

# Keratoviz-A multistage keratoconus severity analysis and visualization using deep learning and class activated maps

Priya Dhinakaran<sup>1</sup>, Mamatha Gowdra Shivanandappa<sup>1</sup>, Srijan Devnath<sup>3</sup>

Department of Information Science and Engineering, RV College of Engineering, Bangalore, India

## Article Info

### Article history:

Received Nov 27, 2021

Revised Aug 3, 2022

Accepted Aug 25, 2022

### Keywords:

Class activated map

Deep learning

Keratoconus

Topography map

Visualization

## ABSTRACT

The detection of keratoconus has been a difficult and arduous process over the years for ophthalmologists who have devised traditional approaches of diagnosis including the slit-lamp examination and observation of thinning of the corneal. The main contribution of this paper is using deep learning models namely Resnet50 and EfficientNet to not just detect whether an eye has been infected with keratoconus or not but also accurately detect the stages of infection namely mild, moderate, and advanced. The dataset used consists of corneal topographic maps and pentacam images. Individually the models achieved 97% and 94% accuracy on the dataset. We have also employed class activated maps (CAM) to observe and help visualize which areas of the images are utilized when making classifications for the different stages of keratoconus. Using deep learning models to predict the detection and severity of the infection can drastically speed up and provide accurate results at the same time.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Priya Dhinakaran

Department of Information Science and Engineering, RV College of Engineering

Mysore Road, Pattangere Metro Station, India

Email: priyad@rvce.edu.in

## 1. INTRODUCTION

Keratoconus is a disease which can be identified by the subsequent thinning process of the cornea. It is a relatively common disorder that often involves multiple layers of the cornea, resulting in severe myopia and astigmatism. The loss of thickness of the cornea is often the result of collagen fibrils that is missing from the eye [1], [2]. The primary causes of Keratoconus are mostly associated with rubbing of the eye, tissue disease or any genetic history. The slit-lamp exam of the central and posterior cornea leads to the appropriate diagnosis can be made [3]. Algorithms that make use of video keratography allow for the detection of keratoconus [4]. These devices allow better screening of patients for surgery. Detection at an early stage helps patients avoid complicated intrusions and procedures [5], [6]. In a nutshell diagnosis is made based on the history of changing refraction, poor best spectacle-corrected vision, scissoring reflex on retinoscopy, abnormalities in keratometry, corneal topography, and corneal tomography, in association with corneal thinning and inferior steepening; characteristic slit-lamp findings [7], [8].

Keratoconus is a lifelong eye disease as it stands today, with no immediate cure in sight. However, it can be managed effectively depending on which stage it is currently at for the person infected [9]. Scleral contact lenses typically are the treatment for mild to moderate keratoconus [10]. Corneal cross-linking (CXL) is a non-invasive process used for advanced stages of keratoconus which can reinforce and provide support to a thinning cornea. Keratoconus however is not cured by this process. In other words, CXL strengthens and may stabilize the cornea, but it doesn't return the cornea to normal thickness [11].

Recently, deep learning models involving image convolutions have been frequently used to assist ophthalmologists primarily quickly and accurately in the field of medical imaging analysis, where deep learning techniques have shown great performance in detecting various ocular imaging, optical coherence tomography (OCT), glaucoma, age-specific macular degeneration (AMD), and retinopathy of prematurity (ROP) [12]. Convolutional neural networks (CNN's) have demonstrated their ability in recognizing images without the need for indices input for training, thereby showing usability for pattern recognition of colored corneal topography maps [13]. As a result, an ensemble of Resnet50 and EfficientNet will be used which are two very different but powerful deep neural models which are known to be incredibly effective in image classification tasks, both inside and outside the medical field [14]–[16].

Deep learning models and techniques have produced incredible results in the field of healthcare especially in the detection and classification of various diseases and abnormalities. These models and techniques have helped doctors and specialists in not just faster detection and diagnosis, but also quicker treatment and prediction of further complications. This also has helped immensely indirectly in other ambits, like to provide medical care to the patients and at a fair price, to discover better methods of treatment, the creation of innovative medical programs, the success of the treatment, healthcare, infrastructure, hospitals, infection control, and the improvement of patient care, as well as the retention and overall better customer relationship management. Some of the following notable and recent publications have been studied and the insights are given below.

In 2021, research done by [17] led to publication of a paper on a hybrid deep learning construct for detecting Keratoconus wherein the dataset used comprised of corneal topographic maps. Earlier in 2020, a predictive modelling research to identify early stage keratoconus was conducted and simultaneously a paper was published by the contributors of [18]. The contributors in [19] have published a paper in 2021 highlighting SmartKC: a low-cost, smartphone-based keratoconus diagnosis system comprising of a 3D-printed Placido's disc attachment, an light-emitting diode (LED) light strip, and an intelligent smartphone app to capture the reflection of the Placido rings on the cornea. Research conducted in 2021 by [20] has led to presentation of a paper on diagnosability of keratoconus using deep learning with Placido disk-based corneal topography which shows the wide range of data being used for training the various models.

Contribution in [21] led to the implementation and application of deep learning techniques to develop an intelligent Pterygium diagnosis system. Also in 2021, contributors of [22] used disk-based corneal topography as their primary dataset on which they applied deep learning models to detect the presence of keratoconus. A survey on artificial intelligence (AI) in ophthalmology and specifically in the area of keratoconus classification was conducted in 2021 by [23]. Another survey was performed in the same year by [24] which examined machine learning techniques for the diagnosis of corneal diseases. Last but not the last, research conducted in [25] led to the publication of a paper about sensitivity and specificity of Sirius indices in diagnosis of keratoconus and the detection of early-stage keratoconus. Around the concerned field of ophthalmology, many have published works regarding the usage of deep learning models in the binary detection of keratoconus. The research conducted in this paper pertains to not just the detection and presence of keratoconus but also part of a study to categorize and classify it in increasing stages of severity.

## 2. RESEARCH METHOD

The research method adopted aims to achieve the objective of classifying the severity and amount of keratoconus infection as discussed above through a series of deep learning models. The models considered are i) Resnet50 and ii) EfficientNet. The models have been trained on a pre-labelled dataset consisting of pentacam and corneal topographic maps.

### 2.1. Data collection and augmentation

Data collection involves the aggregation of various relevant data sources to compile a final collection which is used in the analysis process that succeeds it. Upon detection of keratoconus and its varying severity levels, maximal keratometry (Kmax), central corneal thickness (CCT) and thinnest corneal thickness (TCT) [26] are the most relevant parameters that help identify it. In this case, the dataset has pentacam and corneal topographic indices to help detect hot spots that help to identify the severity of keratoconus. The first step taken was data preprocessing for the given dataset. It consists of 4 different classes-normal, mild, moderate, and advanced representing the stages of infection. Each class folder contains around 550 images.

The Figure 1 images represent the severities of the keratoconus disease on the eye. Starting from the first image, one can gradually make out an increasing pattern of corneal topographic indices around the center of the eye which increase in intensity, corresponding to the severity level. The most advanced stage has a hotspot around the center where the maximum convergence takes place regarding the index values. To start off processing the dataset, augmentation was performed on the dataset to increase its size. The type of

augmentation was limited to changing the shear length, increasing the zoom level, and performing very minimal rotational shift to maintain the accuracy of the image and its desired class. The size of each class data has been increased from an initial 500 to 5000. Upon using shear augmentation for transforming an image, the image will be shifted along either the y axis or the x axis and it done to rectify and change the angles as shown in Figure 2. It is used to mimic the views of the images from different angles and shear helps distort images for the computer to identify these different angles of the image.

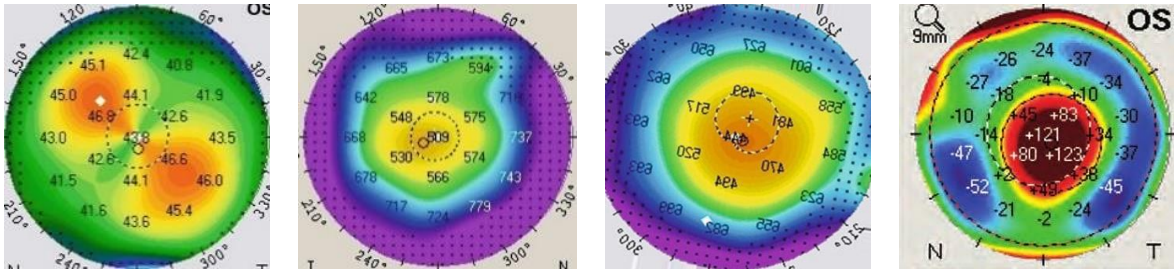


Figure 1. Images representing the different classes and stages of keratoconus, from left to right, normal, mild, moderate, and advanced

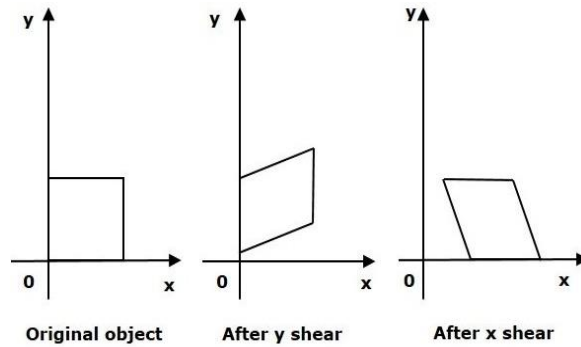


Figure 2. The effect of shear done along the y axis and the x axis for any object [26]

Rotation is a straightforward augmentation method in which the input image is rotated through a predefined angle either clockwise or anti-clockwise. Unlike shear, rotation doesn't distort the image but causes a change in the image being viewed from a non-perspective angle as shown in Figure 3. It doesn't depend on the axes for rotation as it has a fixed pivot point which is often times the center of the image.

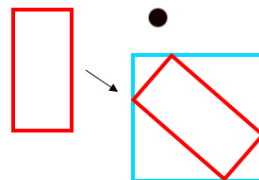


Figure 3. The effect of rotation using a fixed pivot point, being the top right of the image [26]

Lastly, zooming helps focus on a certain part of the image until a specified range and crops the remaining part of the picture. It helps train the model by forcing it to focus on randomly selected areas of the image which helps it make better generalized decisions on the whole image. Thus, the results of augmentation performed on the dataset are.

It can be clearly observed that the effect of image augmentation on the above sample image as shown in Figure 4. A standard scale pentacam image has been put through a series of transformations all

while keeping the original meaning intact. Rotation involves rotating the image at a certain specified angle, zoom involves focusing on a certain part of the image and shear involves a slight shift towards a certain axis. The dataset consists of pentacam and corneal topographic images across a varied sample with indices labeling for reference to the corneal values present in the sample. The indices help with the initial classification of the images in the 4 different classes.

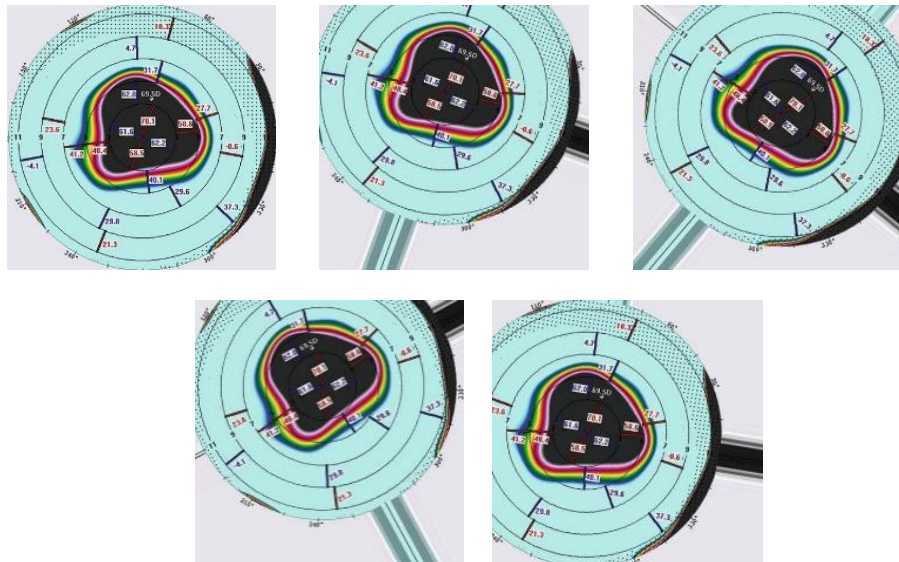


Figure 4. Original image at the top, augmented images involving rotation, zoom and shear

**2.2. Models used**

We have used two very popular deep learning models such as ResNet50 and EfficientNet on the preprocessed dataset. Residual networks (ResNet) are a neural network which has been deployed numerous times for image classification tasks. ResNet involves the technique of skip connections where the original input blocks are also fed into subsequent convolutional layers as shown in Figure 5. These connections are used to add the current output to an output from a previous layer. The vanishing gradient problem, which is a big reason why experts were skeptical of using very deep neural networks, is mitigated by these connections [27]. The ResNet-50 model consists of a convolution and an identity block, which is present over 5 layers. The ResNet-50, which stands for 50 skip connections, has over 23 million trainable parameters. ResNets and their framework are the reason it is possible to train ultra-deep neural networks [28].

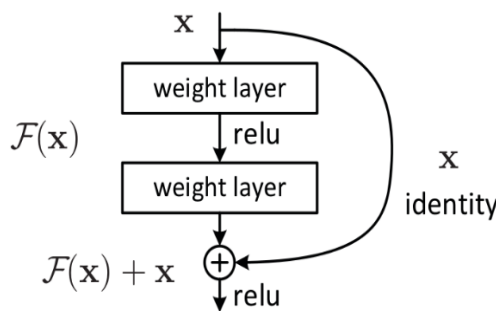


Figure 5. Skip connections connect the input layers to the subsequent convolutional blocks, acting as simple identity mappings and acting as a shortcut connection [28]

As one can clearly observe from the Table 1, ResNet50 the model has a number of convolutional layers stacked together and interconnected using the skip connections. The 50 in the Resnet50 stands for the total number of skip connection layers present in the architecture. When compared to the traditional Resnet18

or Resnet34, one can observe the depth of the 50 being multiple layers more than the former. Apart from the convolutional layers, one can regularly apply max pooling to reduce down the feature size and get important features highlighted.

Table 1. Resnet50 architecture as compared to the other layer models

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112x112			7x7, 64, stride 2		
conv2_x	56x56			3x3 max pool, stride 2		
		{3x3,64} x2	{3x3,64} x3	{1x1,64}	{1x1,64}	{1x1,64}
		{3x3,64}	{3x3,64}	{3x3,64} x3	{3x3,64} x3	{3x3,64} x3
				{1x1,256}	{1x1,256}	{1x1,256}
conv3_x	28x28	{3x3,128} x2	{3x3,128} x4	{1x1,128}	{1x1,128}	{1x1,128}
		{3x3,128}	{3x3,128}	{3x3,128} x4	{3x3,128} x4	{3x3,128} x8
				{1x1,512}	{1x1,512}	{1x1,512}
conv4_x	14x14	{3x3,256} x2	{3x3,256} x6	{1x1,256}	{1x1,256}	{1x1,256}
		{3x3,256}	{3x3,256}	{3x3,256} x6	{3x3,256} x23	{3x3,256} x36
				{1x1,1024}	{1x1,1024}	{1x1,1024}
conv5_x	7x7	{3x3,512} x2	{3x3,512} x3	{1x1,512}	{1x1,512}	{1x1,512}
		{3x3,512}	{3x3,512}	{3x3,512} x3	{3x3,512} x3	{3x3,512} x3
				{1x1,2048}	{1x1,2048}	{1x1,2048}
	1x1			average pool, 1000-d fc, softmax		
FLOPs		1.8x109	3.6x109	3.8x109		7.6x109

EfficientNet on the other hand makes use of a scaling method which spans across the basic attributes of the neural network like the height, depth, width, and inner layer depth. The compound scaling method works on the assumption that the bigger the input image, the more layers the model has to accommodate for the receptive field. The models sometimes accidentally are either too deep, too wide, or too lengthy in terms of density. When there is an increase in the value of these characteristics it helps the model initially but now the model has more parameters but is more inefficient. In EfficientNet they are scaled in a way where gradually everything is increased [29].

We can make out the differences between the different attributes being scaled according to their scaling methods as shown in Figure 6. We can compare the different scaling methods with the baseline architecture in Figure 6(a). We can observe width scaling which ensure a wider model in terms of adding more hidden layers in Figure 6(b). We can observe depth scaling in Figure 6(c) which highlights on increasing number of layers within the model while Figure 6(d) touches on resolution scaling which increases the model complexity within a particular layer. Finally compound scaling in Figure 6(e) makes sure that we apply the resolution scaling is applied onto all the different layers in our model. Width scaling increases the width of the model and its corresponding layers, depth scaling increases the subsequent layers present in the model, resolution scaling increases the input intensity of the model to capture a higher resolution image and compound scaling adds a compounding effect to the original layers of the model. The formula for compound scaling is defined for the scaled attributes.

$$\begin{aligned}
 \text{Depth: } & -d = a^\phi \\
 \text{Width: } & -w = \beta^\phi \\
 \text{Resolution: } & -r = \gamma^\phi \\
 \text{s.t } & a, \beta^2, \gamma^2 \sim 2 \\
 & a > 1, \beta > 1, \gamma > 1
 \end{aligned}
 \tag{1}$$

The alpha, beta, and gamma values in the above formula are known as the scaling multiplier for depth, width, and resolution respectively. The scaling method is named compound scaling, which means that primary attributes of the model namely height, depth and width are scaled together, instead of independently to increase accuracy [30]. There are various EfficientNet models ranging from B0 to B7, the former being the base model developed by AutoML MNAS and the latter being a series of scaled up versions emerging from the base model themselves. The total number of layers in the base B0 model is about 237 while the B7 contains around 817 [31]. The entire B0 architecture is represented in the Figure 7.

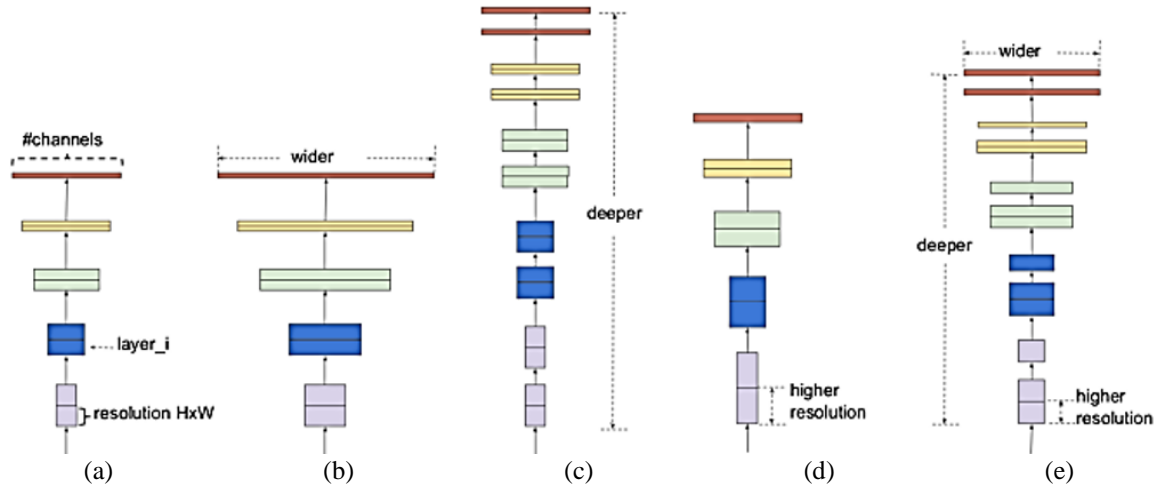


Figure 6. Comparison of the different scaling methods based on the different attributes. From left to right, (a) baseline model, (b) scaling the width, (c) depth scaling, (d) resolution scaling and (e) compound scaling [29]

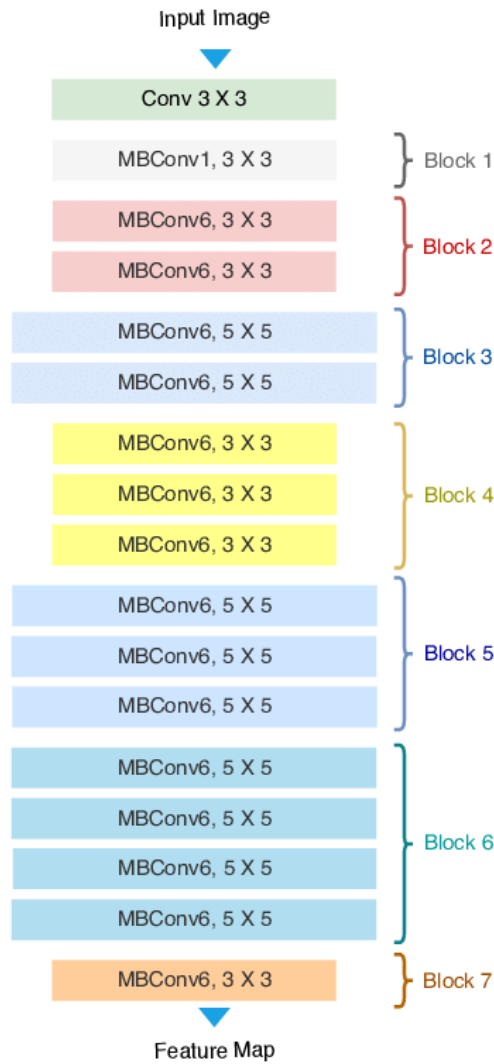


Figure 7. EfficientNet B0 architecture [31]

### 2.3. Training phase

The implementation involving Resnet50 model includes the following steps:

- i) The ResNet50 model is imported from TensorFlow Keras.
- ii) A base model is saved by downloading Resnet50 trained on ImageNet dataset to get the pretrained weights and the top layer is removed to add a custom input layer.
- iii) A GlobalAveragePooling2D is initiated thereafter which reduces the feature values by taking the average values of a convolution map as shown in Figure 8.
- iv) A dense layer is added after with 1024 layers with 'ReLU' activation function. Rectified linear unit (ReLU) or helps classify the input directly or return a zero-value depending on the sign of the input. It also helps avoid the common vanishing gradient error which is the reason why its preferred over other activation functions as represented in Figure 9.

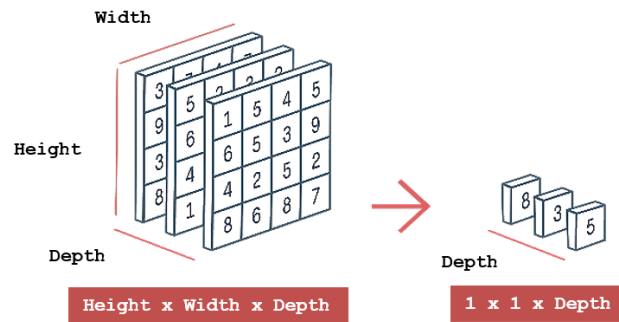


Figure 8. The architecture and visualization of global average pooling 2D

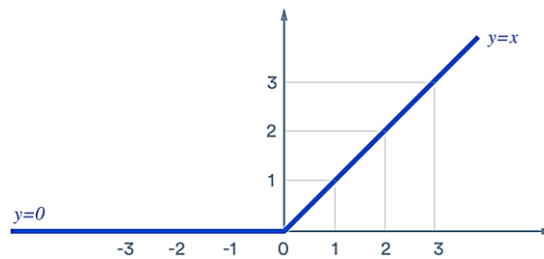


Figure 9. The ReLU activation function, giving a positive value only when the input is non-negative

- v) The final dense layer contains the 4 output classes and softmax as the activation function as the necessity is multiclass classification for which softmax is preferred over Sigmoid as the latter is used for binary classification.

$$s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \quad (2)$$

The above formula is the mathematical definition of Softmax function where  $Z_i$  values are the input values of the model and  $K$  is the number of output classes. In model compilation, Adam was used as the optimizer, sparse categorical cross entropy as the loss function and validation accuracy as the metric. The model was trained for 50 epochs, holding a validation dataset out as well. A snippet of the final model is presented as shown in Figure 10.

The implementation involving EfficientNet model includes the following steps:

- a. The EfficientNet B0 model is installed from Keras.
- b. A base model is saved by downloading EfficientNet B0 trained on ImageNet dataset to get the pretrained weights and the top layer is removed to add a custom input layer of size  $56 \times 56$ .
- c. A custom python function for the model is initialized with number of classes as the parameter.
- d. The model's weights are freeze so that the initial weights can be reused to compile the model faster along with a layer of GlobalAveragePooling2D added after.

- e. A BatchNormalization layer is added in next which effectively normalizes the output of a previous layer of a CNN which becomes another layer’s input as shown in Figure 11. In this way, the model is trained more efficiently and faster as every batch is normalized and trained independently.
- f. A dropout layer is added which randomly skips over a specified percentage of input values during training of the model as represented in Figure 12. This helps to regularize the model and help prevent overfitting. Its drops random neuron values so every step is randomized. In this model we have specified 20% rate of dropout.
- g. The final dense layer contains the 4 output classes and Softmax as the activation function as the necessity is multiclass classification.
- h. A snippet of the final EfficientNet B0 model is presented as shown in Figure 13.

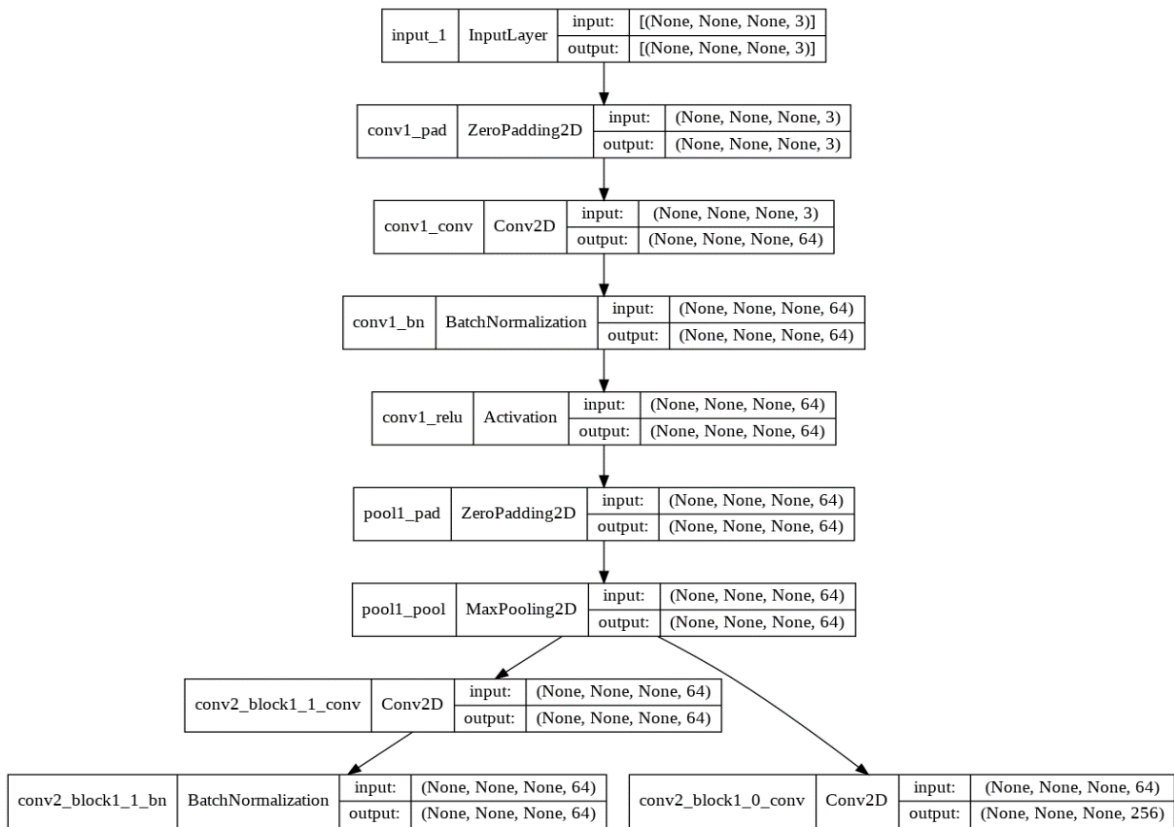


Figure 10. The starting few layers of the custom Resnet50 model showing the different layers added. The model in total contains 40 layers

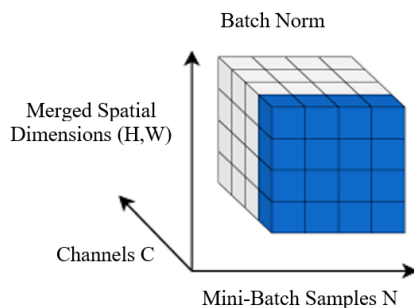


Figure 11. BatchNormalization normalizes the input layer to every batch present in the CNN model, effectively speeding up model training [32]

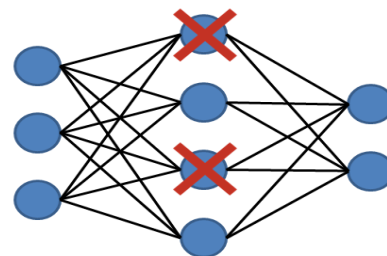


Figure 12. Dropout randomly drops a percentage of neurons in each layer. This helps to regularize the model and avoid the overfitting problem, which occurs in training deep neural networks [33]



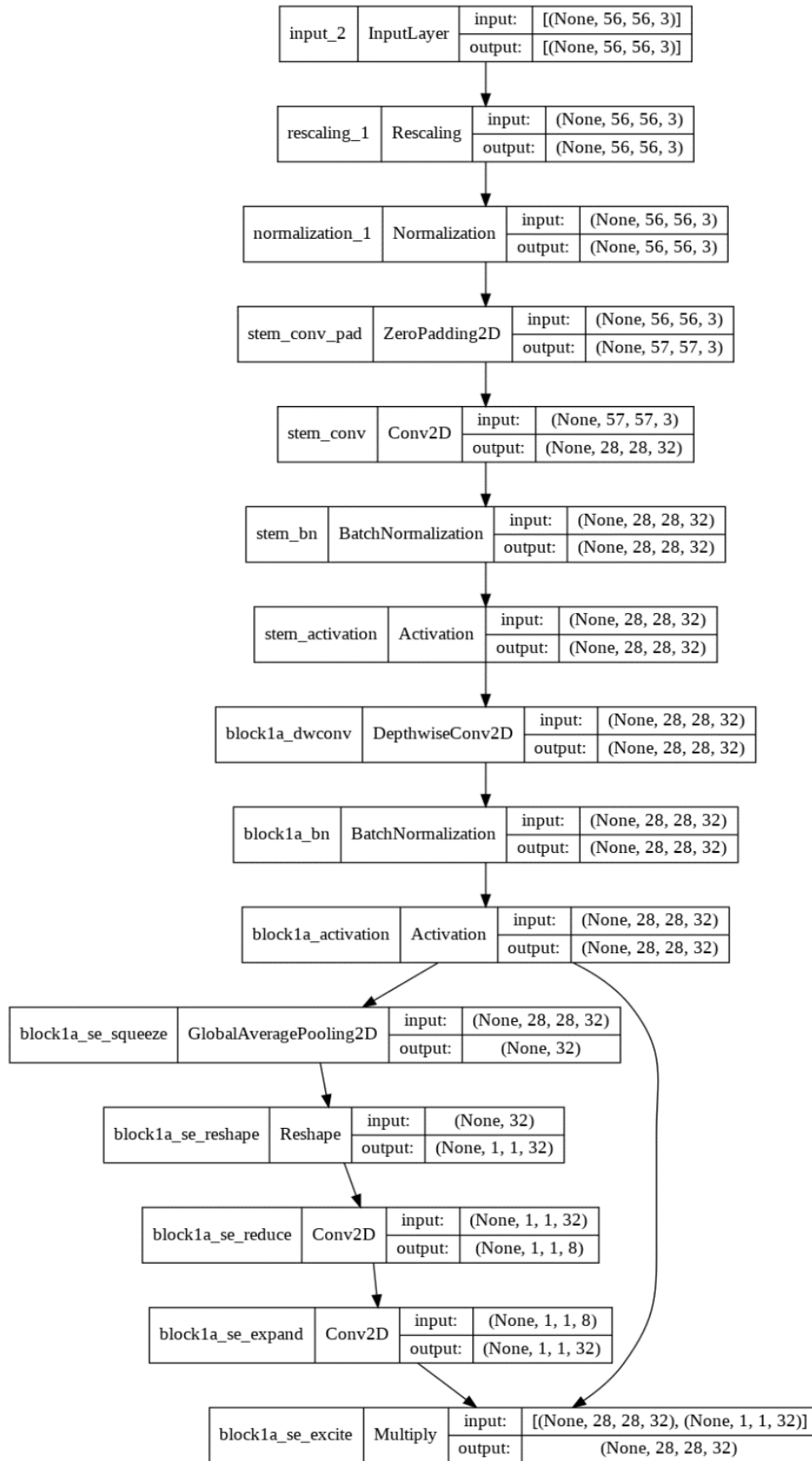


Figure 13. The starting few layers of the custom EfficientNet model showing the different layers added. The model in total has 250 layers

### 3. RESULTS AND DISCUSSION

#### 3.1. Output and Inference graphs for the models used

In Figure 14 the accuracy and loss parameters are noted while using 25 epochs or iterations. One can observe the average training accuracy of 97%-98% and 93%-94% validation accuracy across the epochs. Additionally, the average training loss of the Resnet model is around 0.05 and validation loss is around 0.35 for the 50 epochs. Typically, the loss will go down as the model trains longer and a clear general downward trend is present despite a spike in loss halfway across the model. Every change estimates the variation, which will correspond to updates of the neural system. The accuracy validated is taken for every 50 iterations. The model's loss parameter also has been proven. The pass entropy of the complete model is represented via this parameter. When the model's accuracy is progressed, the loss parameter decreases substantially.

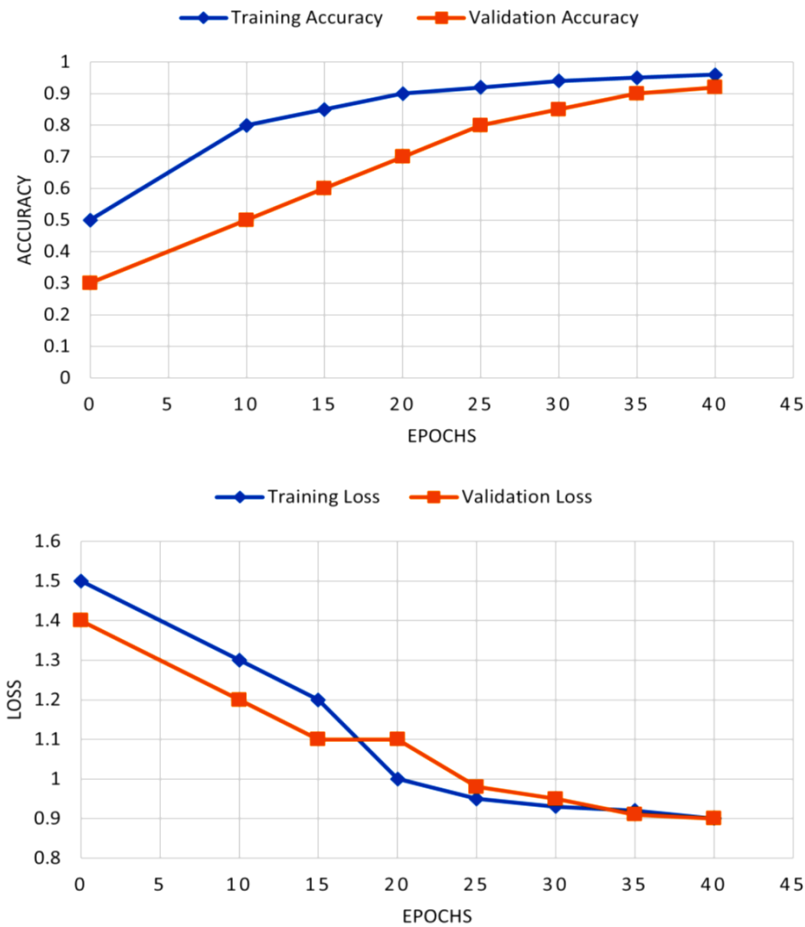


Figure 14. Training and loss values for training and validation data respectively

From the Figure 15 one can observe that the EfficientNet model has higher validation accuracy than the training counterpart around 94% for the former and 92% for the latter. The model generalizes extremely well on the validation data and it's a good sign as the model is neither underfitting nor overfitting. However, the accuracies are comparatively lower and the losses higher than the Resnet50 model. In the longer scope of things through, choosing a model which generalizes well on new and unseen data will be very important for medical purposes.

One can observe that the Resnet50 can achieve comparatively higher training and validation accuracy than the EfficientNet, however the latter is the only one which has a higher validation accuracy than the training counterpart, thus the EfficientNet model generalizes better. Another important observation is that the loss value is consistent with the Resnet50 model, the training has lower loss than the validation part and thus justifies the training data having higher accuracy than the validation one. However, in the EfficientNet model, even though accuracy of the validation data is higher, the loss of the training data is lower than the validation. Thus, the summary of the observations obtained by the two models have been finalized in Table 2.

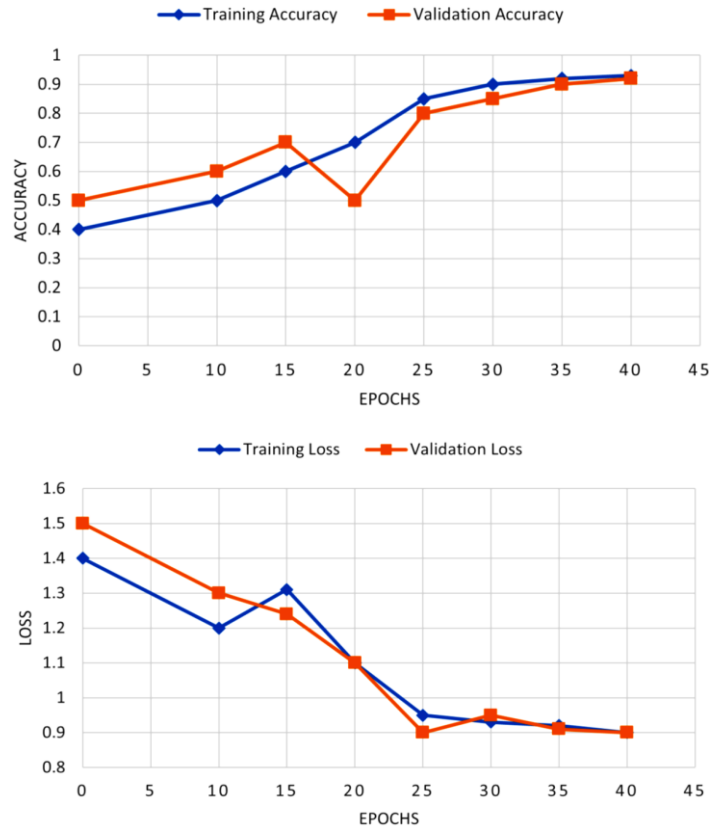


Figure 15. Training and loss values for training and validation data respectively

Table 2 Summary of accuracies obtained by both the models

Model Name	Training Accuracy	Validation Accuracy	Epochs	Training Loss	Validation Loss
Resnet50	98%	95%	25	0.05	0.35
EfficientNet	94%	95%	40	0.4	0.5

**3.2. Grad CAM visualization**

We have obtained great results using both models on our training and testing dataset. However, to better understand the intuition behind the algorithm’s decision making, we will be making use of an algorithm known as grad computer-aided manufacturing (CAM) or gradient class attention maps to visualize the network’s ability to capture features that are important in the task of classifying the images into the 4 different classes/stages of keratoconus. Grad-CAM uses response variables gradient which is fed into the final convolutional layer to produce a rough heat map highlighting the important regions in the image for classifying that image into a particular category. We can implement grad CAM using Keras, namely using the Keras CAM library.

The following are the steps to follow when implementing Grad CAM:

- A custom python function is defined which has to convert the images into array format in order to receive the numerical values for that image which the Grad CAM model can process as shown in Figure 16.
- Another python function is defined for generating the heatmap for the respective image so as to later superimpose it with the actual final image as shown in Figure 17. The heatmap takes into account the feature values of the input image which it decides to either highlight or leave unattended.
- The final python function combines the heatmap to the original image but not without performing some preprocessing as shown in Figure 18. The heatmap is slightly colored and rescaled to a range of 0-255.

Thus, the result for some of the different severities and the feature importance has been shown in the Figure 19. From the Figure 19, the visualizations help us understand the reasoning behind the model’s classification of certain images into the target classes. Figure 19(a) highlights about the advanced severity of keratoconus which has been later visualized in Figure 19(b) using grad CAM. Similarly, we have performed the same for moderate severity level in Figure 19(c) and its corresponding visualization in Figure 19(d). The

highlighted areas focus on the most important features the model focuses on while making the predictions. Certain indices are included in the highlighted areas, indicating the region of importance and the metrics behind the classification. Understanding these regions using pentacam and corneal topographic images will be crucial in quickly and efficiently classifying severity of keratoconus.

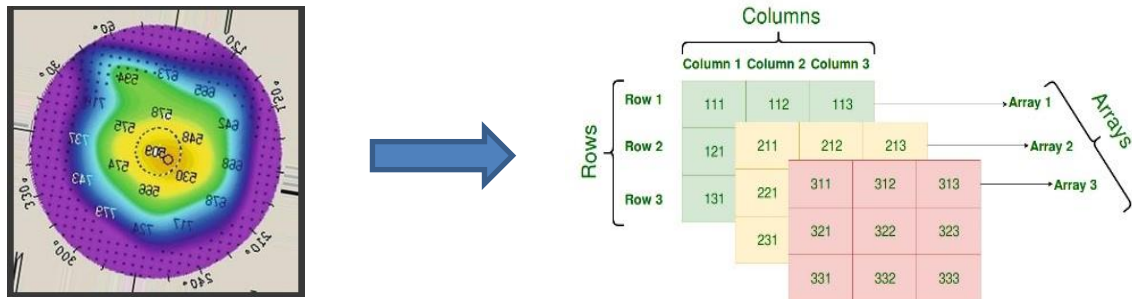


Figure 16. The input 2D image is converted to its corresponding image array [33]

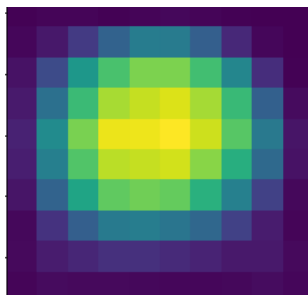


Figure 17. The subsequent python function will generate the corresponding heatmap for the original image [34]

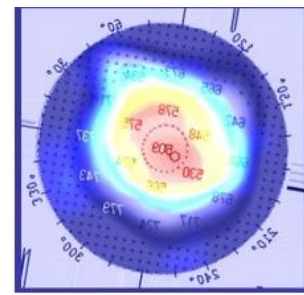


Figure 18. The final function will superimpose the processed heatmap onto the original image

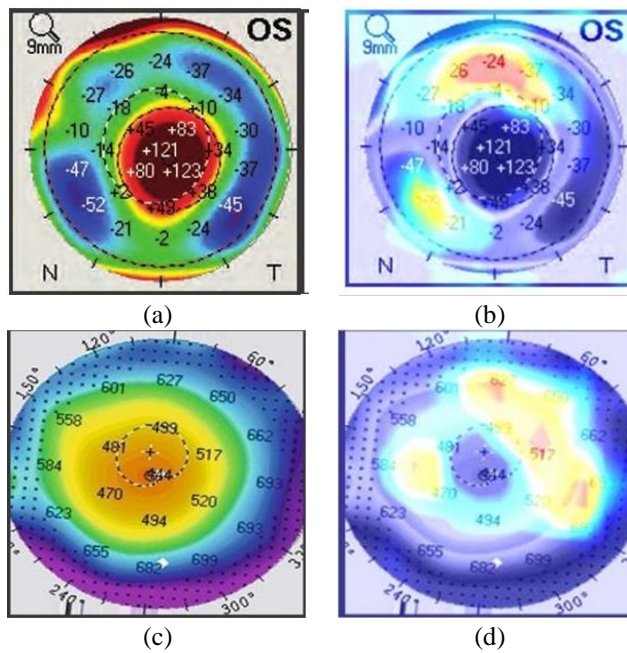


Figure 19. Before and after using grad CAM for detecting the most important features of infected eye, (a) before advanced detection, (b) after advanced detection, (c) before moderate detection and (d) after moderate detection

### 3.3. Score CAM visualization

Score CAM extends on the functionality provided by grad CAM and solves the problem of irrelevant noise and gives a much cleaner understanding of the decision making of the model as represented in Figure 20. The biggest change when compared to the traditional grad CAM is instead of taking into account gradient of the output layer which leads to uncertainty and noise, it uses the weights obtained from the output of a particular class. This ensures it's a more generalized way of detecting the patterns instead of being dependent on the gradient.

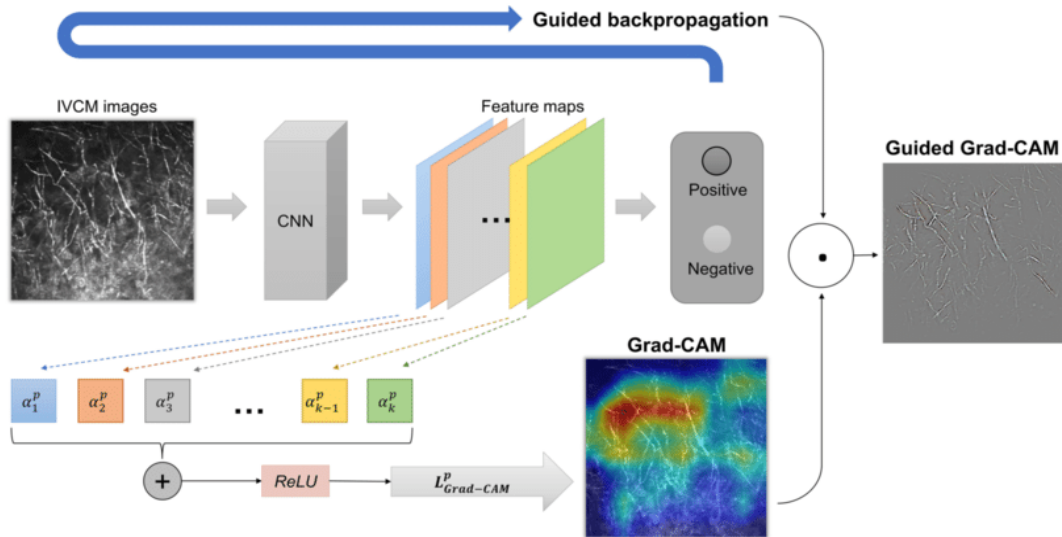


Figure 20. The entire pipeline of score CAM showing the flow till the class discriminative saliency map

The following are the steps to follow when implementing score CAM:

- a. The first step involves training a CNN using the input images and doing a forward pass. The activations are noted from the last convolutional layer.
- b. Every activation map obtained from the last layer is upsampled to the same size as the input image.
- c. The activation maps after upsampling are normalized to maintain the relative intensities between the pixels.

$$A^k_{i,j} = \frac{A^k_{i,j}}{\max(A^k) - \min(A^k)} \tag{3}$$

- d. After the normalization is complete, the highlighted areas of the activation maps are projected on the input space by multiplying each normalized activation map with the input image to obtain a masked image.

$$M^k = A^k \cdot I \tag{4}$$

- e. The masked images are then passed to CNN with SoftMax output

$$S_k = \text{Softmax}(F(M^k)) \tag{5}$$

- f. After getting the scores for each class the score of the target class is extracted to represent the importance of the kth activation map.

$$w^c_k = S^c_k \tag{6}$$

- g. Then the computation of the sum across all the activation maps for the linear combination between the target class score and each activation map is done. This results in a single activation map having the same size as that of the input class.

- h. Finally, ReLU is applied to the pixels of the final activation map. ReLU is applied because only the features that have a positive effect on the target class are required. The final output of score CAM is almost similar to the grad CAM output however it has much cleaner output than the previous grad CAM output as shown in Figure 21.

$$L_{Score-CAM}^c = ReLU(\sum_k^1 w_k^c A^k) \quad (7)$$

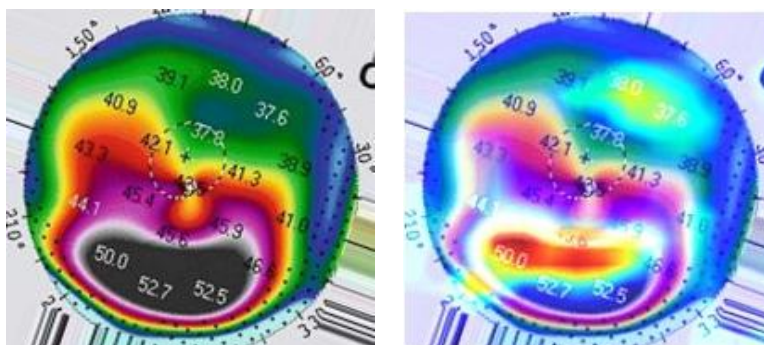


Figure 21. Before and after score CAM implementation

#### 4. CONCLUSION

This research work has proposed using two of the most popular and effective image classification models, ResNet50 and EfficientNet to classify 4 varying stages of keratoconus infection using a dataset consisting of corneal topographic and pentacam images. An average accuracy of 98% was observed with the help of the Resnet50 model and 94% with the EfficientNet model. At the end we have also employed grad CAM heatmaps to understand the most important features taken into considerations by the models while making decisions and classifications. The grad CAM's have helped us visualize clearly where the model pays the most attention while making decisions. The outcome of this study can help in keratoconus (KCN) detection that aids ophthalmologists for better KCN diagnosis and management using a standardized system.

#### ACKNOWLEDGEMENT

The authors would like to thank for the assistance and contribution from the Department of Information Science and Engineering, RV College of Engineering, Bangalore. The authors also would like to acknowledge and thank Dr Gairik, Narayana Netralaya Hospital for his support and insights into this field and study.





#### REFERENCES

- [1] A. H. Al-Timemy, N. H. Ghaeb, Z. M. Mosa, and J. Escudero, "Deep transfer learning for improved detection of keratoconus using corneal topographic maps," *Cognitive Computation*, Jun. 2021, doi: 10.1007/s12559-021-09880-3.
- [2] H. Pratt, "Deep learning for diabetic retinopathy diagnosis and analysis pratt," The University of Liverpool (United Kingdom), 2019.
- [3] A. Lavric *et al.*, "Keratoconus Severity detection from elevation, topography and pachymetry raw data using a machine learning approach," *IEEE Access*, vol. 9, pp. 84344–84355, 2021, doi: 10.1109/ACCESS.2021.3086021.
- [4] S. R. Lin, J. G. Ladas, G. G. Bahadur, S. Al-Hashimi, and R. Pineda, "A Review of machine learning techniques for keratoconus detection and refractive surgery screening," *Seminars in Ophthalmology*, vol. 34, no. 4, pp. 317–326, May 2019, doi: 10.1080/08820538.2019.1620812.
- [5] A. H. Al-Timemy *et al.*, "A hybrid deep learning framework for keratoconus detection based on anterior and posterior corneal maps," *Investigative Ophthalmology and Visual Science*, vol. 62, no. 11, 2021.
- [6] R. Rampat *et al.*, "Artificial intelligence in cornea, refractive surgery, and cataract: basic principles, clinical applications, and future directions," *Asia-Pacific Journal of Ophthalmology*, vol. 10, no. 3, pp. 268–281, May 2021, doi: 10.1097/APO.0000000000000394.
- [7] A. Lavric and P. Valentin, "KeratoDetect: keratoconus detection algorithm using convolutional neural networks," *Computational Intelligence and Neuroscience*, pp. 1–9, Jan. 2019, doi: 10.1155/2019/8162567.
- [8] P. A. Accardo and S. Pensiero, "Neural Network-based system for early keratoconus detection from corneal topography," *Journal of Biomedical Informatics*, vol. 35, no. 3, pp. 151–159, Jun. 2002, doi: 10.1016/S1532-0464(02)00513-0.
- [9] B.-I. Kuo *et al.*, "Keratoconus screening based on deep learning approach of corneal topography," *Translational Vision Science and Technology*, vol. 9, no. 2, Sep. 2020, doi: 10.1167/tvst.9.2.53.




- [10] X. Chen *et al.*, “Keratoconus detection of changes using deep learning of colour-coded maps,” *BMJ Open Ophthalmology*, vol. 6, no. 1, Jul. 2021, doi: 10.1136/bmjophth-2021-000824.
- [11] F. Cavas-Martínez, L. Bataille, D. G. Fernández-Pacheco, F. J. F. Cañavate, and J. L. Alio, “Keratoconus detection based on a new corneal volumetric analysis,” *Scientific Reports*, vol. 7, no. 1, Dec. 2017, doi: 10.1038/s41598-017-16145-3.
- [12] H. A. H. Mahmoud and H. A. Mengash, “Automated Keratoconus detection by 3D corneal images reconstruction,” *Sensors*, vol. 21, no. 7, Mar. 2021, doi: 10.3390/s21072326.
- [13] K. Kamiya *et al.*, “Keratoconus detection using deep learning of colour-coded maps with anterior segment optical coherence tomography: a diagnostic accuracy study,” *BMJ Open*, vol. 9, no. 9, Sep. 2019, doi: 10.1136/bmjopen-2019-031313.
- [14] H. Maile *et al.*, “Machine learning algorithms to detect subclinical keratoconus: systematic review,” *JMIR Medical Informatics*, vol. 9, no. 12, Dec. 2021, doi: 10.2196/27363.
- [15] I. R. Hidalgo *et al.*, “Evaluation of a machine-learning classifier for keratoconus detection based on Scheimpflug tomography,” *Cornea*, vol. 35, no. 6, pp. 827–832, Jun. 2016, doi: 10.1097/ICO.0000000000000834.
- [16] D. Gatinel, J. Malet, T. Hoang-Xuan, and D. T. Azar, “Corneal elevation topography: best fit sphere, elevation distance, asphericity, toricity, and clinical implications,” *Cornea*, vol. 30, no. 5, pp. 508–515, May 2011, doi: 10.1097/ICO.0b013e3181f4fa7.
- [17] A. H. Al-Timemy *et al.*, “A hybrid deep learning construct for detecting keratoconus from corneal maps,” *Translational Vision Science and Technology*, vol. 10, no. 14, Dec. 2021, doi: 10.1167/tvst.10.14.16.
- [18] G. Castro-Luna and A. Pérez-Rueda, “A predictive model for early diagnosis of keratoconus,” *BMC Ophthalmology*, vol. 20, no. 1, Dec. 2020, doi: 10.1186/s12886-020-01531-9.
- [19] M. H. Beale, M. T. Hagan, and H. B. Demuth, “Deep Learning Toolbox™ user’s guide.” The MathWorks, Inc., 2019.
- [20] S. Gairola *et al.*, “SmartKC,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 4, pp. 1–27, Dec. 2021, doi: 10.1145/3494982.
- [21] W. Xu, L. Jin, P.-Z. Zhu, K. He, W.-H. Yang, and M.-N. Wu, “Implementation and application of an intelligent pterygium diagnosis system based on deep learning,” *Frontiers in Psychology*, vol. 12, Oct. 2021, doi: 10.3389/fpsyg.2021.759229.
- [22] K. Kamiya, Y. Ayatsuka, Y. Kato, N. Shoji, Y. Mori, and K. Miyata, “Diagnosability of keratoconus using deep learning with placido disk-based corneal topography,” *Frontiers in Medicine*, vol. 8, Oct. 2021, doi: 10.3389/fmed.2021.724902.
- [23] M. Aatila, M. Lachgar, H. Hrimech, and A. Kartit, “A survey on artificial intelligence in ophthalmology: keratoconus classification,” *International Journal of Computer Engineering and Data Science (IJCEDS)*, vol. 1, no. 1, pp. 22–27, 2021.
- [24] S. K. Jameel, S. Aydin, and N. H. Ghaeb, “Machine learning techniques for corneal diseases diagnosis: a survey,” *International Journal of Image and Graphics*, vol. 21, no. 2, 2021.
- [25] R. Kanimozhia and R. Gayathrib, “Comparative thresholding analysis for keratoconus detection using probability density function,” *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 9, pp. 571–577, 2021.
- [26] H. Sonar, A. Kadam, P. Bhoir, and B. Joshi, “Detection of keratoconus disease,” *ITM Web of Conferences*, vol. 32, Jul. 2020, doi: 10.1051/itmconf/20203203019.
- [27] A. Salman, T. Darwish, A. Ali, M. Ghabra, and R. Shaaban, “Sensitivity and specificity of Sirius indices in diagnosis of keratoconus and suspect keratoconus,” *European Journal of Ophthalmology*, vol. 32, no. 2, pp. 790–797, Mar. 2022, doi: 10.1177/11206721211060139.
- [28] N. Impana, K. J. Bhoomika, S. S. Suraksha, K. Sawhney, D. Priya, and G. Mamatha, “Comparative analysis of deep learning methods for detection of keratoconus,” *Journal of University of Shanghai for Science and Technology*, vol. 23, no. 6, pp. 1546–1553, 2021.
- [29] M. B. Souza, F. W. Medeiros, D. B. Souza, R. Garcia, and M. R. Alves, “Evaluation of machine learning classifiers in keratoconus detection from orbscan II examinations,” *Clinics*, vol. 65, no. 12, pp. 1223–1228, Jan. 2010, doi: 10.1590/S1807-59322010001200002.
- [30] M. C. Arbelaez, F. Versaci, G. Vestri, P. Barboni, and G. Savini, “Use of a support vector machine for keratoconus and subclinical keratoconus detection by topographic and tomographic data,” *Ophthalmology*, vol. 119, no. 11, pp. 2231–2238, Nov. 2012, doi: 10.1016/j.ophtha.2012.06.005.
- [31] H. D. Perry, J. N. Buxton, and B. S. Fine, “Round and oval cones in keratoconus,” *Ophthalmology*, vol. 87, no. 9, pp. 905–909, Sep. 1980, doi: 10.1016/s0161-6420(80)35145-2.
- [32] R. Ishii, K. Kamiya, A. Igarashi, K. Shimizu, Y. Utsumi, and T. Kumanomido, “Correlation of corneal elevation with severity of keratoconus by means of anterior and posterior topographic analysis,” *Cornea*, vol. 31, no. 3, pp. 253–258, Mar. 2012, doi: 10.1097/ICO.0B013E31823D1EE0.
- [33] I. Issarti, A. Consejo, M. Jiménez-García, S. Hershko, C. Koppen, and J. J. Rozema, “Computer aided diagnosis for suspect keratoconus detection,” *Computers in Biology and Medicine*, vol. 109, pp. 33–42, Jun. 2019, doi: 10.1016/j.compbiomed.2019.04.024.
- [34] T. K. Yoo *et al.*, “Adopting machine learning to automatically identify candidate patients for corneal refractive surgery,” *npj Digital Medicine*, vol. 2, no. 1, Dec. 2019, doi: 10.1038/s41746-019-0135-8.

## BIOGRAPHIES OF AUTHORS






**Priya Dhinakaran**     is working as Assistant Professor in ISE Department, RV College of Engineering, Bengaluru. She has around 20 research papers in her credit in international journals and conferences. She is currently pursuing her Ph.D. in the domain of visual computing in healthcare. She is involved in various consultancy projects with IT companies. Her research interests are AI, Deep Learning, IoT and Cyber Security. She is involved as a resource person in various centers of excellence in domains such as IoT and Visual Computing. She can be contacted at email: priyad@rvce.edu.in.



**Mamatha Gowdra Shivanandappa**    is currently working as a Professor & Associate Dean (PG studies) in ISE Department, RV College of Engineering, Bengaluru. She is working in the areas of Cloud computing, AI, IoT, Software Engineering and Networks. She has around 45 publications to her credit in International Journals and conferences. She is currently involved in consultancy and research work with companies and agencies for the submission of proposals. She is also responsible for establishing “Women in Cloud Center of Excellence in India” in association with WiC, USA for training in cloud technologies. She is also a part of research advisory committee for many internal and external research scholars for guidance. She can be contacted at email: [mamathags@rvce.edu.in](mailto:mamathags@rvce.edu.in).



**Srijan Devnath**    is a 3rd year B.E student at RV College of Engineering Bangalore, where he is studying Information Science and Engineering from 2019 and is expected to graduate in 2023. His research interests are concerning the application and impact of computer and analytical algorithms and models across multiple fields including but not limited to healthcare, finance, computer science, microsystems, IoT devices, mechanical and semi-autonomous systems, and mechanics. He has a deep interest in Machine learning and Deep learning with prolific experience in their practical applications and uses. He can be contacted at email: [srijandevnath.is19@rvce.edu.in](mailto:srijandevnath.is19@rvce.edu.in).