EVALUATING THE EFFECTIVENESS OF SQUEEZENET ARCHITECTURE FOR EARLY DETECTION OF EYE DISEASES ON RESOURCE LIMITED DEVICES: A COMPARATIVE ANALYSIS

Anichebe Osita, Hull University

Abstract

One of several major constraints inhibiting the deployment of neural networks on single-board computers and embedded devices is its large memory and power consumption, Gholami et al. (2018). We aim to introduce squeezeNet, a AlexNet-level accuracy architecture with 50x fewer parameters and< 0.5 MB model size, (Iandola et al., 2016). The fewer parameter attribute due to its avoidance of depth wise-separable convolutions makes its implementation more conservative on memory and energy consumption metrics on its deployment on mobile processor platforms.

1. Introduction:

Artificial Intelligence (AI) has been gaining momentum in various fields, including healthcare services Galbusera et al., (2019). The ability of AI to process large amounts of data and identify patterns have made it a valuable tool in disease detection and diagnosis in the medical field. The treatment of diseases is more effective when detected at an early stage. Eye diseases inclusive of cataracts, glaucoma and diabetic retinopathy are among the most common and serious health which is more prevalent within the developing world due to a lower spread of high capacity computational platforms, and availability of low-cost energy required for the functioning of diagnostic tools of eye diseases for an early detection. Convolutional Neural Networks (CNNs) have shown promising results when applied for the extraction and classification of raw image input data. This property is utilized in detecting eye diseases from retinal images. Among CNNs, SqueezeNet is a lightweight architecture with a small number of parameters, making it an attractive option for deployment in resource-limited environments. (Gholami et al.,2018).

Early detection of eye diseases is crucial to prevent vision loss and improve the quality of life of affected individuals. However, manual screening of large volumes of retinal images by trained specialists is time-consuming and costly. Therefore, automated methods for detecting eye diseases have been developed to increase the efficiency and accuracy of screening programs. This study aims to evaluate the effectiveness of SqueezeNet architecture in detecting eye diseases from retinal images against the performance and resource metrics obtained using several major CNN model architectures.

The main research question of this project is to analyse the performance of the SqueezeNet architecture comparing its results to other CNN architectures and implementations for a multi-class eye disease classification from retinal images.

The expected outcome of this project is to determine the effectiveness of the SqueezeNet architecture in detecting eye diseases from retinal images and compare it with other CNN architectures. The results of this study can guide the development of efficient and accurate automated screening tools for eye diseases. And also provide guidance for future model deployments on resource restricted devices such as drones and wearables technologies.

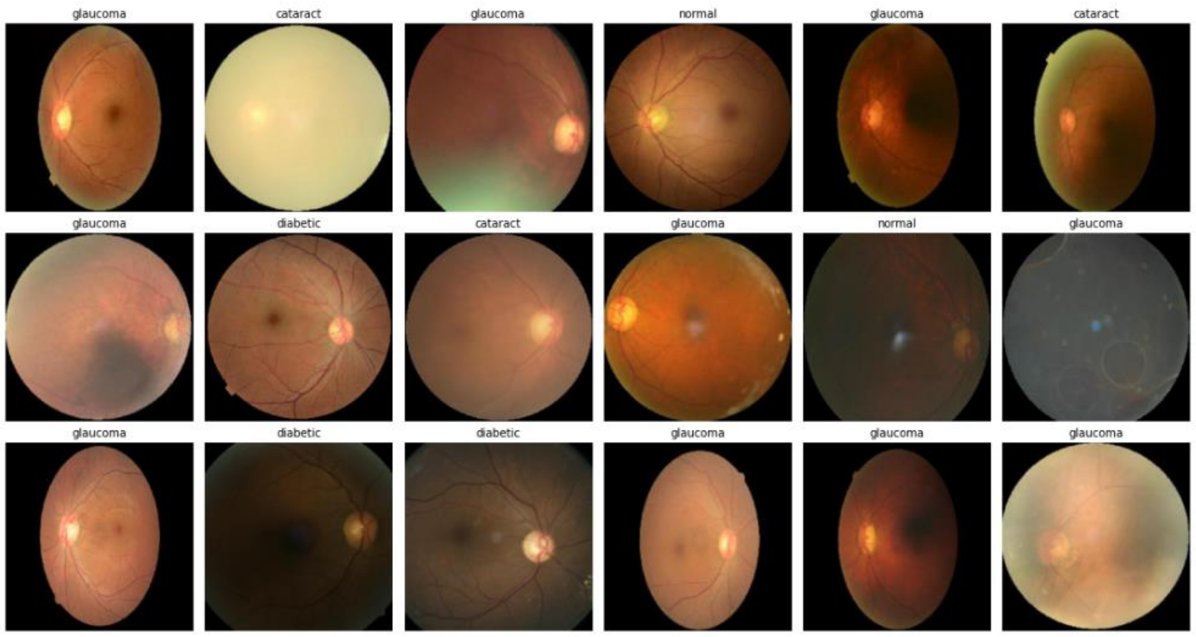


Fig. 1.

Some examples of retinal images (cataracts, glaucoma, normal and diabetic retinopathy) from the dataset

Background:

Automated methods for detecting eye diseases from retinal images have been a topic of interest for researchers in recent years. Several studies have explored the use of deep learning techniques, specifically convolutional neural networks (CNNs), for disease detection in retinal images. These techniques have shown promising results in terms of accuracy and efficiency compared to traditional methods.

CNNs have been applied to various medical imaging tasks, including Diabetic retinopathy detection through deep learning techniques: A review, A study conducted by Alyoubi et al, (2020) utilized a CNN to detect diabetic retinopathy in retinal images utilizing inception-v3 CNN implementation . another study by Agrawal, P., Girshick, R., & Malik, J. (2014) titled: An improved SqueezeNet model for the diagnosis of lung cancer in CT scans. Aimed at Utilizing squeezeNet model for lung cancer diagnosis and benchmarking performance result against other well-known and established models inclusive of LeNet-50, DenseNet-121, ResNet-50 and VGG-11.

Similarly, Glaret subin and Muthukannan (2022) proposed an optimized convolutional neural network for the detection of multiple eye diseases. These studies demonstrate the potential of CNNs in the automated detection of eye diseases from retinal images.

SqueezeNet is a CNN architecture designed to have a small number of parameters, making it a lightweight and efficient option for deployment in resource-limited environments. (Iandola et al., 2016). The architecture achieves this by replacing the 3x3 filters in the convolutional layers with 1x1 filters, reducing the number of parameters without compromising performance. SqueezeNet has been shown to be effective in various image classification tasks, including the ImageNet dataset, where it achieved comparable accuracy to deeper CNN architectures with significantly fewer parameters.

Several studies have explored the use of SqueezeNet for medical image classification tasks. For example, Akpinar et al, (2020) used SqueezeNet for the detection of chest Abnormality using x-ray images based on SqueezeNet Similarly, Zhang et al. (2020) used SqueezeNet for the classification of lung nodules in CT images and achieved an accuracy of 90.2%. These studies demonstrate the potential of SqueezeNet in medical image classification tasks, which can be extended for eye disease detection from retinal images.

However, there has been limited research on the effectiveness of SqueezeNet for the detection of eye diseases from retinal images considering its computational resource requirements. This study aims to fill this gap by evaluating the performance of SqueezeNet in comparison to other major CNN architectures for the detection of eye diseases from retinal images and provide a comparative analysis of its performance and resource requirements. The evaluation will focus on the computational requirements and model size of the architectures while maintaining a good degree of accuracy in their predictions.

In summary, this project aims to evaluate the effectiveness of the SqueezeNet architecture for the early detection of eye diseases from retinal images. The related work indicates the potential of CNNs and SqueezeNet for medical image classification tasks, including disease detection. However, there has been limited research on the effectiveness of SqueezeNet for eye disease detection from retinal images. This study aims to fill this gap and provide insights into the potential of SqueezeNet as a lightweight and efficient option for disease detection in resource-limited environments.

Report Objective:

The objective of this study is to evaluate the effectiveness of the SqueezeNet architecture compared to other major CNN architectures on a retina images dataset for a multi-classification within resource-limited devices.

The report describes the pre-processing of the images, model architecture and fine-tuning, training and evaluation of the model. The report also includes the benchmark model used, hyperparameters used for training, and key metrics for evaluating the model's accuracy and efficiency. Additionally, the report presents a comparative analysis of the performance of the parallel models trained using different architectures and evaluated with accuracy. The objective of the report is to provide insights into the performance of the models and their potential for eye disease classification.

This report aims to provide a comparative analysis of the above performance on different CNN architectures. The outcome of this study will guide the development of efficient and accurate automated screening tools for eye diseases that can be deployed in low-resource settings and portable host device.

Methodology:

In this project, we will use five deep learning architectures, SqueezeNet, VGG-11, MobileNetV3, ResNet and a custom CNN architecture implementation, for image classification. Both architectures will be implemented using the TensorFlow deep learning library.

We will use the retina image dataset, which contains over ten thousand images belonging to four categories inclusive glaucoma, cataract, diabetic and normal vision as our primary dataset for training and testing all architectures within this study.

To implement SqueezeNet and VGG-M, we will use transfer learning, which involves fine-tuning a pre-trained model on a new dataset. In this case, we will use the pre-trained models provided by the TensorFlow library, which have been trained on the ImageNet dataset. This will enable us to leverage the pre-trained models' feature extraction capabilities and significantly reduce the training time required for our models.

For our experimental setup, we will randomly split the ImageNet dataset into training and testing sets using a 70:30 split. We will train both models on the training set using the same hyperparameters, such as learning rate, batch size, and optimizer, and evaluate their performance on the testing set using standard evaluation metrics such as accuracy, precision, recall, and F1 score.

We will also compare both performance and accuracy metrics on each model architectures to assess their computational efficiency. Finally, we will conduct ablation experiments to identify the reasons for any performance differences observed between the two architectures. Specifically, we will investigate the impact of removing specific layers or modifying the architecture's hyperparameters on the model's accuracy and training time.

We will report our results in tables and graphs and provide a detailed analysis of our findings. We will also provide a description of our experimental setup and methodology, along with code snippets and references to relevant literature, to enable other researchers to reproduce our results.

EXPERIMENTS:

For our experiment, we utilised a eye disease retinal images dataset obtained from Kaggle at https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification.

The dataset used for our model analysis contains 10,000 images post data augmentation. Image classes of type Cataract, Diabetic retinopathy, glaucoma and normal was balanced at a ratio of 23%, 27%, 26% and 24% respectively. The benchmark model used is the SqueezeNet CNN implementation.

The unlabelled images were first labelled in the four different eye disease folders (cataract, diabetic retinopathy, glaucoma, normal) to a new format that includes the name of the disease and a unique number.

We resized the images to new dimension of (512*512) to standardize the image data for our model training afterwards, we updated the names of the images in the image folder according to the name of the eye disease, using a dictionary to map the original class names to the new class names. Finally, the images were converted to into NumPy arrays, and class labels was extracted from the file names to use as labels for a machine learning model. The labels are transformed using a LabelEncoder to convert them into numerical values.

The model begins with a convolutional layer with 96 (7x7) filters, a stride of 2, and padding. This is followed by a max pooling layer. Four "fire" blocks consisting of a squeeze layer and two expand layers then follow. These blocks add depth to the model without increasing parameters.

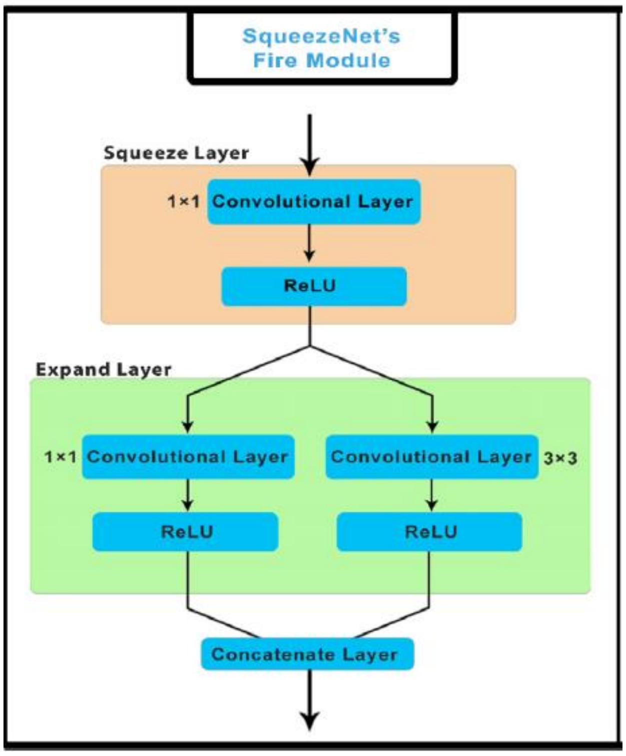


Fig 2: SqueezeNet Fire module: Agrawal, P., Girshick, R., & Malik, J. (2014)

Next, there are two convolutional layers with 64 (1x1) filters, and a dropout layer with a rate of 0.5 to prevent overfitting. The final convolutional layer has 4 filters of size (1,1), followed by a ReLU activation function. The output is flattened, and a dense layer with 4 units and a softmax activation function produces the final output.

The output from the convolutional layers is flattened, and a dense layer with 4 units and a softmax activation function is added to produce the final output of the model.

The model is compiled with the sparse categorical cross-entropy loss function, the Adam optimizer, and the accuracy metric.

- The hyperparameters used for training our base SqueezeNet model implementation are as follows:
- Number of epochs: 15
- Dropout rate: 0.5
- Batch size: 32
- Loss function: Sparse categorical cross-entropy
- Optimizer: Adam
- Metrics: Accuracy
- Input shape: The shape of the input images is (224,224,3)

Fire Filter: A list of four integer type entries: 16, 64, 128 and 256 representing the number of filters in each of the fire module.

The data is split into training, validation, and testing sets using the train_test_split function from the sklearn module. The testing data set contains 15% of the data, the validation set contains 20% of the data, we assigned 65% of the data for model training.

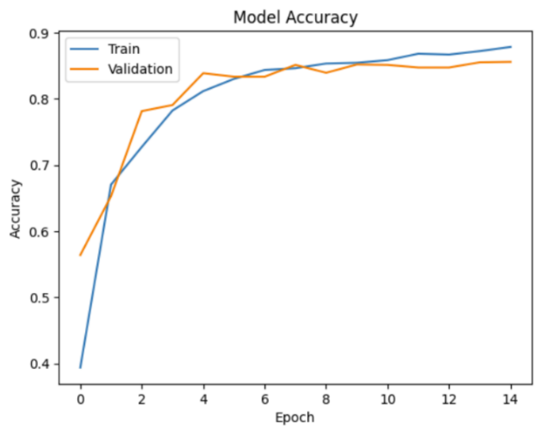
The CNN architecture used is a variant of the SqueezeNet architecture, which consists of a series of convolutional layers followed by Fire modules, which are composed of a squeeze layer and two expand layers. The output of the last Fire module is flattened and passed through a fully connected layer with a SoftMax activation function, which outputs the predicted class probabilities.

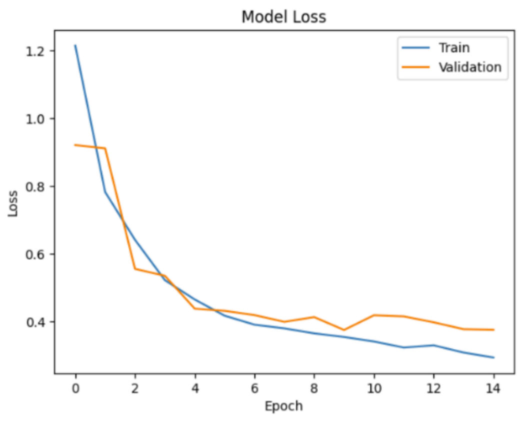
We ran a fit on 15 epochs to avoid overfitting the model, At the end of each epoch, we evaluated the model's accuracy and loss on the validation data. The initial accuracy of the model was only 8.102, which was quite low.

The team then adjusted the network through fine-tuning to improve the model's performance. We did the following:

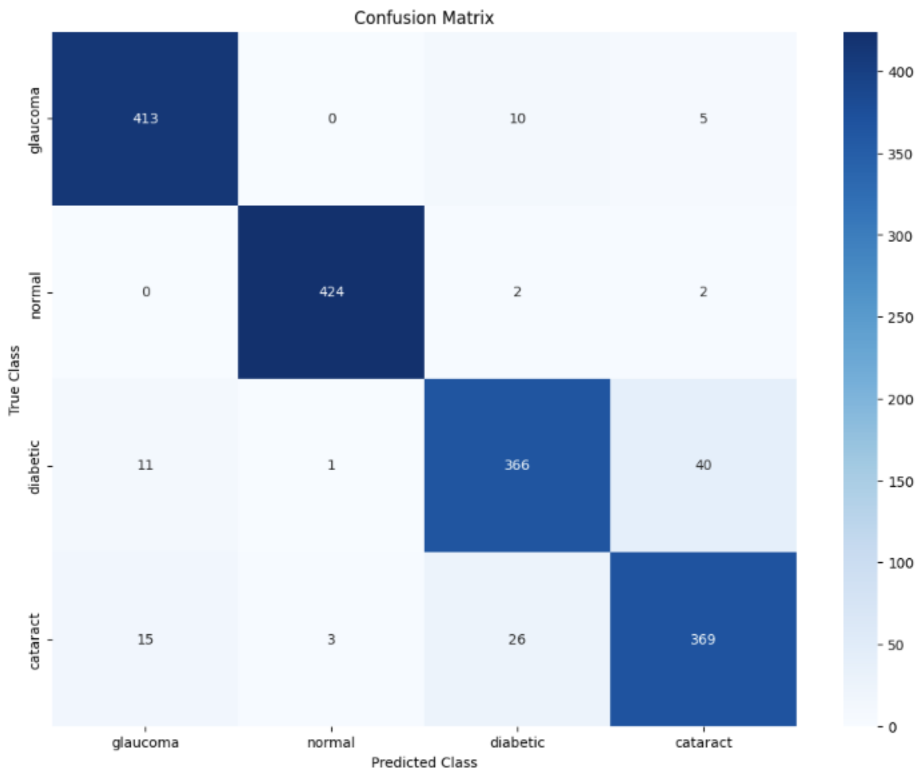
- Removed two last pooling layers at the end of the convolution layer to reduce the models complexity, speed up the training process and prevent an over down-scale of the input future to be learnt.
- We also Augmented the data to increase the training images. Here we took advantage of the perfect symmetry of the retina images. They were rotated, inverted and flipped to present more variations to the model and, we made sure to maintain a balanced set across the four classes of our dataset images used for model training.

After fine-tuning, we observed that the accuracy of the model increased to 0.8566 and the loss decreased to 0.12, which indicated a significant improvement in performance.





To visualize the performance of the model, the team plotted a train/validation plot curve on model loss and accuracy. This curve showed the changes in loss and accuracy during the training process, providing valuable insights into the model's performance.



Finally, the team proceeded to make a scatter plot to visualize the data and gain further insights into the model's behaviour.

During evaluation, we obtained various key metrics on both the models accuracy and efficiency. These include the models:

- Accuracy

- Loss
- RMS error
- F1-score
- Recall score.
- Precision Score
- Model total memory
- Inference time
- And Parameter count.

Using data obtained here as a benchmark for our comparative analysis. Using same retina image dataset, epoch count and image augmentation, we further trained a parallel model utilizing VGG-11 (Visual Geometry Group), MobileNetV3, Residual Network (ResNet), A pretrained MobileNetV3 model and a custom CNN architecture which has three convolutional layers, two max-pooling layers, a dropout layer, and two fully connected layers. It is trained with the Adam optimizer and evaluated with accuracy. It includes regularization to prevent overfitting.

Output metrics is gotten after training the model and is benchmarked to the SqueezeNet for comparative analysis. We plotted a comparative histogram to quickly infer upon the performance, accuracy and features and help visualize the distribution of model performance across the dataset

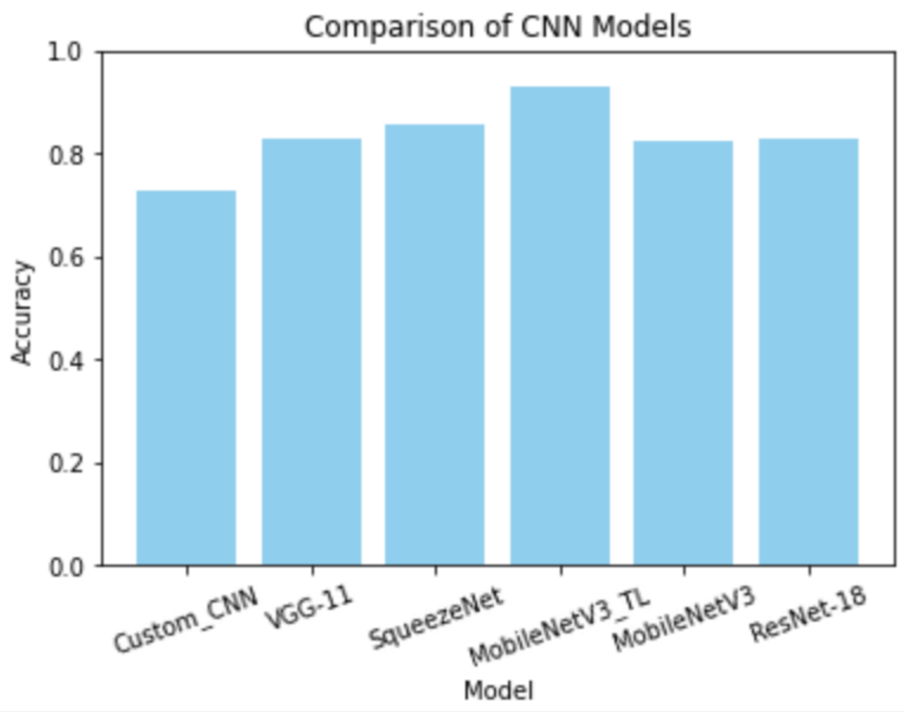


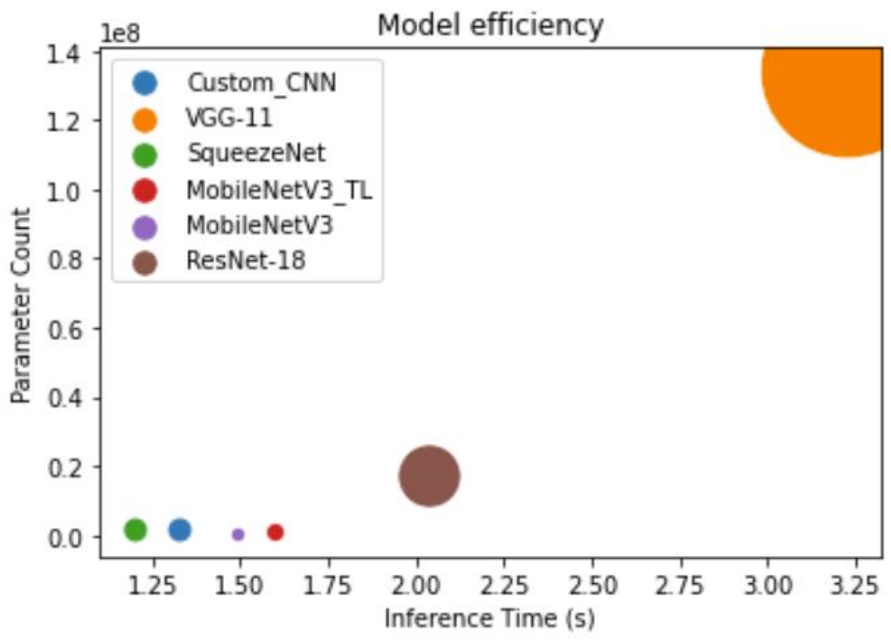
Fig 3: A histogram comparing the various study model accuracy score after model training.

RESULTS:

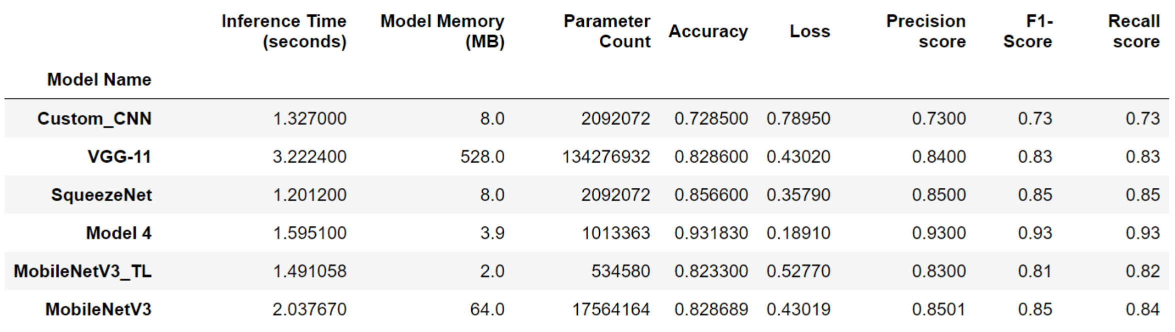
The result of the experiment presents the SqueezeNet CNN architecture as a viable model owning to its top performance on the efficiency metrics and evaluation measures.

The SqueezeNet model stands out with the lowest inference time of just 1.2 seconds, which is significantly faster than VGG-11 and MobileNetV3. This makes it a great option for real-time applications that require fast predictions.

In addition to its fast inference time, the SqueezeNet model also has a relatively small model memory of only 8 MB, which is tied for the smallest memory footprint among the six models. This makes it a good option for applications with limited memory resources.



The figure above presents a graphical representation of the result obtained for the models efficiency using the plot of models time inference against the models parameter count. The radius of the model on the plot indicates its relative model size. It is visible that the MobileNetV3 modelled over transfer learning and its implementation utilizing local retina image data set obtains over 30% efficiency in its model size over the SqueezeNet implementation, but the SqueezeNet is 0.3 more times faster in modeling coming and with lesser parameter count.



Model Name	Inference Time (seconds)	Model Memory (MB)	Parameter Count	Accuracy	Loss	Precision score	F1-Score	Recall score
Custom_CNN	1.327000	8.0	2092072	0.728500	0.78950	0.7300	0.73	0.73
VGG-11	3.222400	528.0	134276932	0.828600	0.43020	0.8400	0.83	0.83
SqueezeNet	1.201200	8.0	2092072	0.856600	0.35790	0.8500	0.85	0.85
Model 4	1.595100	3.9	1013363	0.931830	0.18910	0.9300	0.93	0.93
MobileNetV3_TL	1.491058	2.0	534580	0.823300	0.52770	0.8300	0.81	0.82
MobileNetV3	2.037670	64.0	17564164	0.828689	0.43019	0.8501	0.85	0.84

In addition, the SqueezeNet model was able to achieve these results while using the median parameters of the other AI model. The SqueezeNet model has only 2.09 million parameters while the VGG M model has 134.5 million parameters with both having comparatively even results on models accuracy. This means that the SqueezeNet model is more computationally efficient and requires less memory to train and deploy.

CONCLUSION:

Based on the results of the study, it can be concluded that SqueezeNet architecture is an effective option for the early detection of eye diseases on resource-limited devices. Its lightweight design and low number of parameters make it ideal for deployment in environments with limited computational power and energy resources. Its particularly unmatched inference time makes it more suitable as the modelling architecture choice for fast computation on resource sparce devices. Based on the research, MobileNet CNN architecture implementation may be able to attain superiority at its smaller model size but ultimately, SqueezeNet's efficiency, inference time and accuracy in detecting eye diseases make it a valuable tool in the medical field, particularly in areas where high capacity computational platforms and energy resources are not readily available.

REFERENCES:

Gholami, A., Kwon, K., Wu, B., Tai, Z., Yue, X., Jin, P., Zhao, S. and Keutzer, K., 2018. Squeezenext: Hardware-aware neural network design. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1638-1647).

Alyoubi, W.L., Shalash, W.M. and Abulkhair, M.F., 2020. Diabetic retinopathy detection through deep learning techniques: A review. *Informatics in Medicine Unlocked*, *20*, p.100377.

Glaret subin, P. and Muthukannan, P., 2022. Optimized convolution neural network based multiple eye disease detection.

Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J. and Keutzer, K., 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.

Akpinar, K.N., Genc, S. and Karagol, S., 2020, June. Chest X-ray abnormality detection based on SqueezeNet. In *2020 international conference on electrical, communication, and computer engineering (ICECCE)* (pp. 1-5). IEEE.

Agrawal, P., Girshick, R. and Malik, J., 2014. Analyzing the performance of multilayer neural networks for object recognition. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13* (pp. 329-344). Springer International Publishing.

Madani, M., Behzadi, M.M. and Nabavi, S., 2022. The role of deep learning in advancing breast cancer detection using different imaging modalities: A systematic review. *Cancers*, *14*(21), p.5334.

Asadi, P., Mehrabi, H. and Asadi, A., 2022. Deep Convolutional Neural Networks for Crack Detection Using a Cost-Efficient Single Board Computer-Based Unmanned Aerial Vehicle (UAV) Platform for Bridge Inspection. In *International Conference on Transportation and Development 2022* (pp. 190-201).

Galbusera, F., Casaroli, G. and Bassani, T., 2019. Artificial intelligence and machine learning in spine research. *JOR spine*, *2*(1), p.e1044.