feel the size of Microaneurysms observed indicates that further proliferation was required before using the resultant output features as an input to a classifier network.

Other work focuses on the detection of Microaneurysms as presented by Murugan and Roy [9]. The authors present a three-point framework for automated detection of Microaneurysms within the retinal tissue. The authors start with data preprocessing, moving onto candidate selection, before training a CNN architecture for detecting the class of interest. The final component (CNN) incorporates a voting mechanism to determine the accuracy of the output. The authors report an AUC (area under the curve) of 92%. This is a significant result given the visual characteristics of the Microaneurysms as mentioned above.

Tan et al. [10] propose a multiclass, two-stage detection architecture for the classification of various DR signs; Microaneurysms, Hemorrhages, and Exudates. The authors subscribe to the proliferation of the relevant classes via segmentation. Focusing on the segmentation of the Exudates, the authors report a sensitivity of 71.58%. It is not clear as to the rationale for selecting segmentation over object detection, as Exudates are non-uniform in their visual appearance. As a result, quantifying the ability of the architecture based on pixel-wise segmentation of the class of interest would be a more challenging task. Alternatively, utilizing object detection, where bounding boxes have more flexibility, can achieve higher performance, and address the non-uniformity issue found with Exudates. However, the selection of segmentation also comes at a higher computational cost as reported by the authors, stating an inference time of approximately 3000 seconds, when running without a GPU-enabled device.

Focusing on the reduction of computational load, Guo et al. [11] propose a fundus segmentation architecture for the classification and segmentation of three classes: Microaneurysms, Vessels and Hard Exudates. These classifications have distinct visual features that can be learned, helping the architecture to generalize. Microaneurysms are the smallest in both width and height. Vessels can vary in length, whilst Exudates are much bigger in size compared to Microaneurysms and Vessels. A lightweight CNN is proposed consisting of less than 40,000 parameters, that is trained on 5 open-source retinal fundus image datasets. The authors present their contribution as a custom CNN architecture with the ability to generalize high-resolution fundus representations without dramatic consequences on the architecture's inference time.

Huang et al. [12] focus their research on pathogenic triggers, arguing that the majority of the literature is based on algorithmic design and development. The authors propose a transformer-based network integrated with an attention module. The proposed network is initially used for exploiting global dependencies within lesion features. Secondly, it is used for the facilitation of exchanges between lesion and vessel features. Regarding network selection, we feel that the decision for implementing a transformer network was debatable.

Transformers have been very successful in the natural language processing (NLP domain), overtaking popular architectures such as Recurrent Neural Networks (RNN).

However, the use of image data for training transformer networks requires a significant number of representative samples for each class. This requirement can be difficult to meet as working in the domain of lesions detection, in most cases, the datasets are significantly biased towards normal classifications. It follows that, scaling of the data needs to be done to obtain sufficient samples for training the transformer network, yet at the same time, the scaling of the data needs to be representative to ensure the trained network has generalized properly. Furthermore, the use of transformers rules out the deployment of trained architectures onto standard CPU hardware without the relevant level of pruning.

Zhou et al. [13] propose a collaborative learning framework for further enhancement of diabetic retinopathy related lesion grading and segmentation. We observe that the authors subscribe to an approach whereby transformer models manipulate the attention mechanism. This is done to facilitate image annotation refinement, with respect to class-specific information. Interestingly, when evaluating the performance via the AUC curve for precision and recall, the accuracy is stated as 70.44%. There could be various reasons for this performance, such as insufficient amounts of data required for training the transformer network. Additionally, the correct classification of segmentation is more stringent as opposed to object detection due to the pixel-level performance consideration for the former, hence resulting in reduced performance.

Rosas-Romero et al. [14] propose a classification framework for the detection of Microaneurysms. Based on the characteristics found within Microaneurysms, the authors focused on the preservation of densely populated regions forming small dots, suggesting Microaneurysms, whilst removing larger Vessels running across the retinal tissue. Post retraining potential Microaneurysms candidates, the authors implement Principal Component Analysis (PCA) along with radon transforms to calculate the most likely candidates representing Microaneurysms. The authors report a classification accuracy of 95.93%.

Shan and Li [15] propose a sparse auto-encoder technique for Microaneurysm detection within retinal tissue. The proposed framework is initiated by splitting images into small segments, followed by an implementation of the sparse auto-encoder mechanism for key feature extraction from the segments. The proposed framework based on regional segmentation and classification of true Microaneurysms achieved notable performance with an AUC of 96.2%.

Tang et al. [16] present a framework dubbed as 'splat-feature' classification aimed at the detection of defects within retinal tissue, observed via a fundus image of the retina. The framework is based on supervised learning, instigating with non-overlapping segment-based partitioning. Each 'splat' (segment) consists of similar pixels in terms of color and spatial locality. Optimal feature extraction from each splat was commissioned through a wrapper mechanism before being introduced as an input to the classification network, achieving an AUC of 96%.

In summary, we conclude there are various research gaps or potential enhancements that can be made to positively contribute to the field of automated DR detection via fundus images. Firstly, although there are a handful of open-source datasets [17] for retinal based DR detection, these are either heavily imbalanced towards the normal class (as is the case with most medical related datasets), or they contain a very small number of samples. To overcome the quantity issue, developers may apply various random augmentations, that are not necessarily representative of the underlying features of the original data, hence contributing to non-generalized architectures.

Furthermore, we observe that the majority of the research undertaken subscribes to the implementation of segmentation processes for identifying and enhancing potential defective regions of the fundus image before feeding the extracted features into a classification network. This methodology can be computationally very demanding, along with additional time requirements as preparing image data for segmentation architectures requires pixel-wise annotation of the training data. Additionally, the pixel-wise annotation will induce a human bias factor, that may negatively impact the overall architectural performance, depending on the accuracy of pixel-wise annotations.

B. CONTRIBUTION AND PAPER ORGANIZATION

Our contributions are based on addressing the issues highlighted in the summary of the literature review. Firstly, we address the issue of data scarcity, by investigating the process of procuring fundus images of the retina i.e., selection and placement of hardware, we can describe the types of variances that may occur due to both internal and external factors such as shading and fundus camera specification. This enables us to model the variance within our data scaling strategy, via a proposed 'Retina-Based Affine Mapping' mechanism. Thus, we do not only scale the data for the sake of increasing the sample quantity of each class, but rather scale it with respect to introducing representative samples.

Secondly, we introduce a Design Flow Mechanism (DFM) for assisting the development of a custom CNN architecture for exudate detection. The proposed DFM assists the development of a lightweight architecture by constraining the number of internal parameters in relation to state-of-the-art (SOTA) image classification models. The success of the two proposed frameworks for data modelling and CNN development is demonstrated as the developed model contained the least number of internal parameters compared to referenced SOTA architectures, whilst achieving a validation accuracy of 99.51%. Further endorsing the high efficacy of the proposed mechanisms, the developed architecture retains its performance across a rigorous evaluation process, with scrutiny of architectural, computational and post-deployment metrics.

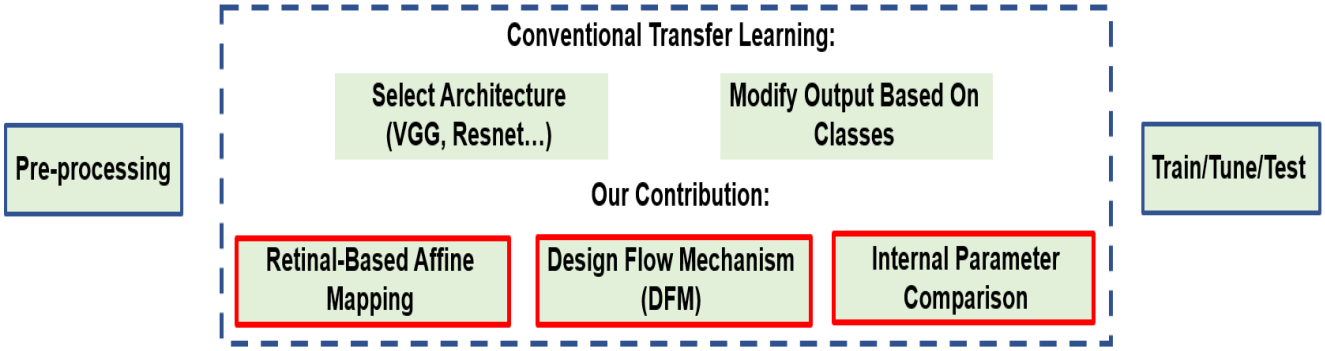

FIGURE 1. Abstract solution comparison.

Figure 1 shows a visual comparison of our contribution against the methodologies presented in the literature section.

II. METHODOLOGY

A. DATASET

Diabetic Retinopathy has four distinct stages: mild, non-proliferative, severe, and proliferative. There are around a dozen publicly available datasets that can be used for research into automated detection of DR. For our research we selected the EyePACS [18], although the dataset had to go through a filtering process for recreating a dataset to suit the aims of the research. There were several reasons for recreating the dataset. Firstly, the original dataset contained almost 90,000 images. Although this was a significantly large set, more than a quarter of the images belonged to the 'no-dr' category i.e., normal fundus images. Furthermore, the dataset was split into four categories depending on the severity of the DR, as our focus was on early-stage detection of DR within the non-proliferative stage i.e., differentiation between normal, and images containing exudates. For this purpose, the dataset was unsuitable in its original form since the non-proliferative category within the dataset contained multiple other signs (classes) within the same image, such as Microaneurysms, Cotton Wools, and Hemorrhages. Thus, the first task was to inspect and extract retinal fundus images that only contained exudates.

Figure 2 presents a visual representation of the two classes. An initial inspection of the dataset was carried out to understand key features which differentiate the two classes, as well as the degree of internal variance for each class. Figure 2 (A & B) belong to the normal class. However, we can clearly observe an element of internal variance in the color depth of the two images. Figure 2 (B) would be much easier for the network to distinguish as its darker shade clearly distinguishes it from fundus images containing exudates, shown in Figure 2 (C& D).

We can observe that exudates appear as yellowish flecks, however, they do not have a uniform structure and can be devised or scattered around the retinal tissue. Exudates also have internal variance in the form of color shading. However, a more significant finding is highlighted in Figure 2 (A & D). We observe that each retinal image contains an optic disk highlighted in green. However, the brightness of this optic disk in normal fundus images, as shown in Figure 2 (A), caused by shading or instrument specification can increase the color intensity towards a bright yellow circle. As noted, earlier exudates are yellowish flecks that can be densely populated, taking a similar visual form to that of the optic disk, hence potentially misleading the network into differentiating an optic disk from a normal image from exudates.

B. PROPOSED METHODOLOGY FOR DATA MODELLING

There were two primary reasons that led us to the development of a custom data scaling mechanism presented as 'Retinal-Based Affine Mapping', shown in Figure 3. This was due to the internal variance and similarity between two classes, such as the similar nature of the high-intensity optic disk in normal fundus images with that of densely populated exudates.

Our proposed methodology presented at an abstract level in Figure 3 involved data scaling, via specifically selected image processing mechanism. Each technique introduced for scaling is selected and tuned, based on addressing issues across the two classes as well as internal variance. The proposed techniques are parametrically tuned in a progressive manner to generate images that are highly representative of the retinal domain, enabling the creation of a concise dataset for the exploration and automated detection of exudates, in the early stage of DR, that if diagnosed in a timely manner can significantly impede the progression rate of DR via targeted medication.

After the generation of a scaled representative dataset, a lightweight CNN architecture was developed. The design stage consisted of an internal parameter comparison (IPC) mechanism to keep a check on the number of resultant parameters due to the introduction of each convolutional block. The comparison mechanism assisted in the number of convolutional blocks that were introduced without increasing the complexity of the architecture, by benchmarking against state-of-the-art image classification architectures such as Resnet-18, and VGG. The internal structure of the IPC is presented in detail in a latter section.

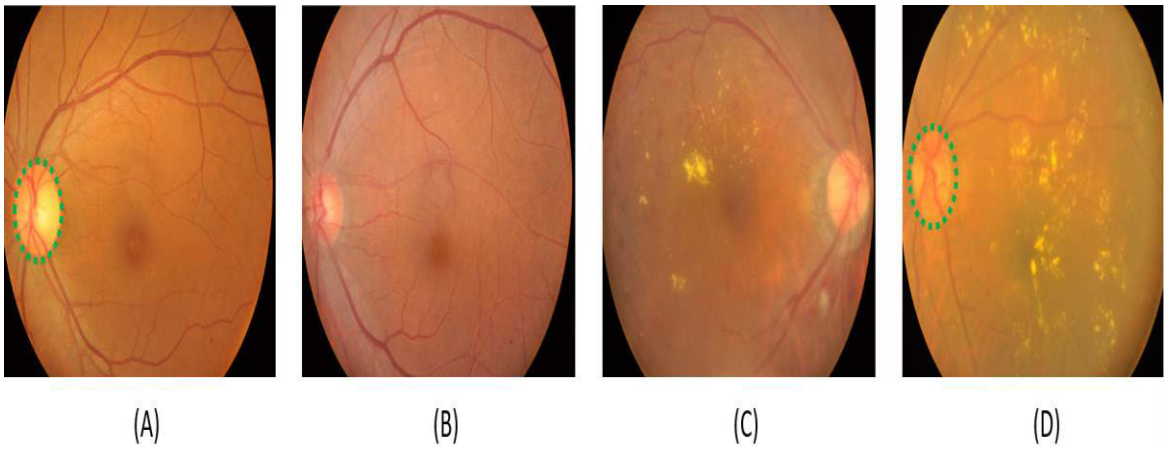

FIGURE 2. Internal variance inspection.

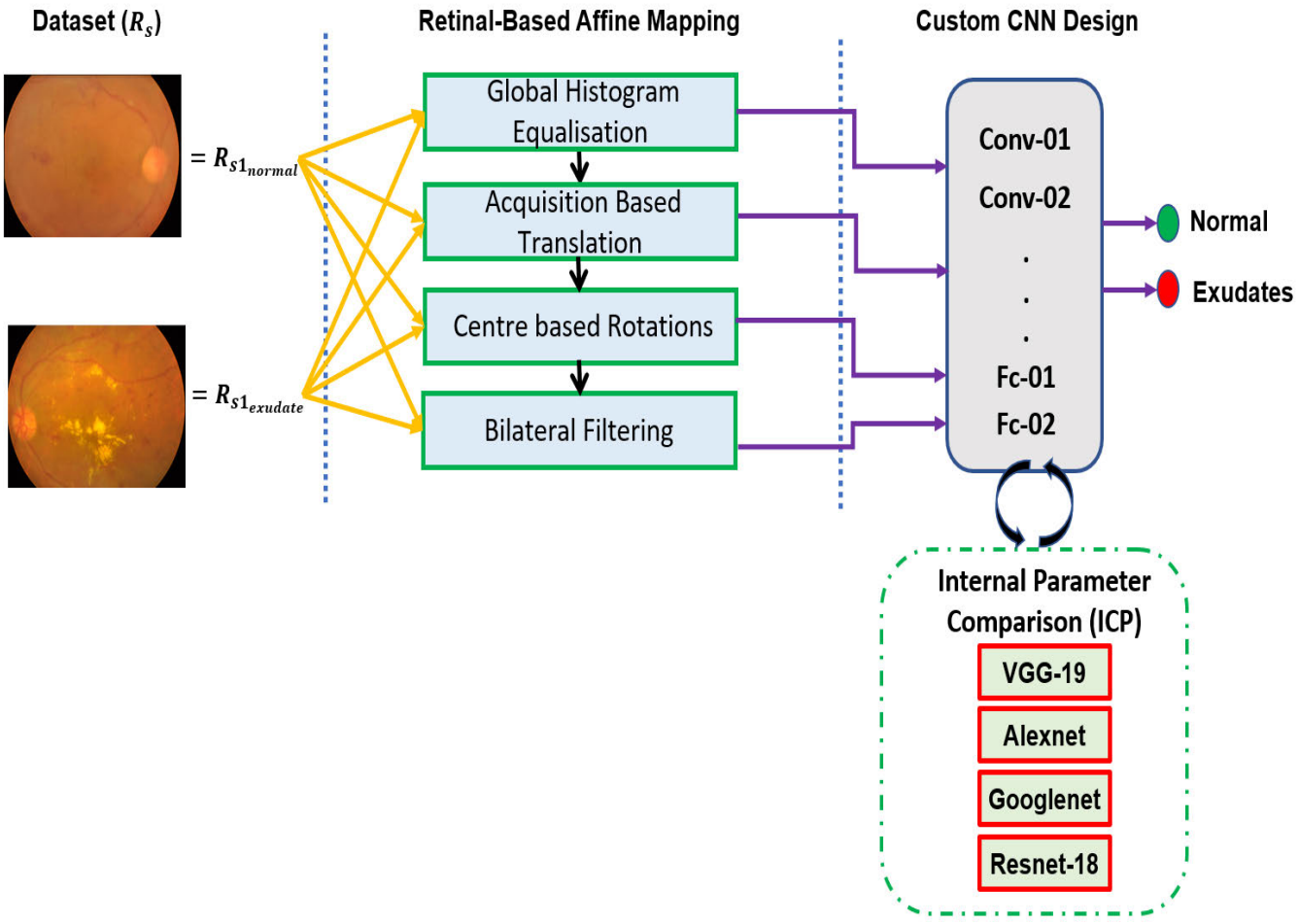

FIGURE 3. Methodology overview.

C. GLOBAL HISTOGRAM EQUALIZATION

The scaling of the dataset as per the 'Retinal-Based Affine Mapping' mechanism presented in Figure 3, starts with an image equalization protocol. As identified in the preceding section, the dataset contained both internal and cross-class variance. One of the ways in which this was manifested was the various shades of retinal tissue caused by varying specifications of instruments utilized for capturing the fundus images. We felt that this low-level contrast between the retinal tissue and the exudates in various images may

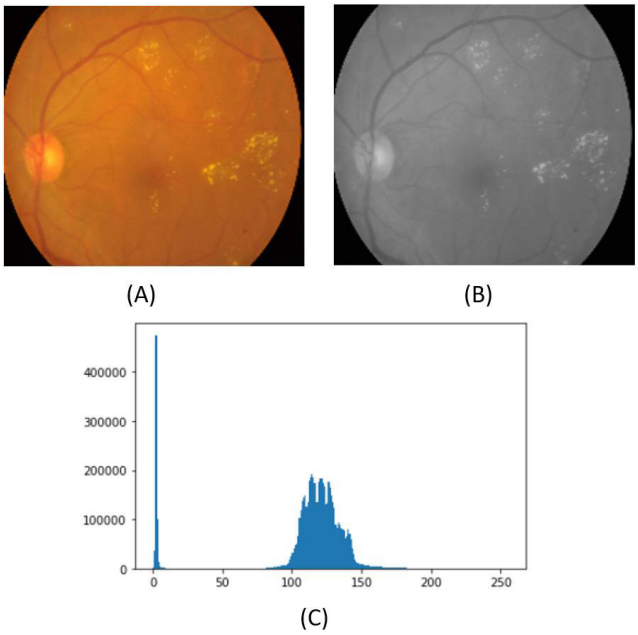

FIGURE 4. Image conversion and histogram.

compromise the capability of the network to correctly differentiate between the two. Figure 4 illustrates the process of obtaining a histogram for a given fundus image: (A) original image; (B) conversion to greyscale; (C) histogram, for the respective image. The image selected to showcase the process is deliberate, as the exudates manifested are not densely populated in a constrained region of the retinal tissue but rather scattered.

It can be observed via the histogram that the majority of the pixels amount to the darker region (left area of the histogram). The exudates, although noticeable, remain around the mid-region of the histogram implying that there is a low level of contrast between the background retinal tissue and the exudates. Also, we observe the brightness of the optic disc also contributes to mid-region pixels. Hence, it can be appreciated that in the case of eliminating the optic disk from the fundus image, the frequency of mid-region pixels would significantly decrease. Therefore, we implement global histogram equalization as a contrast-enhancement mechanism for increasing the pixel contrast intensity between the retinal tissue and the exudates. An objection may be raised here to the selection of global equalization as compared to regional based equalization via techniques such as Contrast Limited Adaptive Histogram Equalization (CLAHE). The premise for our argument would be that by implementing global equalization we are also increasing the intensity of the optic disk, hence the problem of potential misclassification still remains. Actually, exudates are non-uniformly distributed and can be densely populated or sparsely scattered and not constrained to a particular region so, we cannot accurately predict their location before applying CLAHE to the identified region. Hence, we apply global equalization, accepting the fact that the intensity of the optic disk would also be enhanced. However, due to the uniform shape (circle) of the optic disk, we can identify it via the feature extraction process during the CNN training stage. The histogram process starts by placing each pixel value $f[x, y]$ into one of L uniformly spaced segments $h[i]$;

$$h[i] = \sum_{x=1}^{N}\sum_{y=1}^{M} \begin{cases} 1, & if\ f[x,y] = i \\ . & \\ 0, & otherwise \end{cases}$$

where $L = 2^8$ and $MxN =$ image dimensions

Then we calculate the cumulative distribution function (CDF):

$$CDF[j] = \sum_{i=1}^{j} h[i]$$

The input image is then scaled via the CDF function, producing the output image:

$$g[x, y] = \frac{CDF[f[x-y]] - CDF_{min}}{(N \times M) - CDF_{min}} \times (L-1)$$

where $CDF_{min} =$ smallest non-zero number of the CDF

A progression from the original to globally equalized image is shown in Figure 5. It can be seen from Figure 5 (C), that the global equalization mechanism was successful in enhancing the low-level contrast between exudates and the background retinal tissue. At the same time, we observe darker pixels for the vessels as well as the macula (highlighted in green in Figure 5 (C)). The manifestation of these darker pixels does not have a negative impact in differentiating exudates as they appear at the other end of the spectrum post-equalization i.e., brighter pixels. Conversely, the optic disc (red circle in figure 5 (C)) also manifests into brighter pixels similar to the exudates, however, as mentioned earlier this can be resolved via the convolutional process for feature extraction in the CNN development phase.

D. ACQUISITION-BASED MODELING

The second data scaling technique was aimed at modelling variance, which may occur as a result of differences in hardware devices used for procuring fundus images. Primarily, there are two types of digital fundus camera choices for an ophthalmologist. These are shown in Figure 6 as (A) Static and (B) Mobile. Static devices are fixed, and the patient places their chin on a pre-defined point to provide the correct level of coverage for capturing their retina. Conversely, mobile devices, are handheld by the optometrist, and hence an element of human interaction in the process of procuring the fundus images contributes to human bias, such as unwanted movement, resulting in border pixel elimination of the retina.

The focus of acquisition-based translations was on the modeling of variations introduced in the captured fundus images when using a 'Mobile' fundus camera device. Before any modelling was undertaken it was imperative to understand and appreciate the factors that were contributing to the

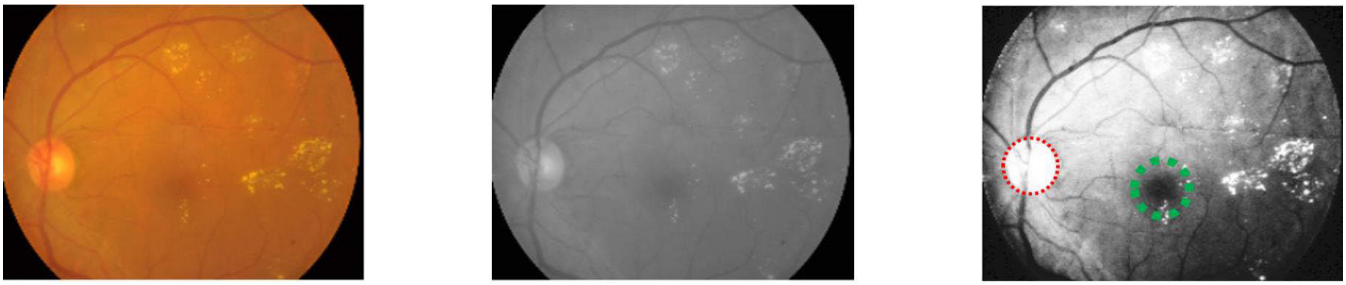

FIGURE 5. Global equalization comparison.

FIGURE 6. Fundus camera variants (A) Static (B) Mobile.

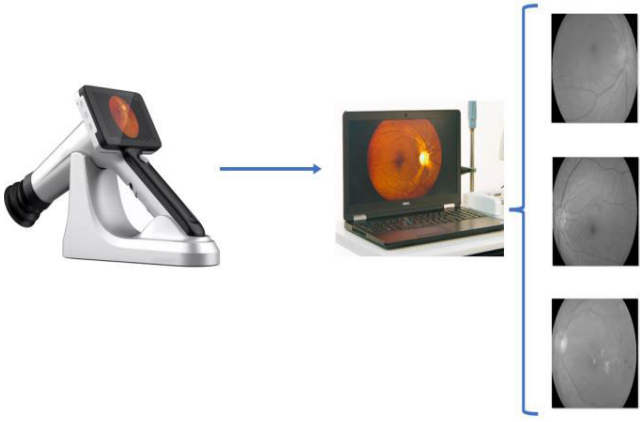

FIGURE 7. Fundus procurement process.

variations and how the resultant variations manifested in the practical realm. Figure 7 presents a retina fundus capturing process, utilizing a mobile digital fundus camera device.

It can be observed there are three distinct stages in the process of acquiring digital fundus images via a mobile handheld device. The first stage involves the optometrist operating the digital device for capturing fundus images of the retina. Here, the handling of the capturing device by the optometrist as opposed to having a fixated device can result in rotational variations in the resultant image. The retina has distinct components that appear as a fundus, and these are hosted in a particular region of the retina, optic disk, and macula. The optic disk as mentioned earlier can be similar in appearance to exudates i.e., yellowish color. Hence, depending on the severity of fluctuations in the handheld device the location of the optic disk which is approximately center-right or center-left depending on the left or right eye, can be significant. Thus, a model that has generalized on the assumption that all optic disks are strictly center-right, or center-left may misclassify this type of variance as exudates.

After appreciating the type of variance that may be introduced, a translational mechanism was implemented for newly-generated fundus images with a pre-defined shift variable. By defining the shift variable in the form of (z_x, z_y), we present the transformation matrix as;

$$M = \begin{bmatrix} 1 & 0 & z_x \\ 0 & 1 & z_y \end{bmatrix}$$

The matrix M in practice is converted into an array (float32) followed by the applying of an affine transform:

$$dst\ (x, y) = src(M_{11}x + M_{12}x + M_{13}, M_{21}x + M_{22}y + M_{23}$$

where src = input image, dst = output image and M = transformation matrix

E. CENTRE-BASED ROTATIONS

The images produced after pixel shift resulted in new images that shifted the content of the fundus images with respect to the pre-defined number of pixels in the shift variable. This also resulted in the elimination of certain pixels that were affected by the shift element at the margins of the fundus images. The issue here was that exudates are indiscriminate in where they appear on the retinal image, including borders of the retina. To provide further variations and scaling of the original dataset without losing the margin-pixels, centre-based rotations were introduced. Conventionally, the rotation of an image with respect to an angle (Θ) is achieved via the transformation matrix:

$$M = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

However, our focus was based on referencing the centre of the image for the rotation, hence the modified transformation

matrix:

$$M = \begin{bmatrix} \alpha\beta(1-\alpha)\cdot centre.x - \beta\cdot centre.y \\ -\beta\alpha\cdot centre.x + (1-\alpha)\cdot centre.y \end{bmatrix}$$

where;

$$\alpha = scale \cdot cos\theta, \beta = scale \cdot sin\theta$$

where center = centre of rotation (input image), θ = rotation angle in degrees and scale = isotropic scale factor

F. BILATERAL FILTERING

The final technique for scaling the original dataset was bilateral filtering. Exudates appearing on the retinal tissue are non-uniform in their shape though they consist of margins (edges). These edges, depending on the level of contrast with respect to the retinal tissue, can be stark or blurred. The objective therefore is to effectively remove background noise whilst preserving the edges and lines found within the retinal tissue. The first component of the bilateral filter is essentially linear Gaussian smoothing:

$$g(x) = \left(f * G^s\right)(x) = \int_R f(y)\, G^s(x-y)dy$$

where $f(y)$ weight $= G^s(x-y)$ and depends solely on the spatial distance of $\|x-y\|$

Bilateral filtering introduces a weighting component that is dependent on the tonal distance of $f(y) - f(x)$, hence the result:

$$g(x) = \frac{\int_R f(y)\, G^s(x-y)dy G^t(f(x) - f(y))dy}{\int_R G^s(x-y)G^t(f(x) - f(y))dy}$$

Note the weights solely depend on image values, hence explicit normalization is required to make sure the sum of all weights is equal to one.

A sample set of the generated images post-bilateral filtering on the original dataset is presented in Figure 8. It can be seen that the blurring of the background (retinal tissue) whilst preserving the edges and margins of the elements within, had enhanced the contrast in favor of the exudates. Furthermore, as bilateral filter is also indiscriminate in its application, edges/lines and margins for other components such as capillaries and the optic disk are also preserved. The preservation of these unwanted components that may potentially lead to misclassifications can now be dampened via the convolutional block design of the CNN, presented in the subsequent section.

G. PROPOSED ARCHITECTURE DESIGN MECHANISM

The resultant dataset post modelling (presented in the preceding sections) was separated into training, validation, and testing datasets as presented in Table 1.

The next stage of the research focused on the development of a lightweight architecture specifically aimed at the detection of exudates within retinal fundus images. The inspiration for developing a custom CNN architecture as opposed to transfer learning via state-of-the-art (SOTA) networks, was to propose a computationally friendly architecture that can be deployed on-site without demanding specific GPU hardware for carrying out inference. Our proposed mechanism, presented in Figure 9, enables us to maintain a streamlined process in determining the correct number of convolutional blocks, whilst keeping a check on the number of resultant parameters with respect to various SOTA architectures. The SOTA architectures used for parameter comparison in our proposed development mechanism were VGG-19 [19], Resnet-18 [20], and Googlenet [21]. The design stage was split into blocks, with each block consisting of a convolutional layer, activation function, max-pooling, and a fully connected layer. The parameters of interest i.e., learnable parameters, were based on the convolutional and fully connected layers. Max-pooling was introduced to complement the data transformation described earlier. This enables aggregation of the resultant convolutional feature maps, reducing the size and also reducing positional dependency:

$$h^l_{xy} = max_{i=0,\ldots,s,j=0,\ldots,s} h^l_{(x+i)(y+j)}$$

where z^l = pre-activation output of layer l, h^l = activation of layer l

Various activation functions have been implemented in different CNN architectures. In our research, we selected the ReLu activation function. The rationale for this was manifold. Firstly, the fact that it addresses the issue of vanishing gradients, which is a major problem with activation functions such as sigmoid. Also, it is computationally expedient as its mathematically composition is essentially a max operation. Hence, any negative value is replaced with a zero and all positive values are retained without transformation:

$$f(x) = \max(0, x)$$

Once the fundamental components of a block have been determined (explained above), Figure 9 presents the proposed logic for determining the correct number of CNN blocks. The proposed flow mechanism was designed to achieve both research objectives i.e., lightweight architecture and at the same time achieving high performance. The proposed mechanism starts by developing the first block (number of filters, number of neurons for fully connected layer). The resultant parameters of this block are first compared with the benchmark SOTA architectures (VGG-19, Resnet-18, and Googlenet), and if the number of parameters exceeds any one of the SOTA architecture measures, then the block had to be redesigned with a reduced number of filters and neurons within the fully connected layer. If the number of learnable parameters within the architecture was less than all of the SOTA architectures, then the architecture was trained on a scaled and split dataset presented in Table 1. After training, the model was validated to observe if the accuracy was above 70% (second research objective) and if so, the model was regularized and trained for a greater number of epochs before finally being deployed on the test dataset. In the case of

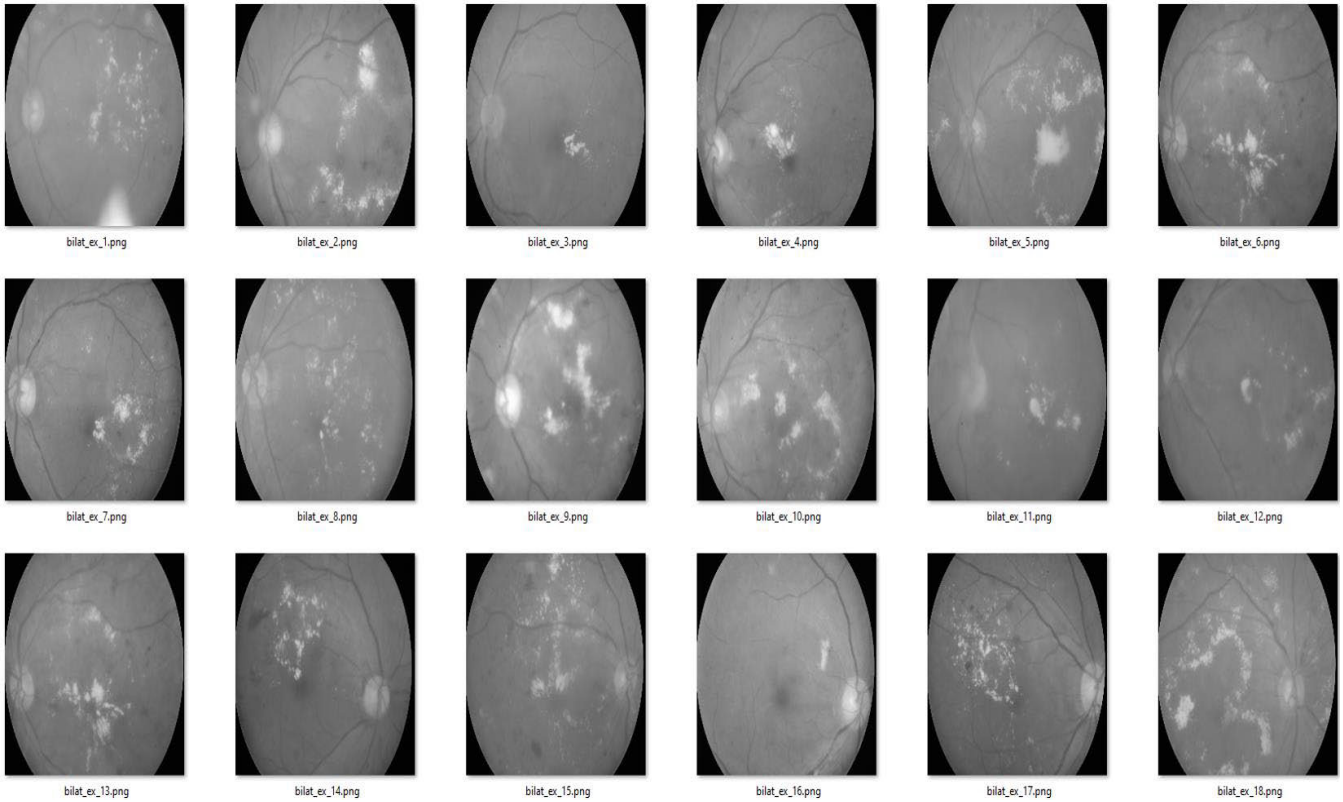

FIGURE 8. Generated augmentations for bilateral filtering.

the accuracy not achieving the 70% threshold, an additional convolutional block was introduced into the existing architecture to enhance its generalization capacity, whilst keeping a check on the number of resultant parameters. The developed CNN architecture based on the Design Flow Mechanism (DFM) proposed in Figure 9 is presented in Figure 10. The architecture consists of 2 convolutional blocks and two fully connected layers feeding into the final output layer containing two outputs. An additional batch normalization component was included within the convolutional blocks, to reduce internal covariant shift that may occur within the two classes. The compliance of the developed architecture with respect to the two criteria outlined in the design mechanism is presented in detail within the results section.

H. REGULARIZATION MECHANISM EXPERIMENTATION

In order to provide the most suitable regularization strategy, the strategies under investigation were batch normalization (BN) and Dropout (DO). These were tested on the actual training data, presented earlier in Table 1. However, as this was preliminary training for regularization selection, the training epochs were reduced to 15. Once the appropriate regularization strategy had been selected, the training epochs were increased and formal training of the architecture with training, validation and test sets were carried out, as shown in the results section. The hyperparameters used for selecting the regularization strategy are presented in Table 2. Furthermore, the learning rate was set at 0.02 as opposed to the conventional 0.001. The rationale for a higher learning rate was to accelerate the training process with respect to the limited GPU use provided by Google Colaboratory. More importantly, as this was an experimental stage for selecting the appropriate regularization strategy, post-selection the batch size and epochs were increased for the final stage of training.

TABLE 1. Processed dataset post splitting.

Class	Training	Validation	Test
Normal	1455	311	313
Exudates	1915	410	411

I. PROPOSED ARCHITECTURE (NO REGULARIZATION)

Figure 11 presents the (a) loss and (b) accuracy of the proposed architecture presented earlier in Figure 10. It can be seen that the performance of the architecture was superior to a random classifier i.e., an accuracy ranging around 50-55% would render the architecture as simply guessing the outputs without proper generalization. This was evidence that the architecture had the internal capacity to generalize on the dataset and hence justified the decision to apply various

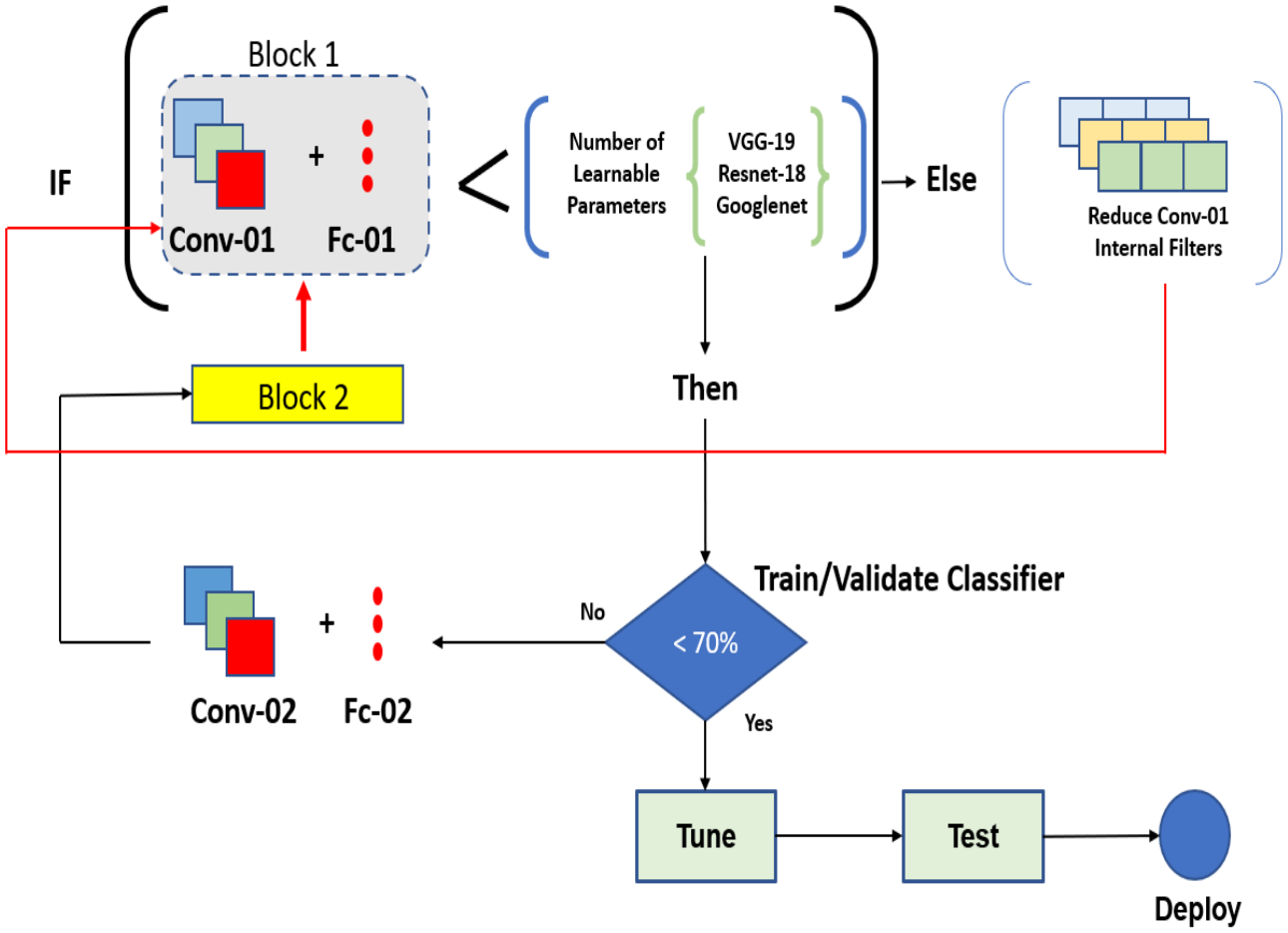

FIGURE 9. Proposed design flow mechanism.

TABLE 2. Hyperparameters.

Batch Size	32
Epochs	15
Learning Rate	0.02
Loss	Cross Entropy
Optimizer	SGD-M

regularization strategies with the aim to improve the overall generation capacity of the architecture.

J. DROPOUT

The first regularization procedure to be employed to the developed architecture was dropout. In general, CNN architectures trained on small datasets, as is the case with the dataset presented in Table 1 for this research, have a high propensity towards overfitting [22]. In the context of image classification, during training the architecture may memorize input features, presenting a false sense of generalization. As a result, when testing on the unseen test data or introducing new images post model deployment with minor variations to the originally trained images, the model performs poorly. Theoretically, this problem can be solved by training on all possible model configurations on the same dataset and then averaging the prediction obtained from all models. However, from a practical standpoint this is not feasible due to computational, cost and time constraints.

To overcome this problem, we borrowed a method from the Machine Learning (ML) domain known as the 'ensemble method'. The completion of the ensemble methodology involves the training of multiple models (based on the same architecture) with differing configurations, before averaging for the optimum result. However, the dilemma with applying the ensemble method in deep learning, specifically CNN's, is that each architectural configuration would contain millions of parameters. Hence, the implementation of even a small number of these 'mini' CNN networks based on the ensemble framework would present computational, memory and storage issues, especially when trying to combine multiple mini architectures.

Based on the above premise, dropout can be utilized for the approximation of several CNN architecture configurations in a parallel fashion. This was accomplished by randomly 'switching-off several neurons within the hidden layer(s) of

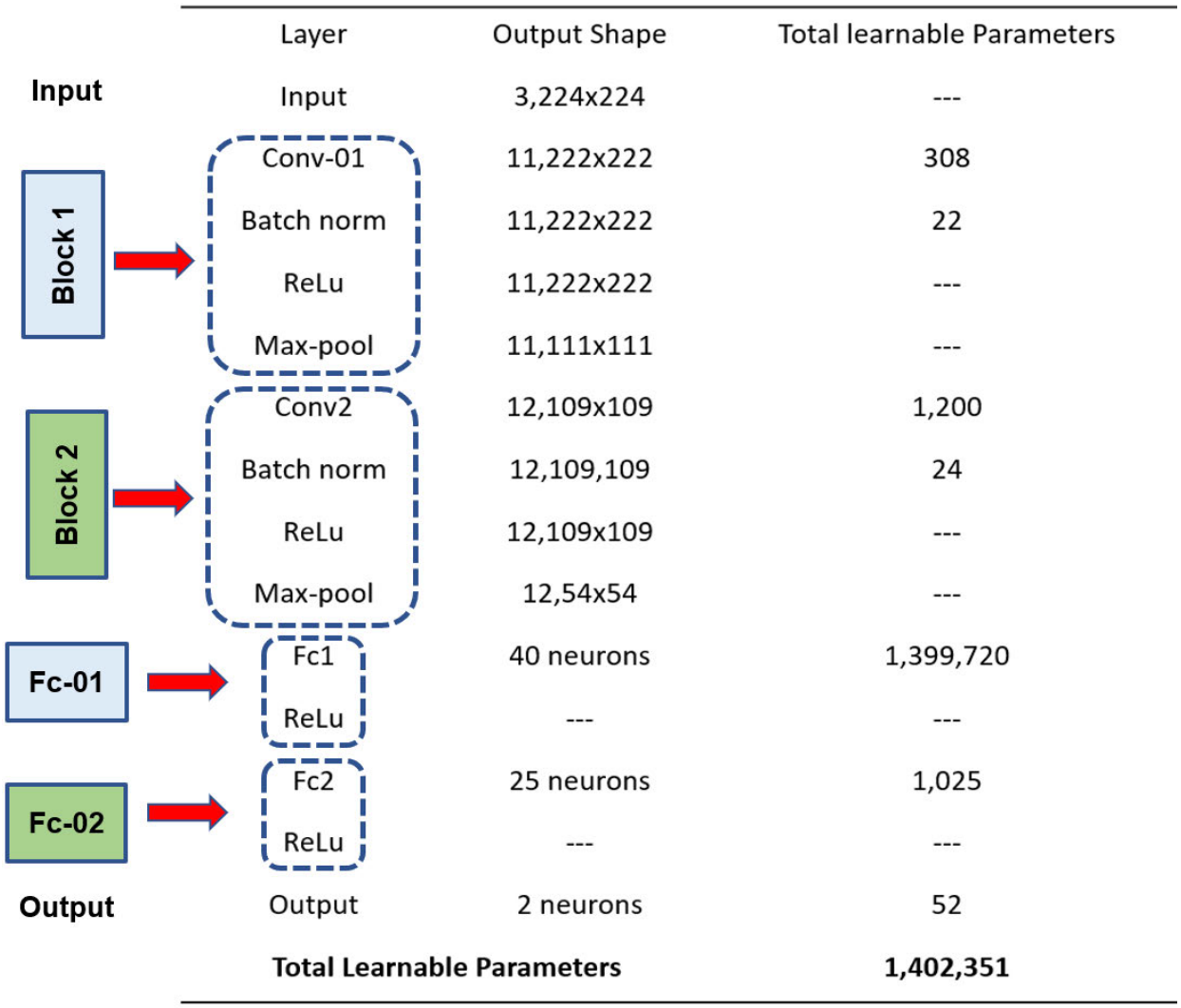

FIGURE 10. Internal composition for designed architecture.

the architecture. As a result, connections to and from the 'disabled' neurons were provisionally disabled. By applying this process on different layers within the developed architecture, we were effectively creating multiple mini architectures. Due to this, neurons within the hidden layers that may have previously co-adapted to prevail over mistakes from prior layers, would be unable to do so, hence compelled to go through a more generalized learning process. The dropout mechanism included a hyperparameter 'p', referring to the probability of neurons that would be momentarily disabled in various parts of the network. The dropout rate could also be independently configured for different layers within the architecture. As per our architecture a global dropout rate of 0.5 was selected. Figure 12 presents the performance of the architecture based on the implementation of dropout at a drop rate of 0.5. It is evident from the performance both (a) loss and (b) accuracy, that the introduction of dropout provided significant improvement in the performance.

K. BATCH NORMALIZATION

Although dropout resulted in good performance when inspecting the training and validation accuracies in Figure 12 (B), there was a small difference of 2.04% between the two respective accuracies. This difference pointed towards a degree of overfitting. Hence the next regularization technique was implemented with the aim to reduce the difference between the two respective accuracies whilst maintaining high performance via batch normalization.

For batch normalization, the dataset was divided into batches and the batches used within the training process. For example, we consider two batches: Batch one and Batch two. Batch one would contain a subset of the two classes i.e., normal and exudates. It is likely that batch two would contain another subset of the respective classes that have contrasting features whilst belonging to the same class.

Consequently, when projecting both batches onto a feature space it we observe that even though the subsets contain the

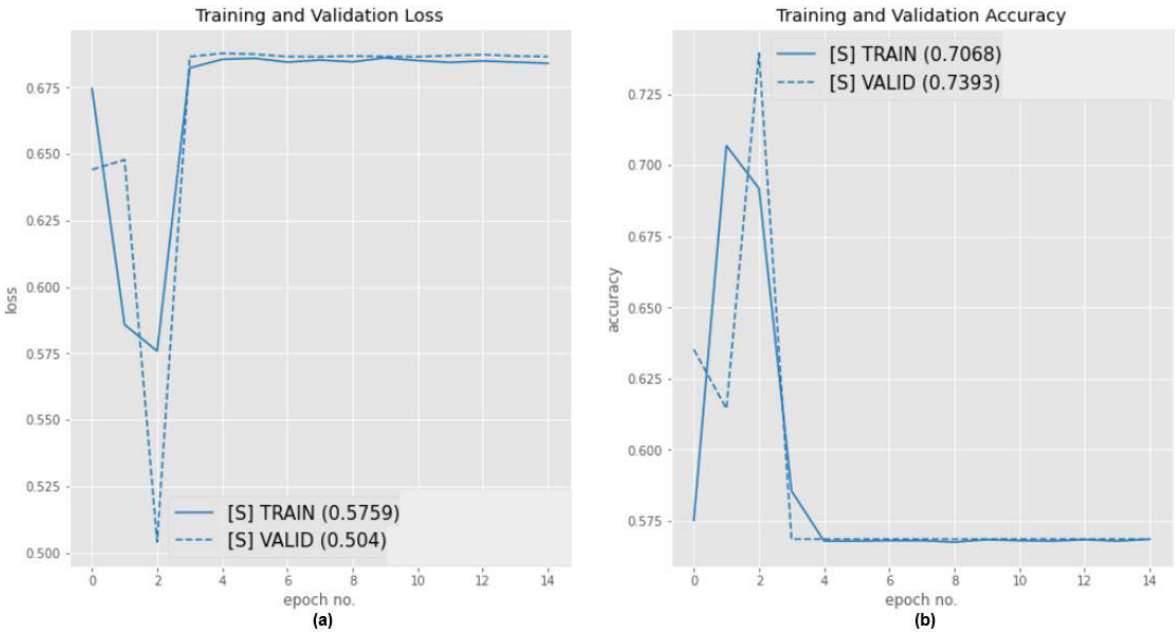

FIGURE 11. Pre-regularization performance for initial model.

same classes, each subset populates a different region within the feature space. This problem may be resolved by randomizing the data at the input layer; however, doing so within the hidden layers whilst training could also be beneficial. The justification for this was due to the input distribution of every hidden neuron changing every time there was a parameter update, resulting in internal covariate shift.

Due to the manifestation of internal covariate shift, training can become slow requiring a small learning rate for convergence. To address this, we implement batch-based normalization within the mini-batches.

The mathematical expression for batch normalization is shown below.

$$\boldsymbol{u}^{[l]} = \frac{1}{\boldsymbol{m}} \sum_i \boldsymbol{z}^{[l](i)}$$

$$\sigma^{[l]2} = \frac{1}{\boldsymbol{m}} \sum_i (\boldsymbol{z}^{[l](i)} - \boldsymbol{u}^{[l]})^{\mathbf{2}}$$

$$\boldsymbol{z}^{[l]} = \boldsymbol{W}^{[l]} - \alpha^{[l]} = \boldsymbol{g}^{[l]}(\underline{z}^{[l]})$$

$$\boldsymbol{z}^{[l](i)}_{\boldsymbol{norm}} = \frac{\boldsymbol{z}^{[l](i)} - \boldsymbol{u}^{[l]}}{\sqrt{\boldsymbol{\sigma}^{[l]2} + \boldsymbol{\epsilon}}}$$

$$\underline{\boldsymbol{z}}^{[l](i)}_{\boldsymbol{norm}} = \boldsymbol{\gamma}^{[l]} \boldsymbol{z}^{[l](i)}_{\boldsymbol{norm}} + \boldsymbol{\beta}^{[l]}$$

where i= ith data point in the mini-batch, l=lth layer in the network and k=kth dimension in a given layer of the network.

The mean of the batch is calculated via the first equation. The second equation was used for calculating mini-batch variance. Then z_{norm} was computed based on the subtraction of the mean from z and then dividing by the standard deviation. Epsilon (ϵ) was defined as a small value, residing within the denominator to prevent zero based division. The final calculation was accomplished through the multiplication of z_{norm} with respect to a scale and adding of a shift (β). Noting that $\underline{z_{norm}}$ replaces z becoming the input to the nonlinearity i.e., activation function (ReLu). β and γ were learned via the training of the architecture with W (weight parameters). Starved of batch normalization, the input layer $a^{[l-1]}$ would pass through a preserved transform, followed by a non-linear activation function ($g^{[l]}$), providing the definitive activation function ($a^{[l]}$) from the unit, shown below.

$$\boldsymbol{a}^{[l]} = \boldsymbol{g}^{[l]}(\boldsymbol{W}^{[l]}\boldsymbol{a}^{[l-1]} + \boldsymbol{b}^{[l]})$$

Alternatively with the application of batch normalization with a transform (BN),

$$\boldsymbol{a}^{[l]} = \boldsymbol{g}^{[l]}(\boldsymbol{BN}(\boldsymbol{W}^{[l]}\boldsymbol{a}^{[l-1]}))$$

Furthermore, the execution of batch normalization introduces two new parameters for each unit that is, β and γ. If β = mean and γ = square root of γ squared plus epsilon, then $\underline{z_{norm}} = z$ and the outcome is essentially an identity function. This demonstrates that the implementation of batch normalization would not in any case decrease the performance of the optimizer as in that case it reserves the right to opt for the identity function. In other words, the optimizer in our case SGD-M, would only utilize batch normalization if it was manifesting a positive impact on the performance of the architecture during training.

Figure 13 presents the (a) loss and (b) accuracy for the training and validation sets with the implementation of batch normalization. Firstly, it can be observed that the introduction of batch normalization provided a further increase in the performance of the architecture during training, compared to dropout. More importantly, it achieved the objective outlined before its implementation, which was

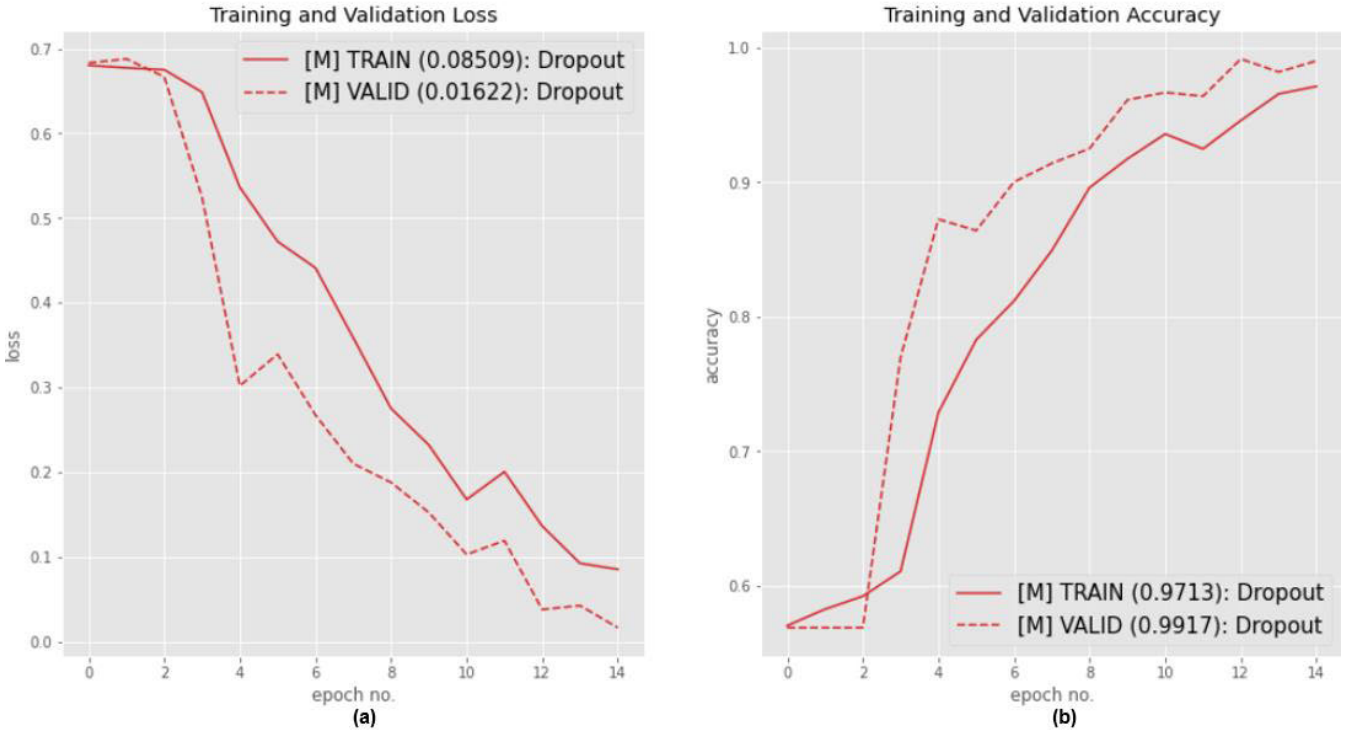

FIGURE 12. Implementation of dropout @ p= 0.5.

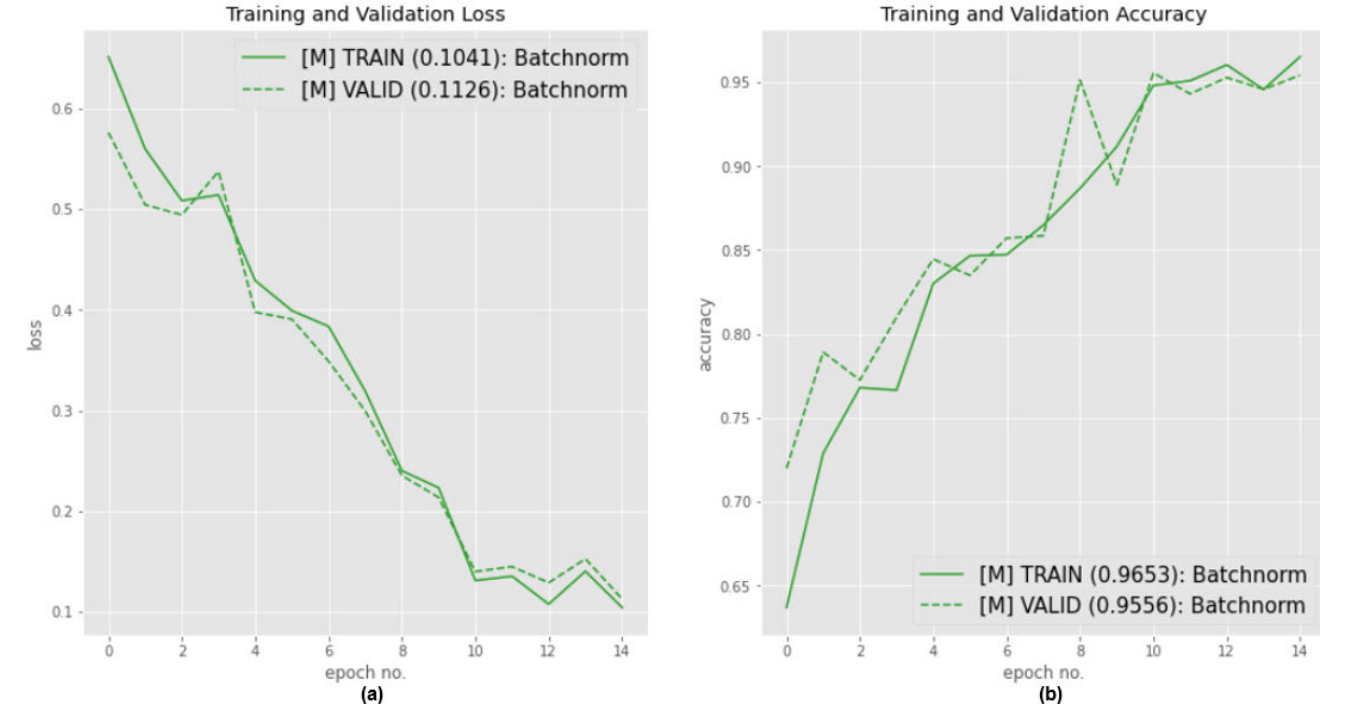

FIGURE 13. Post batch normalisation training performance.

the reduction of overfitting whilst maintaining high performance. Compared to the percentage of overfitting for dropout (2.07%), post implementation of batch normalization, this reduced to 0.67%. At the same time, the validation accuracy also improved to 99.03% compared to 98.75% for dropout.

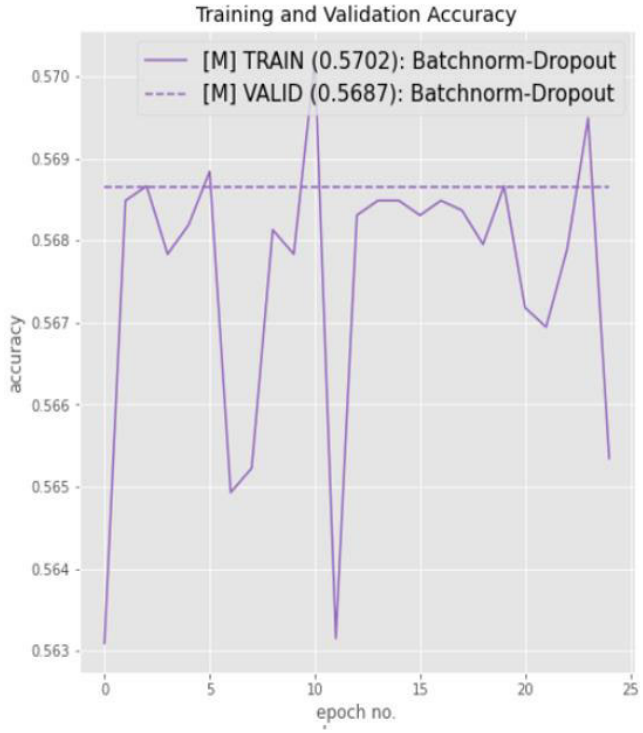

FIGURE 14. Batch normalisation combined with dropout.

L. COUPLING BATCH NORM AND DROPOUT

Noting the impressive performance by both regularization techniques, the next natural progression was to combine the two techniques together, that is implement batch normalization with dropout simultaneously within the same architecture during training. Unexpectedly, as shown in Figure 14, this had an adverse impact on the architecture's performance, with both the training and validation accuracies reducing. This significant decrease in performance was potentially due to the disharmony between the two techniques caused by the shift in variance. The dropout behaves differently to batch norm, in that, it changes the standard deviation of the training distribution but not for the validation set. This difference in the training and validation distribution resulted in the decrease of the architecture to essentially become a random classifier.

M. RATIONALE FOR SELECTING BATCH NORM

Figure 15 provides a comparison between the training and validation accuracies for (a) dropout and (b) batch normalization. Although, it appears logical to select dropout as the regularization technique for integration within the proposed architecture, batch normalization was selected. The rationale for this is multifaceted. Firstly, although dropout provided the highest accuracy (99.17%), the difference between the training and validation accuracies i.e., degree of overfitting stood at 2.04%. At the same time batch normalization also provided impressive performance with the validation accuracy

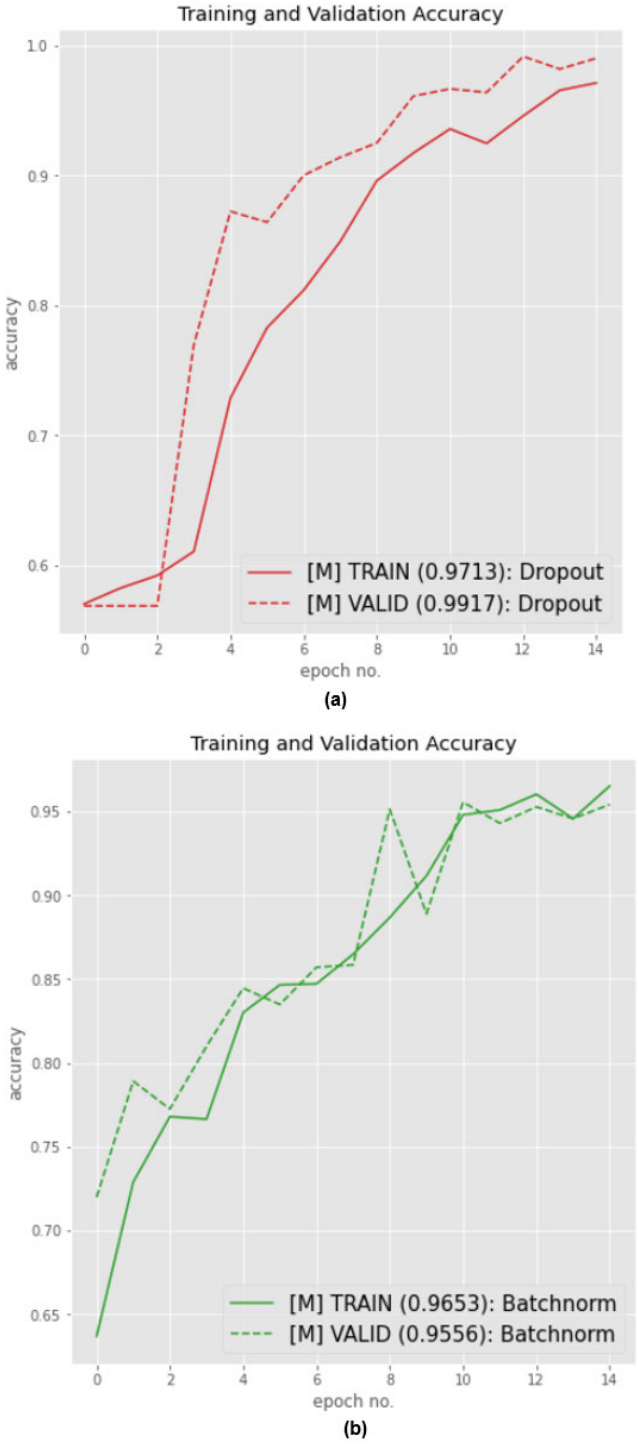

FIGURE 15. Dropout vs batch normalisation.

at 95.56%, but more importantly the degree of over fitting was significantly less (0.97%) with respect to the training accuracy.

Additionally, observing Figure 15, it can be noticed that the training and validation curves through batch normalization remained in close proximity as the training progressed,

TABLE 3. Hyperparameters.

Batch Size	32
Epochs	25
Learning Rate	0.02
Loss	Cross Entropy
Optimizer	SGD-M

TABLE 4. Increased training performance.

Metric	Score (%)
Accuracy	99.72
Precision	99.76
Recall	99.68
F1-score	99.76
MCC	99.43

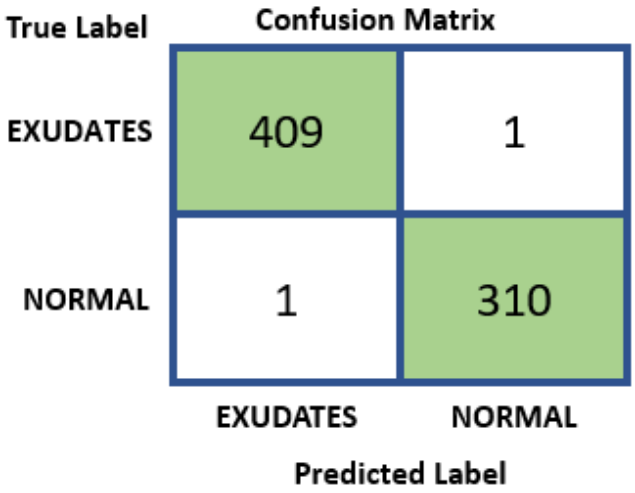

FIGURE 16. Increased epochs performance (Confusion Matrix).

whilst a higher difference throughout the training process was observed for the dropout technique.

III. RESULTS

A. HYPER-PARAMETERS

The development of the CNN architecture would be an iterative process that could require multiple iterations before obtaining an architecture that meets both research objectives i.e., computationally lightweight but at the same time highly performant. Hence, to make sure that each iteration was carried out in a controlled manner, providing a fair play field for each design proposal, the hyperparameters for model training were globally defined. The globally defined hyperparameters are presented in Table 3.

B. INCREASING TRAINIG TIME(EPOCHS)

The first step towards enhancing the performance of the architecture was increasing the epochs i.e., the training time. For the regularization phase within the experiment, the epochs were limited to 15 to cater for the GPU constraints with Google Colaboratory but more importantly, the focus was on selecting the most promising regularization strategy for the

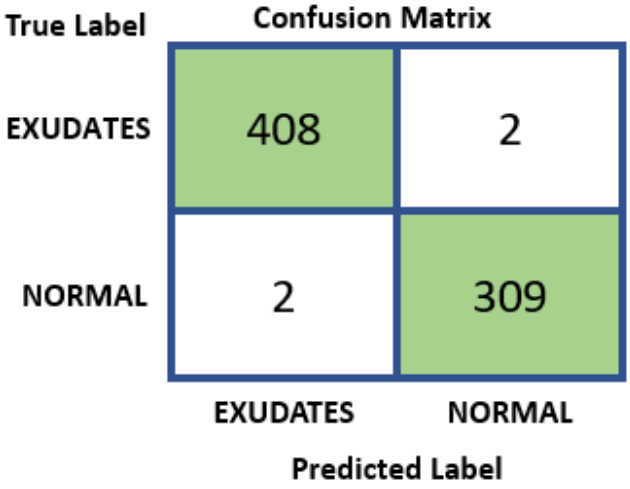

FIGURE 17. Decreased LR performance (Confusion Matrix).

proposed architecture. Once batch normalization had been selected, the training time could be increased to achieve further generalization. The effectiveness of our proposed architecture via DFM coupled with the correct regularization strategy is illustrated in the confusion matrix (Figure 16).

To provide a more robust evaluation of performance, Table 4, presents a granular metric-based evaluation of the trained architecture. The architecture achieves a score of 99.43% using the Matthews Coefficient Correlation (MCC), supporting the performance presented in the confusion matrix (Figure 16).

C. DECREASING LEARNING RATE

Before comparing with SOTA architectures, an additional strategy was tested to observe its impact on the performance of the architecture. This involved reducing the learning rate from 0.02 to 0.001, to overcome any local minima that may be blocking the architecture's convergence path towards optimal performance. Interestingly, from the confusion matrix it can be observed that the decreased learning rate resulted in a minor decrease within the architecture's classification ability, as shown in Figure 17.

Similarly, the small decrease in performance was evident in granular evaluation across a range of metrics, with the overall MCC falling to 98.87%, as shown in Table 5. This result supports the effectiveness of our DFM and the 'Retinal-based Affine Mapping' strategy for introducing domain specific augmentations. As the architecture was able to converge with a higher learning rate, this can be interpreted as less training time, resulting in less GPU usage time.

Figure 18 provides a comparison between the performance of the two learning rates. It can be observed that the higher learning rate (0.02) was able to outperform the conventionally utilized learning rate of 0.001. As mentioned earlier the result attests to the high efficacy of the proposed retinal mapping strategy and the DFM but also has post deployment benefits.

TABLE 5. Decreased LR-0.001 performance.

Metric	Score (%)
Accuracy	99.45
Precision	99.51
Recall	99.36
F1-score	99.51
MCC	98.87

FIGURE 18. Learning rate comparison.

For example, in the case of data drift, due to external factors, the architecture would be able to be retrained to an acceptable convergence degree, in less time, requiring less computational resources with the learning rate at 0.02 as opposed to 0.001.

D. SOTA COMPARISON

The previous section demonstrated that the developed architecture was able to highly generalize on the retinal dataset, providing an overall F1-score 99.76% with a learning rate of 0.02.

Table 6 presents the metric-based comparison of the architectures under investigation. Firstly, focusing on the F1-score it can be observed that ResNet-18 and GoogleNet provided optimal performance, with the proposed architecture coming in second place with a differential of 0.49%. What is more interesting and requires further explanation is the AlexNet performance. AlexNet, was selected for comparison as its internal architectural composition is the most similar to the proposed architecture. That is, AlexNet contains 5 convolutional blocks followed by 3 fully connected layers, whilst the proposed architecture contained only two convolutional blocks followed by 2 fully connected layers. Intuitively, due to the similarity in the architectural composition, as shown in Figure 19 it would be expected that both architectures would demonstrate similar performance. However, this was not the case, as the proposed architecture provided highly effective performance across all metrics, whilst AlexNet performance can be categorized as essentially a random classifier. Furthermore, this again testifies to the efficacy of the retinal mapping augmentations and the proposed DFM, highlighting the fact that simply increasing the convolutional blocks does not guarantee better performance, but rather improving the quality of the dataset via the introduction of domain specific sample generation and carefully tuning the architectural computational parameters provides a more robust and well generalized architecture.

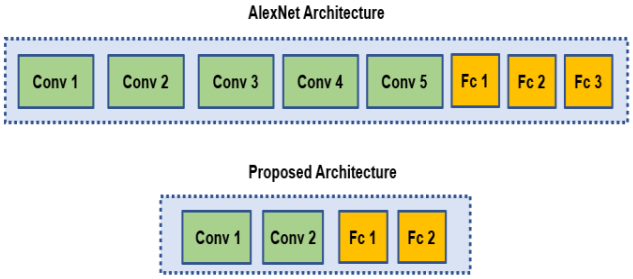

FIGURE 19. Internal block comparison.

TABLE 6. SOTA performance comparison.

	Metric	Score (%)
Proposed Architecture		
	Accuracy	99.45
	Precision	99.51
	Recall	99.36
	F1-score	99.51
	MCC	98.87
ResNet-18		
	Accuracy	100
	Precision	100
	Recall	100
	F1-score	100
	MCC	100
AlexNet		
	Accuracy	56.87
	Precision	100
	Recall	56.87
	F1-score	72.50
	MCC	---
GoogleNet		
	Accuracy	100
	Precision	100
	Recall	100
	F1-score	100
	MCC	100

Whilst both ResNet and GoogleNet provided higher performance, albeit by a differential of 0.49% the objective of the research was to not only provide a high performant architecture but one that is also lightweight regarding its internal layer composition, making it a feasible option for deployment on computationally constrained devices. Hence an architectural and post deployment based comparison was required to provide a holistic overview of the most suitable model for the application.

E. COMPUTATIONAL COMPLEXITY COMPARISON

This section focuses on the validating the second part of the research objective i.e., the development of a lightweight architecture. Table 7 presents the architectural and computational statistics for the respective architectures.

TABLE 7. Architectural & computational comparison.

Architecture	GMAC's	Parameters (M)
Proposed	0.03	1.40
ResNet-18	1.82	11.69
AlexNet	0.72	61.10
GoogleNet	1.51	13.00

Focusing on the computational complexity (GMACs), it is clear that the proposed architecture was the most effective. Multiply-accumulate operations (GMACs) are utilized for assessing the speed of an architecture looking at the number of computations involved within its internal operations. Alternatively, Floating Point Operations Per Second (FLOPS) could have been utilized for achieving the same objective. However, the rationale for selecting GMACs was based on the fact that the majority of computations within a neural network are essentially dot products:

$$\boldsymbol{Y} = \boldsymbol{w}[0]*\boldsymbol{x}[0]+\boldsymbol{w}[1]*\boldsymbol{x}[1] + \ldots..\boldsymbol{w}[\boldsymbol{n}-1]*\boldsymbol{x}[\boldsymbol{n}-1]$$

Here '$\boldsymbol{w}$' and '$\boldsymbol{x}$' signify two vectors, resulting in '$\boldsymbol{Y}$' in the form of a scalar. With respect to convolutional layers and densely connected layers '$\boldsymbol{w}$' denotes the learned weights and '$\boldsymbol{x}$' corresponds to the layer input.' $\boldsymbol{Y}$' delivers a single layer's result. In the majority of cases, layers contain multiple outputs, hence multiply dot products are computed accordingly.

The 'accumulation' process refers to addition, as the outputs of all resultant multiplications are summed. The first equation comprises of 'n' MACCs. Therefore, a dot product between two vectors of 'n' dimensions utilize 'n' MACC's. A single MACC is shown below:

$$\boldsymbol{w}\,[0] * \boldsymbol{x}\,[0] \ldots$$

With respect to the FLOPS, a dot product contains '2n-1' FLOPS as there are 'n' multiplications and 'n-1' additions. Hence,

$$\boldsymbol{MACC} = 2*\boldsymbol{FLOPS}$$

The widespread utilization of Multiply-accumulate makes it easier for hardware platforms to implement 'fused' multiply-add operations, where a single MACC is known as a single instruction. This is viewed as an effective parameter for model evaluation, establishing the feasibility of deployment of the developed architecture in an edge compute environment.

F. POST DEPLOYMENT EVALUATION

The final comparison is based on deployment performance. Realizing the lack of deployment-based architectural comparison in the present literature and to provide a comprehensive evaluation, the performance of each architecture was evaluated on: Frames Per Second (FPS) and Latency, both measured in seconds. The results are presented in Figure 20. Starting with the FPS we observe that inference speed of the proposed architecture was the greatest compared to all other architectures under investigation, validating the successful achievement presented in the research objective of designing a lightweight but at the same time highly performing architecture. The latency for carrying out an inference on an image from the test set was recorded at 0.40 seconds, also the highest performance across all evaluated architectures.

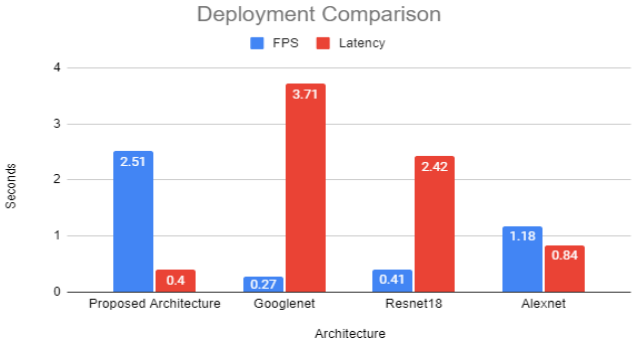

FIGURE 20. SOTA Deployment comparison.

Both top performing architectures with respect to the training metrics (Table 7) performed poorly during the deployment evaluation. GoogleNet was the worst performer for the deployment comparison, achieving 0.27 FPS and a latency of 3.71 seconds. Since the FPS and latency were evaluated on a CPU device, it may be argued that, by replacing the CPU with GPU hardware, the results can be improved, i.e., increased FPS. Although this is possible, the objective of the research is to develop a lightweight architecture that can be deployed on standard CPU hardware, limiting the cost and requirements of new hardware for the end user.

AlexNet, designated as an outlier due to having an accuracy of 56.87%, came in second place with respect to FPS and latency. However, this was not significant as the training performance evaluation (Table 6) demonstrated the architecture as a random classifier, which was unable to generalize from the dataset.

IV. DISCUSSION

The research was initiated with the exploration of exudates. By understanding the domain specific technical details of how exudates are formed, their visual appearance, size, and color we were able to propose certain applicable image processing techniques.

We also investigated the practical procedure for capturing fundus images within a practice. By exploring the two types of devices (Static & Mobile) used for capturing retinal fundus images, we were able to model the output images via the proposal of specific processing techniques. The proposed techniques were not only aimed at scaling the exudate dataset presented in Table 1 but also applying representative augmentations.

Once a representative dataset had been generated, we proposed a CNN development framework (Figure 9), the logic of which was based on a two-stage verification mechanism. The first mechanism would ensure that the designed CNN was computationally lightweight with respect to its internal

TABLE 8. Model performance comparison.

	Architectural		Deployment		Validation
Model	GMACs	Param	FPS	Latency	F1
Proposed	0.03	1.40	2.51	0.40	99.51%
GoogleNet	1.51	13.0	0.27	3.71	100%
AlexNet	0.72	61.1	1.18	0.84	72.50%
Resnet-18	1.82	11.69	0.41	2.42	100%

layers, before progressing to the training stage. This was done via a direct comparison of a number of learnable parameters of the designed architecture against reference SOTA architectures. The latter verification mechanism would evaluate the accuracy of the trained architecture against a threshold of 70%. The proposed design flow mechanism (DFM) enabled us to not only provide a lightweight architecture but one that was also highly performant.

This research provided evaluation of the proposed architecture on a broad range of metrics, in support of the research objectives. Conventionally, architecture evaluation is restricted to standard performance metrics such as precision, recall and F1-score. Although these would be sufficient in validating the first part of our research objective i.e., the development of a high performant architecture, they would not provide any benchmarks for assessing the second part of the research objective i.e., lightweight architecture. Hence, architectural, computational and post-deployment metrics (Fps and latency) were introduced for this purpose.

Table 8 provides a summary of the evaluation of the proposed architecture against SOTA architectures. At the outset, the generalization capacity of the architectures was evaluated via the F1-score. Although, the proposed architecture provided high performance achieving 99.51%, it was not the optimal performer as ResNet-18 and GoogleNet both achieved an F1-score of 100%. However, as the evaluation mechanism was broadened, it was clear that the proposed architecture was the most effective in terms of its internal architectural composition, and post deployment. As shown in Table 8, from the 5 key evaluation metrics, the proposed architecture provided the highest performance in 4 out of 5. When looking deeper at the 5th metric i.e., F1-score it can be observed that the difference was negligible i.e., 0.49% with respect to ResNet-18 and GoogleNet.

V. CONCLUSION

In conclusion, our research has successfully achieved its objective of devising a computational architecture that is not only lightweight but also adept at exhibiting generalizability across the domain dataset. This achievement underscores the pivotal role played by a thorough examination of the pragmatic processes entailed in data acquisition. It emphasizes how these practical facets can be seamlessly incorporated into the design phase, as demonstrated by the innovative Retinal-based Affine Mapping transformations we have introduced.

Furthermore, we hold the firm belief that the successful conception and rigorous testing of our Design Flow Mechanism (DFM) will bestow a valuable tool upon developers working within the realm of automated defect and disease detection. The implementation of DFM into their respective applications is poised to offer a two-fold advantage. Firstly, it establishes a robust framework for crafting bespoke architectures that not only align with high-performance benchmarks but also remain attuned to the intricate landscape of internal computational intricacies inherent in architectural development. Secondly, it facilitates an effective system for ensuring the efficiency and efficacy of these architectures.

Significantly building upon our prior work [4], which delved into the nuanced realm of filter count determination for custom architectures targeting Micro-crack detection in Photovoltaic cells, our current endeavor takes a remarkable leap forward. We introduce a two-tier validation mechanism that meticulously examines and validates the lightweight attributes and exceptional performance potential of the architecture.

To further improve our research we aim to explore hyperparameter tuning post integration into the DFM module, this would allow for further improvement in generalization on a wider range of applications in particular those manifesting subtle and complex defects.

Lastly, our research harmoniously resonates with the scholarly contributions presented by Hussain and Hill [23], which pivot around defect detection within the manufacturing sector. Their emphasis on deploying CNN architectures on edge devices for proximity-based inference dovetails seamlessly with our own aspirations. Our envisioned extension of the proposed DFM within such contexts holds the promise of ushering in a new era of computationally efficient architectures, primed for edge device deployment. This signifies a departure from the conventional reliance on specialized GPU hardware or cloud-based resources for inference, opening up avenues for streamlined, on-device processing.

REFERENCES

[1] M. D. Abràmoff, M. K. Garvin, and M. Sonka, "Retinal imaging and image analysis," *IEEE Rev. Biomed. Eng.*, vol. 3, pp. 169–208, 2010, doi: 10.1109/RBME.2010.2084567.

[2] M. Hussain, H. Al-Aqrabi, M. Munawar, R. Hill, and S. Parkinson, "Exudate regeneration for automated exudate detection in retinal fundus images," *IEEE Access*, vol. 11, pp. 83934–83945, 2022, doi: 10.1109/ACCESS.2022.3205738.

[3] T. A. Chowdhury, D. Hopkins, P. M. Dodson, and G. C. Vafidis, "The role of serum lipids in exudative diabetic maculopathy: Is there a place for lipid lowering therapy?" *Eye*, vol. 16, no. 6, pp. 689–693, Nov. 2002, doi: 10.1038/sj.eye.6700205.

[4] M. Hussain, M. Dhimish, V. Holmes, and P. Mather, "Deployment of AI-based RBF network for photovoltaics fault detection procedure," *AIMS Electron. Electr. Eng.*, vol. 4, no. 1, pp. 1–18, 2020, doi: 10.3934/electreng.2020.1.1.

[5] M. Hussain, H. Al-Aqrabi, M. Munawar, and R. Hill, "Feature mapping for rice leaf defect detection based on a custom convolutional architecture," *Foods*, vol. 11, no. 23, p. 3914, Dec. 2022, doi: 10.3390/foods11233914.

[6] M. Hussain and H. Al-Aqrabi, "Child emotion recognition via custom lightweight CNN architecture," in *Kids Cybersecurity Using Computational Intelligence Techniques* (Studies in Computational Intelligence), vol. 1080, W. M. S. Yafooz, H. Al-Aqrabi, A. Al-Dhaqm, and A. Emara, Eds. Cham, Switzerland: Springer, 2023, doi: 10.1007/978-3-031-21199-7_12.

[7] M. Abdullah, M. M. Fraz, and S. A. Barman, "Localization and segmentation of optic disc in retinal images using circular Hough transform and grow cut algorithm," *PeerJ*, vol. 4, p. e2003, May 2016, doi: 10.7717/peerj.2003.
[8] M. M. Habib, R. A. Welikala, A. Hoppe, C. G. Owen, A. R. Rudnicka, and S. A. Barman, "Detection of microaneurysms in retinal images using an ensemble classifier," *Informat. Med. Unlocked*, vol. 9, pp. 44–57, Jan. 2017, doi: 10.1016/j.imu.2017.05.006.
[9] R. Murugan and P. Roy, "MicroNet: Microaneurysm detection in retinal fundus images using convolutional neural network," *Soft Comput.*, vol. 26, no. 3, pp. 1057–1066, Jan. 2022, doi: 10.1007/s00500-022-06752-2.
[10] J. H. Tan, H. Fujita, S. Sivaprasad, S. V. Bhandary, A. K. Rao, K. C. Chua, and U. R. Acharya, "Automated segmentation of exudates, haemorrhages, microaneurysms using single convolutional neural network," *Inf. Sci.*, vol. 420, pp. 66–76, Dec. 2017, doi: 10.1016/j.ins.2017.08.050.
[11] S. Guo, "LightEyes: A lightweight fundus segmentation network for mobile edge computing," *Sensors*, vol. 22, no. 9, p. 3112, Apr. 2022, doi: 10.3390/s22093112.
[12] S. Huang, J. Li, Y. Xiao, N. Shen, and T. Xu, "RTNet: Relation transformer network for diabetic retinopathy multi-lesion segmentation," *IEEE Trans. Med. Imag.*, vol. 41, no. 6, pp. 1596–1607, Jun. 2022, doi: 10.1109/TMI.2022.3143833.
[13] Y. Zhou, B. Wang, L. Huang, S. Cui, and L. Shao, "A benchmark for studying diabetic retinopathy: Segmentation, grading, and transferability," *IEEE Trans. Med. Imag.*, vol. 40, no. 3, pp. 818–828, Mar. 2021, doi: 10.1109/TMI.2020.3037771.
[14] R. Rosas-Romero, J. Martínez-Carballido, J. Hernández-Capistrán, and L. J. Uribe-Valencia, "A method to assist in the diagnosis of early diabetic retinopathy: Image processing applied to detection of microaneurysms in fundus images," *Computerized Med. Imag. Graph.*, vol. 44, pp. 41–53, Sep. 2015, doi: 10.1016/j.compmedimag.2015.07.001.
[15] J. Shan and L. Li, "A deep learning method for microaneurysm detection in fundus images," in *Proc. IEEE 1st Int. Conf. Connected Health, Appl., Syst. Eng. Technol. (CHASE)*, Jun. 2016, pp. 357–358.
[16] L. Tang, M. Niemeijer, J. M. Reinhardt, M. K. Garvin, and M. D. Abramoff, "Splat feature classification with application to retinal hemorrhage detection in fundus images," *IEEE Trans. Med. Imag.*, vol. 32, no. 2, pp. 364–375, Feb. 2013, doi: 10.1109/TMI.2012.2227119.
[17] T. Kauppi, V. Kalesnykiene, J.-K. Kamarainen, L. L. I. Sorri, A. Raninen, R. Voutilainen, J. Pietilä, H. Kalviainen, and H. Uusitalo, "DIARETDB1 diabetic retinopathy database and evaluation protocol," in *Proc. Med. Image Understand. Anal.*, 2007, pp. 61–65.
[18] EyePACS. (2015). *Diabetic Retinopathy Detection*. [Online]. Available: https://www.kaggle.com/c/diabeticretinopathy-detection/data
[19] K. Simonyan and A. Zisserman. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. [Online]. Available: https://www.robots.ox.ac.uk/~vgg/publications/2015/Simonyan15/simonyan15.pdf
[20] M. Gao, P. Song, F. Wang, J. Liu, A. Mandelis, and D. Qi, "A novel deep convolutional neural network based on ResNet-18 and transfer learning for detection of wood knot defects," *J. Sensors*, vol. 2021, pp. 1–16, Aug. 2021, doi: 10.1155/2021/4428964.
[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
[22] L. Nanni, S. Brahnam, M. Paci, and S. Ghidoni, "Comparison of different convolutional neural network activation functions and methods for building ensembles for small to midsize medical data sets," *Sensors*, vol. 22, no. 16, p. 6129, Aug. 2022, doi: 10.3390/s22166129.
[23] M. Hussain and R. Hill, "Custom lightweight convolutional neural network architecture for automated detection of damaged pallet racking in warehousing," *IEEE Access*, vol. 11, pp. 58879–58889, 2023, doi: 10.1109/ACCESS.2023.3283596.

MUHAMMAD HUSSAIN was born in Dewsbury, West Yorkshire, U.K., in 1995. He received the B.Eng. degree in electrical and electronic engineering and the M.S. degree in Internet of Things from the University of Huddersfield, U.K., in 2019, where he is currently pursuing the Ph.D. degree in artificial intelligence for defect identification.

His research is focused on the detection of various faults in particular micro-cracks forming on the surface of photovoltaic (PV) cells because of mechanical and thermal stress. His research interest includes machine vision, with a focus on the development of lightweight architectures that can be optimized for deployment on edge devices and ultimately on the production floor. He is also researching design-level architectural interpretability, with a focus on explainable AI for sensitive fields, such as medicine and healthcare.