DIABETIC RETINOPATHY CLASSIFICATION USING DEEP NEURAL NETWORKS

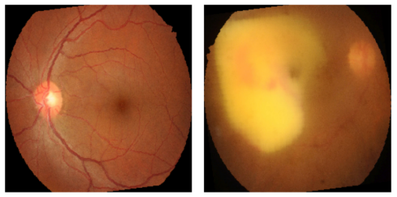
Saiteja Malyala and Sampath Garuda

Institute of Signal Processing and System Theory, University of Stuttgart, Germany

1. Introduction

Diabetic Retinopathy (DR) is an eye disease which damages the retina, and is associated with long-standing diabetes.DR if not detected early, leads to loss of vision.

The aim of this project is to classify the Diabetic Retinopathy into 2 different stages namely non-referable(**NRDR**) and referable(**RDR**) using deep neural networks.



2. Dataflow and methods

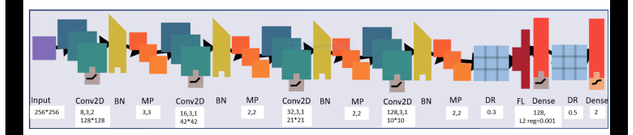
2.1 Dataset

For the mentioned classification task **Indian Diabetic Retinopathy Image Dataset** (IDRID) is used, which contains 516 Color fundus images along with corresponding severity grade.

2.2 Data pipeline

- **Data Preprocessing**
  - Reassign labels for binary classification task.(grades 0,1 to NRDR),(grades 2,3,4 to RDR)
  - Shuffle and partition training dataset to training and validation at 80-20 ratio
  - Resample the training dataset, to avoid class imbalance
  - Image cropped to maximize fundus area, normalized and then resized to 256*256
- **Data Loading methods**
  Two pipelines were implemented to load data.
  - Using from_ tensor_slices method of tf.Data
  - Using ImageDataGenerator, followed by its flow_from_data_frame method
- **Data Augmentation**
  - Methods: Image Rotation, Zoom, Flipping
  - Purpose: To Improve Generalization

2.3 Model Architecture

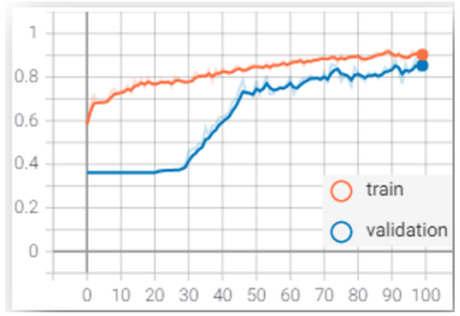


- Architecture includes four blocks of Conv2D, Batch Normalization,Max Pooling followed by dense layers.
- Activation Function used(except for last layer)for all the layers is ReLU, for the last Dense, Softmax is used.
- To avoid Over-fitting L2 regularization and Dropouts were employed.

2.4 Training

- Hyper parameters, learning rate, number of neurons in the last dense layer, epochs were tuned using hparams of tensorflow and visualized on tensorboard.

- The best **Hyperparameters Set**:- Neurons in dense layer:**128**,epochs:**100**, lr:**1e-4**
- The DCNN architecture was trained with ADAM optimizer on best hyperparameter set
- **Statistics:** Params: **453k**,Time: 1hr

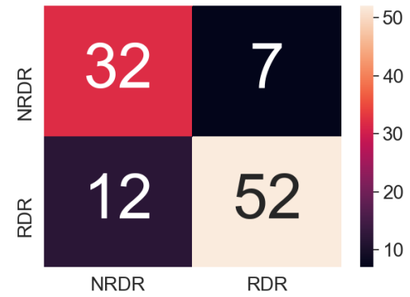


*Train and Validation Accuracy

- Performed **Transfer Learning** by using Resnet50 with Imagenet weights, Fine-Tuned on end Conv layers Training: 10 epochs, lr: 1e-4,opt: Adam.

3. Evaluation Metrics

3.1 Confusion Matrix



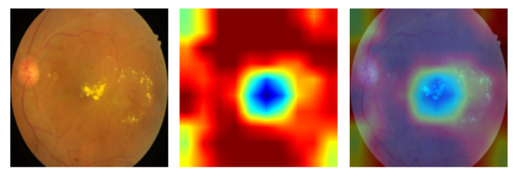
3.2 Test Accuracy

Without Transfer Learning **77.7%** , With transfer learning **81.78%** accuracy was obtained on the Test DataSet

4. Deep Vizualization

4.1 Grad-CAM

Using the gradients flowing into the last convolution layer to produce a map which highlights the important regions of the image in order to make a prediction.



5. Deployment on Web

The model has been deployed on web for the end user utilization. A very simple Web app was developed using Django Video Link for web app Demo