

# **Identifying Retinal Optical Coherence Tomography (OCT) by Image-based Deep Learning**

## **SPE ML Bootcamp Team 3**

**Erica Esatyana (Developer), Christian Woods (Data Engineer), Sabyasachi Prakash (Data Scientist), & Mark Yang (Business Analyst)**

### **1. Problem Introduction**

Swanson and Fujimoto (2017) stated that thirty million OCT scans are performed each year and required a significant amount of time to be accurately analyzed and interpreted. Our company proposed to develop an effective learning algorithm to diagnose various retinal diseases based on medical images using deep learning to gain insight from complex input features provided by the images. By establishing a deep learning model for pathology, it is possible to automatically identify the problem of new patients.

### **2. Dataset**

The dataset was obtained from the open-source platform called 'Kaggle'. It is divided into three folders (train, test, val) and contains subfolders for each image category of NORMAL, neovascularization (CNV), diabetic macular edema (DME), and DRUSEN. The X-Ray images are in .jpeg format and total 84,495 images.

### **3. Features and Preprocessing**

Before we start, make sure to install GPU support in handling huge datasets on your workstation. In this study, the main library used is TensorFlow which is an open source library for machine learning applications such as deep learning. Data augmentation is performed to enable generalization of data points in the training set and attain better testing/validation set. It replaces the original dataset, with a new and randomly transformed (translations, rotations, changes in scale, shearing or horizontal/vertical flips) batch using ImageDataGenerator class. The process does not change the class label and is done at training time.

### **4. Models and Techniques**

We tested numerous image dimensions, epochs, batch\_size, amount of convolution blocks, and with/without dense layers to generate the highest accuracy. Other parameters such as optimizer, loss, convolution block, and activation were defined based on several considerations:

Optimizer: adaptive moment estimation 'adam' - the most popular gradient descent optimization algorithm, calculates the individual adaptive learning rate for each parameter.

Loss: 'categorical\_crossentropy' - a well suited for multi-class classification tasks and a very good measure to distinguish two discrete probability distributions (0 to 1). It only belongs to

one out of many possible labels/categories. This is common to be used on the MNIST (images of the digits).

Convolution block: 'Conv2D (normal/traditional convolution)' and 'SeparableConv2D (depthwise and pointwise convolution)' - Separable convolution is able to process in a shorter amount of time (less computations) compared to the normal convolution. However, it reduces the number of parameters in a convolution and fails to properly learn during the training process.

Activation: Rectified Linear Unit 'ReLU' - the most used activation function and ranges from zero to the max input value. All the negative values resulting in the model become zero and the neurons will be deactivated. ReLu will not activate all the neurons at the same time.

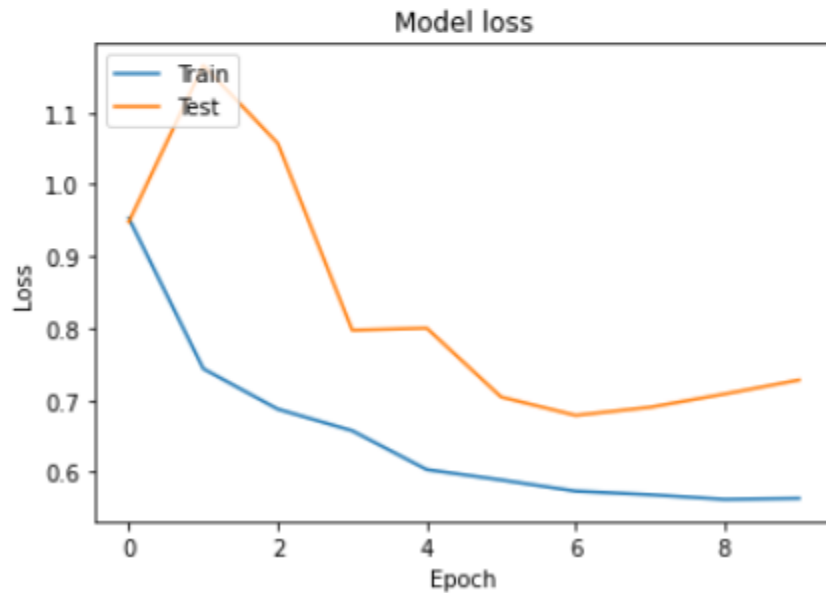
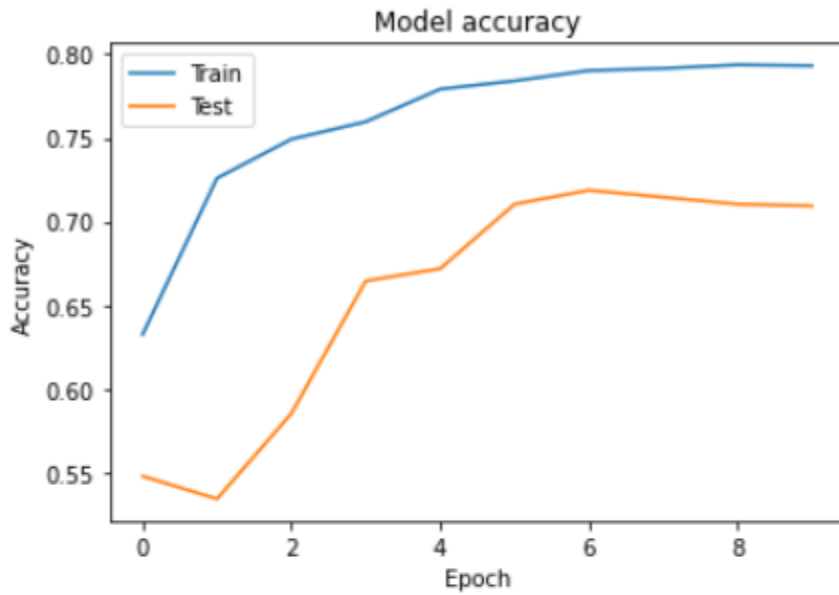
Batch Normalization is crucial to accelerate the training process by halving the epochs and providing some regularization to reduce error.

Flatten Layer is usually located between the convolutional layer and the fully connected layer. It transforms a two-dimensional matrix of input features into a vector.

Furthermore, we fit the model with all the parameters provided and resulted with different loss and accuracy values in each epoch.

## **5. Results and Discussion**

Multiple combinations of different hyperparameters and network architecture were tested in order to obtain the best possible results. The large amount of computational power necessary to execute these models was a challenge we first had to contend with. Implementing progressive resizing to reduce image sizes enables quicker training of the models for parameter optimization. The result can be seen clearly on 'plot learning curves' along with the details of 'train accuracy': 0.759, 'train loss': 0.650, 'test accuracy': 0.656, 'test loss': 0.827. A balanced result between train and test accuracy presents that the model is 'robust'.



## 6. Conclusion

In this project we were successfully able to implement a basic convolution neural network model to classify the ocular conditions normal, CNV, DME, and Drusen. We were able to achieve a test accuracy of 0.72 at an epoch size of 6 using a single convolutional block consisting of two layers along with multiple dense layers. The future direction of this project would focus on further parameter optimization along with completing the implementation of progressive resizing. Both have been shown to improve model accuracy. We believe that with sufficient time this model can be further developed and optimized to achieve satisfactory results.

