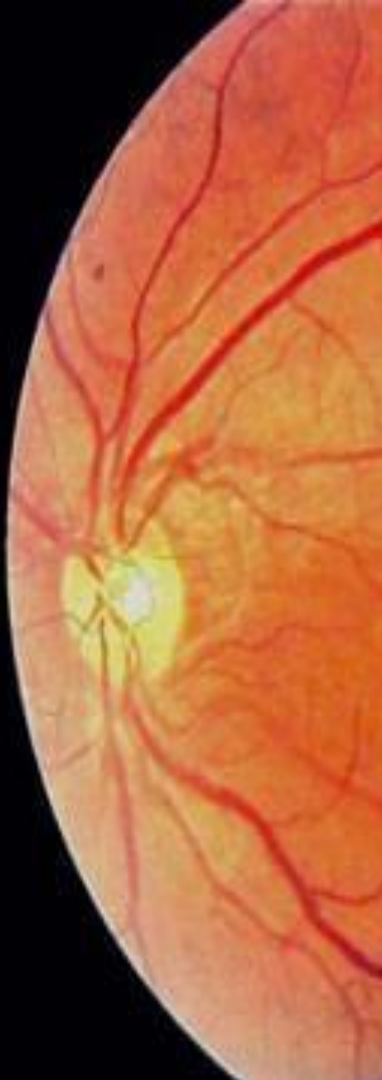
Deep Learning Based Technique For Multiclass Classification Of Diabetic Retinopathy Disease Grading

Presented by- Prerana Bora

Table of Contents



- Analysis of design requirements relevant to the problem identified
- Methodology to solve the Problem
- Design & Implementation of the Project Work
- Performance analysis of the proposed work
- Scope & Limitations
- Future Directions
- Conclusion
- Reference

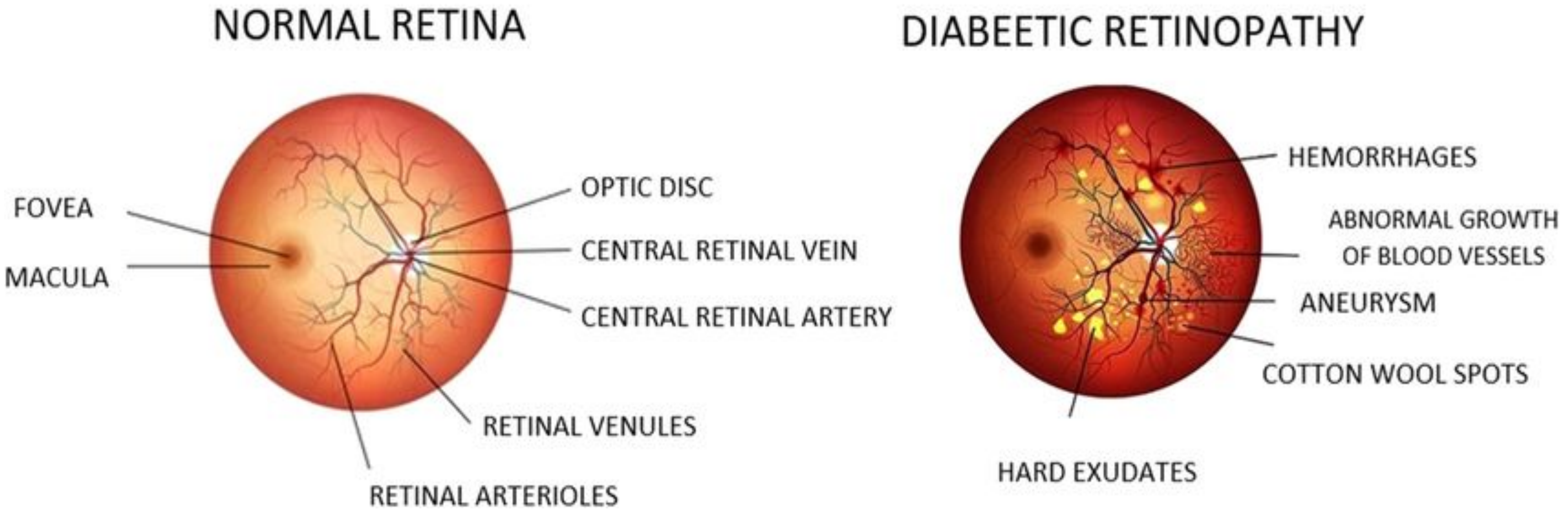
Introduction

Diabetic retinopathy is a condition that occurs as a result of damage to the blood vessels of the retina in people who have diabetes.

These blood vessels expand and cause venous bleeding. Or they can close, stopping blood from passing through.

- Total of 2.6% blindness in adults worldwide are caused by diabetic retinopathy. 1/3rd of the diabetic patients have DR but shows no symptoms, leading to left untreated.
- It is estimated currently in India about **3 million** people aged 40 years or older live with VTDR and are at risk of vision loss.

DIABETIC RETINOPATHY



Problem Definition:

To help ophthalmologist, we can use computer technology to analyze eye images and classify the severity of the disease into five grades, from no disease to severe.

- Detailed pictures of the eye fundus images, are taken using fundus cameras or retinal cameras.
- Fundus images are then enhanced to improve their quality using computer vision techniques.
- Then we use advanced computer programs called deep learning models. These programs learn to recognize patterns in the eye images, by mimicking the human brain's way of recognizing patterns.
- The model learns hidden patterns from a large set of pre-labeled fundus images (Training). Those patterns include blood spots, swelling, and abnormal blood vessel growth.

- The model is designed to classify the severity of diabetic retinopathy into five grades:
    1. No DR (Grade 0): No signs of diabetic retinopathy.
    2. Mild DR (Grade 1): Minor abnormalities, no significant damage.
    3. Moderate DR (Grade 2): Noticeable damage, not severe.
    4. Severe DR (Grade 3): Significant damage, risk of vision loss.
    5. Proliferative DR (Grade 4): Advanced stage, serious risk of blindness.
- Once trained, the model can analyze new fundus images and predict the grade of diabetic retinopathy.

Fundus images are enhanced by applying Computer vision technology

Advanced computer program learns the hidden pattern from the fundus images

Classification

1. No DR (Grade 0)
2. Mild DR (Grade 1)
3. Moderate DR (Grade 2)
4. Severe DR (Grade 3)
5. Proliferative DR (Grade 4)

Motivation

- To help ophthalmologist in early and fast diagnosis of Diabetic retinopathy.
- The ratio of 928:1 between the 18.55 million diabetic retinopathy patients and the 20,000 ophthalmologists in India underscores the urgent necessity for technological interventions to detect diabetes Retinopathy within the mass population.(source:International Diabetes Federation (IDF) Diabetes Atlas (10th edition, 2021))

- Manual classification of the 5 grades of diabetic retinopathy is challenging due to subtle differences in retinal features. Manual Grading can be subjective and prone to vary among different ophthalmologists, leading to inconsistent diagnoses.
- Deep learning models, can automate the classification process, ensuring consistent and objective analysis.

Objectives Of the project is :

- Develop a deep learning model capable of accurately classifying the five grades of diabetic retinopathy.
- To reduce the time in diagnosis so that doctors can check large numbers of people's eyes quickly, and only focus on those who need help the most.
- Use this program in remote places where there aren’t many eye doctors , so everyone can get checked for diabetic eye problems.
- Make this DR detection affordable for everyone .
- Ensure that the program provides a consistent result , reducing errors and uncertainty in diagnosis.
- Help doctors in decision making with better interpretation , leading to improved treatment.

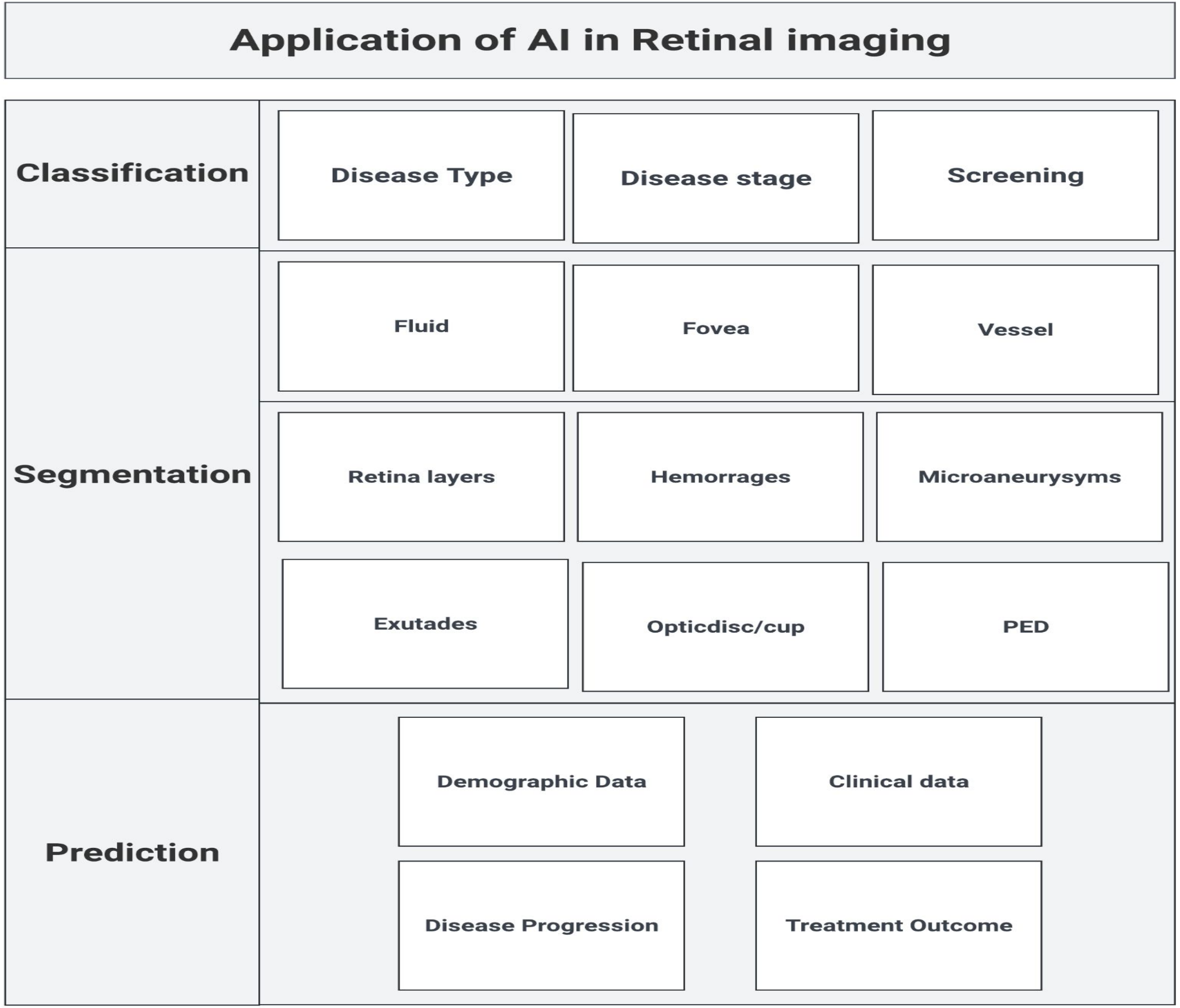
The objectives are to address the following challenges in DR detection:

- Tackling the lack of widespread screening programs.
- Addressing the shortage of skilled ophthalmologists.
- Managing variability in image quality and interpretation.
- Developing scalable and accurate automated screening solutions.

Relevance to the AI and ML

- Recent Trend in diagnosis of diabetic retinopathy : Using Automated screening models using deep learning system.
- Various techniques of AI and ML is used in DR grading :

1. Deep Learning Models:
   - Convolutional Neural Networks (CNNs) are used for image classification tasks in DR grading.
   - Techniques like InceptionV3, ResNet, and U-Net used to extract features from retinal images and classify them.
2. Support Vector Machines (SVM):
   - SVM is used as a supervised learning algorithm for classification tasks in diabetic retinopathy.
3. Clustering Algorithms:
   - K-means clustering used for unsupervised grouping of retinal images based on similarities in features.
4. Ensemble Methods:
   - Random Forests or Gradient Boosting Machines models can combine multiple classifiers to improve classification performance.
5. Transfer Learning:
   - It enable the use of pre-trained deep learning models, fine-tuned on large datasets, for diabetic retinopathy classification tasks.



Application of AI in Retinal imaging
Classification
Disease Type
Disease stage
Screening
Segmentation
Fluid
Fovea
Vessel
Retina layers
Hemorrages
Microaneurysyms
Exutades
Opticdisc/cup
PED
Prediction
Demographic Data
Clinical data
Disease Progression
Treatment Outcome

Existing work and Performance

(Literature Survey)

Diabetic Retinopathy Detection Using Prognosis of Microaneurysm and Early Diagnosis System for Non-Proliferative Diabetic Retinopathy Based on Deep Learning Algorithms(2020)

Lifeng Qiao; Ying Zhu; Hui Zhou

Methodology used:

1. PMNDPR(Prognosis of Microaneurysm and Early Diagnosis System for Non-Proliferative Diabetic Retinopathy) system is used. This system analyzes retinal images and identify early signs of diabetic retinopathy, such as microaneurysms and do early detection.
2. Here they utilized DCNN architecture for semantic segmentation of fundus images.
3. The system takes input images and features extracted by convolutional layers. The output of the convolutional layers is then flattened and analyzed by a fully connected layer with Relu activation, producing a 336-dimensional feature vector. Finally, a softmax function translates the output of the fully connected layer into the likelihood of each class.
4. For Preprocessing Curvelet Transform is used to enhance the dark lesions.
5. Optimal bandpass filtering is used to optimise the contrast of bright lesions.
6. Candidate lesions are then detected through a series of steps, including Gaussian filtering, matched filters, Laplacians, and maximizing mutual information.
7. Finally, candidate extraction involves measuring correlation coefficients of each pixel using non-linear filters with Gaussian kernels, then principal component analysis to reduce dimensionality and identify relevant features.

Quality of this work:

- On average sensitivity values of 97.4%, 98.4% and 95.1%, respectively for detecting dark lesions
- and 96.8%, 97.1% and 95.3% for detecting bright lesions.

Pros:

1. High sensitivity.
2. Efficient Preprocessing technique is used : PCA, curvelet transformation, bandpass filtering.
3. Flexible candidate lesion detection.
4. Classification accuracy increased than Gaussian mixture model and SVM.

Drawbacks:

1. Dependent on the trained image quality.(Aptos dataset is used here)
2. Shows false positive and false negative testing.
3. Complex model approach.
4. High time duration for training.

A Deep Learning Ensemble Approach for Diabetic Retinopathy Detection (2019)

SEHRISH QUMMAR1 , FIAZ GUL KHAN 1 , SAJID SHAH 1 , AHMAD KHAN1 , SHAHABODDIN SHAMSHIRBAND 2,3, ZIA UR REHMAN1 , IFTIKHAR AHMED KHAN 1 , AND WAQAS JADOON 1

Methodology Used:

1. Here used an end to end deep ensemble network for detecting all stages of DR.
2. Ensemble model: Resnet50, Xception, VGGnet
3. Addressed the imbalance data challenged by combine Preprocessing technique: Upsampling and Downsampling
4. The categorical cross entropy Loss function is optimized by using Nesterov accelerated adaptive moment estimation (Nadam) optimizer.
5. This study prefers SGD over Adam optimizer in DR grading.

Quality of this work:

- Accuracy: on average 80.8%
- Recall (Sensitivity) : For class 0 (Normal) highest recall due to a large number of negative examples. For class 1 (mild) lowest recall due to subtle features.
- Precision: 0.63
- F1 score:0. 53

Pros:

1. The ensemble approach enhances classification accuracy.
2. By combining multiple models through stacking, the ensemble method offers increased robustness and generalization capabilities.

Drawbacks:

1. The model may still exhibit shortcomings in accurately detecting all stages of DR.
2. The imbalance in the distribution of samples may contribute to biases in model performance, affecting the overall recall and precision rates.
3. The F1-score, representing the harmonic mean of precision and recall, is relatively low at 53.74%, suggesting a trade-off between precision and recall that may impact the model's overall effectiveness in diabetic retinopathy severity classification.

Automatic Detection and Monitoring of Diabetic Retinopathy Using Efficient Convolutional Neural Networks and Contrast Limited Adaptive Histogram Equalization (2020)

ASRA MOMENI POUR1 , HADI SEYEDARABI 1,2, SEYED HASSAN ABBASI JAHROMI3 , AND ALIREZA JAVADZADEH4

Methodology used:

1. Utilized the Contrast Limited Adaptive Histogram Equalization (CLAHE) method to enhance image quality and normalize intensities as a pre-processing step
2. Employed the EfficientNet B5 architecture for classification, known for its efficiency in scaling network dimensions uniformly.
3. The hyperparameters of the EfficientNet architecture are tuned :depth, width, and resolution scaling are adjusted to find the optimal.
4. The selected EfficientNet model is trained on the preprocessed dataset .
5. Evaluate the model's performance by training on a mixture of Messidor-2 and Messidor datasets and testing on the IDRID dataset

Quality of this work:

- Achieved 92% sensitivity and an AUC of 0.94
- The proposed method is compared with state-of-the-art models, notably achieving an enhanced AUC from 0.936 to 0.945 on the Messidor dataset.

Pros:

1. Implementing CLAHE as a pre-processing step ensures that the input images are appropriately enhanced and standardized before feeding them into the DR monitoring model
2. Efficient Nets are introduced as a state-of-the-art CNN model known for reducing parameters and operations while improving accuracy and speed.
3. Using CLAHE overcomes limitations of other methods like Histogram Equalization (HE) and Adaptive Histogram Equalization (AHE).
4. The obtained results confirm the efficiency of the suggested method without the need for data augmentation.

Drawbacks:

1. Reliance on Messidor-2 and IDRID datasets may restrict the model's ability to generalize to diverse retinal images.
2. Dependence on pre-trained EfficientNet models might not fully adapt to the nuances of retinal image features, potentially limiting performance.
3. Manual tuning of scaling parameters for Efficient Nets, like Efficient Net-B5, poses challenges and may increase computational demands, hindering practical deployment.
4. Need heavy hardware supports like need NVIDIA 1080ti GPU supports.
5. Transformer based model shows better accuracy than this approach.

End-to-End Adversarial Retinal Image Synthesis (2018)

Pedro Costa; Adrian Galdran; Maria Ines Meyer; Meindert Niemeijer; Michael Abràmoff; Ana Maria Mendonça; Aurélio Campilho

Methodology Used:

1. Here proposed an adversarial autoencoder for retinal vessel network synthesis, followed by generation of color retinal images using a generative adversarial network.
2. Utilized Messidor-1 and DRIVE datasets for training.
3. Employed encoder-decoder architecture with U-Net for vessel segmentation.
4. Evaluated through subjective visual assessment and quantitative ISC metric analysis.
5. Found synthetic images to exhibit global consistency, but slightly lower performance in vessel segmentation compared to real images. Future work aims to enhance realism and expand applicability to other medical imaging tasks.

Quality of this work:

ISC Quality Measure on Real/Synthetic Images

	Mean ISC score	Std. dev.
Real Images	0.9832	0.1117
Synthetic Images	0.9671	0.0307

Pros:

1. The adversarial learning framework allows us to model the underlying distribution of plausible retinal images only from training data, without manually interacting with parameters, controlling complex mathematical e models of the retinal anatomy.
2. Once trained, the model improves upon by allowing to generate any amount of realistic retinal images, with associated vessel trees, in an efficient manner.
3. Unlike previous approach, this generate separate parts of the retinal anatomy through the same process, avoiding the combination of complex image processing tasks.

Drawbacks:

1. Trained on a small dataset (614 images) from Messidor-1 database, restricting variability.
2. Synthetic images generated at 256 × 256 resolution, lower than real images.
3. Synthetic vessel networks lack realism, displaying interruptions and inconsistent width variations.
4. Challenges stem from data and computational resource limitations.
5. Future Directions: Incorporate clinical labels into larger datasets for improved realism and applicability.

Data Augmentation for Improving Proliferative Diabetic Retinopathy Detection in Eye Fundus Images

(2020)

Teresa Araújo; Guilherme Aresta; Luís Mendonça; Susana Penas; Carolina Maia; Ângela Carneiro; Ana Maria Mendonça Aurélio Campilho

Methodology used:

- Data augmentation approach by synthesizing synthetic neovascularizations within eye fundus images to simulate grade 4 diabetic retinopathy cases, thereby enhancing the training dataset for DRGraduate
- **DRGraduate**: a deep learning-based grading system, to improve detection accuracy, especially in severe proliferative diabetic retinopathy cases.

Quality of work:

1. High detection accuracy. Increased from 71% to 74%.
2. Better performance in identifying R4-labeled images.
3. Synthetic NVs closely resemble true NVs, enhancing dataset realism.
4. Data integrity is preserved by augmenting R0-R3 images while maintaining original R4 images.

Work Flow

1. Neovascularization Generation:
   - Utilize semi-random generation algorithms with predefined parameters to create synthetic neovascularizations (shapes- Tree like, wheel like, broom like) resembling real structures.
   - Consider characteristics like branching, orientation, and growth to generate diverse NV shapes.

2. Insertion Location and Quantity:
   - Determine insertion points for NVs based on statistical distributions, focusing on regions near the optic disc and surrounding areas.
   - Control the number of inserted NVs per image probabilistically to ensure realism.

3. Color Assignment and Insertion:
   - Match NV colors to surrounding vasculature using a computed color matrix derived from vessel colors.
   - Blend NVs into images while considering factors like vessel type and proximity to the optic disc.

4. Training Data Expansion:
   - Augment original Kaggle DR detection images by adding newly generated NVs, particularly for grade R4 images.
   - Ensure an equitable distribution of NV-inserted images pooled from grades R0-R3.

5. Generated NVs Evaluation:
   - Validate the realism of generated NVs through expert assessment and evaluation of examples.
   - Confirm the resemblance of synthetic NVs to true neovascularizations.

6. Training Approach:
   - Train the DRGraduate model using augmented datasets, optimizing cross-entropy loss.
   - Maintain a balanced representation of DR grades.
   - Validate model performance on multiple datasets.

Drawbacks:

1. Misclassification of other structures as neovascularizations, leading to false positives.
2. Difficulty in distinguishing between neovascularizations and other lesions like IRMAS, especially in early stages.
3. Some neovascularizations with unusual shapes or sizes, may be missed by the mode.
4. Unrealistic color attribution to neovascularizations due to the presence of light artifacts, resulting in color aberrations.

Mohammad Z. Atwany; Abdulwahab H. Sahyoun; Mohammad Yaqub

Supervised method: Binary classification:

1. Quellec et al. used 3 layer CNN architecture with pretrained AlexNet and two networks of Team o_O solutions for the Kaggle challenge .
2. Database: Kaggle DIARETDB1 and private E-ophtha
3. size: (448, 448)
4. Preprocessing: crops, pixel normalization, gaussian filter.
5. Feature extracted: Microaneurysms, hemorrhages, soft and hard exclusion.

1. Xu et al. used stochastic gradient descent as optimizer.
2. Dataset: Kaggle eyepacs.
3. Augmentation: rotation, rescaling, flipping, shearing, translation.
4. Eight 2D conv layer, max pool, fully connected layer softmax.
5. Extracted features: Hard exudates, red lesions, microaneurysms, blood vessels.

1. Jiang et al. Pre-trained CNNs Inception V3, Inception-ResNetV2, ResNet152, Adaboost algorithm for output integration, Adam optimizer.
2. Private dataset used with size 520X520 size
3. Aigmentation:Translation,Rotation,Mirroring,Brightness, contrast, and sharpness adjustments

Supervised method: Multiclass classification:

Study:	Architecture	Dataset	Preprocessing and augmentation	Extracted features
Abràmoff et al.	10 CNNs based on pre-trained InceptionV3	Private dataset	Normalization, resizing to 299 pixels diameter	Referable diabetic macular edema, moderate or worse DR, severe or worse DR, fully gradable
Zhang et al.	ResNet50, InceptionV3, InceptionResNetV2, Xception, DenseNets	13,767 images from an unspecified dataset	Cropping, resizing, histogram equalization, adaptive histogram equalization ,Image enlargement, contrast improvement through contrast stretching for dark images	New fully connected layers on top of pre-trained CNNs, strong model integration
Li et al.	ResNet50 with four attention modules	Messidor dataset and IDRiD dataset	Augmentation, normalization, resizing	Stages of Diabetic Retinopathy Features from ResNet50 used as inputs for attention modules
Pao et al.	Bi-channel CNN with 4 convolutional layers per channel	Kaggle Diabetic Retinopathy dataset (21,123 RGB fundus images)	Resizing to 100x100x3 pixels, flipping, rotation, unsharp masking for gray-level high-frequency parts and green component	Features from green component and gray levels entropy images enhanced with unsharp masking

Self supervised: Binary classification

Study:	Architecture	Dataset	Preprocessing and augmentation	Extracted features
Luo et al.	Self-Supervised Fuzzy Clustering Network (SFCN)	Private dataset (unspecified)	Images resized to (224, 224, 3)	Feature learning module using ResNet50, fuzzy clustering module for self-supervision, reconstruction module with deconvolutional layers.

Self supervised: Multiclass classification

He et al.	CABNet (Category Attention Block Network)	Messidor, EyePACS, DDR	Images resized to (512, 512, 3), Random horizontal flips, vertical flips, rotation	Feature maps from backbone CNN, Global Attention Block (GAB), Category Attention Block (CAB)
Lin et al.	MCG-Net (Graph Convolutional Network with SSL)	ODIR, SSL, GTest		Feature extraction with GCN for classification, SSL for generalization, class correlation capture through CGCN

Transformer based method:

Study:	Architecture	Dataset	Preprocessing and augmentation	Extracted features
Dosovits L. et al.	Vision Transformer (ViT)	Various (unspecified)	Images split into patches (16,16), projected to embeddings with positional markers	Referable diabetic macular edema, moderate or worse DR, severe or worse DR, fully gradable
Sun et al.	Lesion-aware Transformer (LAT)	Messidor-1, Messidor-2, EyePACS	Images resized to (512,512), vertical flips, horizontal flips, random cropping, color jitter Vertical flips, horizontal flips, random cropping, color jitter	Pixel relation-based encoder, lesion filter-based decoder
Kamran et al.	Conditional GAN (ViT-based)	Private dataset	Images of size (576,720), crop size overlapping of (512,512)	Generator with residual, spatial feature fusion, upsampling, downsampling; Transformer encoder blocks for discriminators
Papadopoulos et al.	Transformer-based method for local information extraction	Kaggle EyePACS, Messidor-2, IDRiD	Resizing to 100x100x3 pixels, flipping, rotation, unsharp masking for gray-level high-frequency parts and green component	Attention mechanism to generate heatmaps, focuses on eye regions with lesions

Results: Supervised techniques

Study:	Architecture	Accuracy	F1 Score	Kappa	Comment
Pao et al.	Entropy-based Model	87.37%	0.81		Generates entropies to highlight lesion edges
Zhang et al.	Ensemble of pre-trained networks	96.5%		98.1%	High accuracy, sensitivity, and specificity
Luo et al.	SFCN	81.7%			Uses self-supervised fuzzy clustering
Lin et al.			0.86	38.77%	GCN for classification, SSL for generalization
Li et al.	Single ResNet50	92.6%			Simpler architecture with competitive performance

Results: Transformer

Study:	Architecture	Accuracy	F1 Score	Kappa	Comment
Kamran et al.	VT-GAN	78%			30% better FID and KID scores than SoTA (A2GAN, StarGAN-v2)
MIL-based	MIL Attention Mechanism			95.7%	High-quality attention maps for lesions
MIL-VT	MIL-VT	85.5%	0.94	38.77%	Aggregates patches based on features and attention
LAT	Lesion Aware Transformer (LAT)	96.3%			Uses attention blocks for lesion importance learning

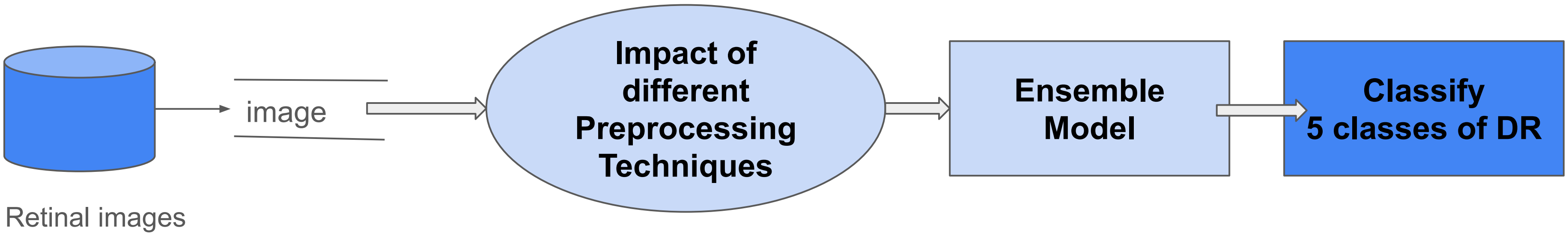
Gaps Identified in Existing Research Work

- Existing methods shows inefficiency dealing with imbalanced data
- Computationally expensive.
- Bias exists for Mild DR cases.
- Poor performance for Proliferative DR detection.
- Challenges face in model Interpretability.
- Accuracy was not meeting expectation.

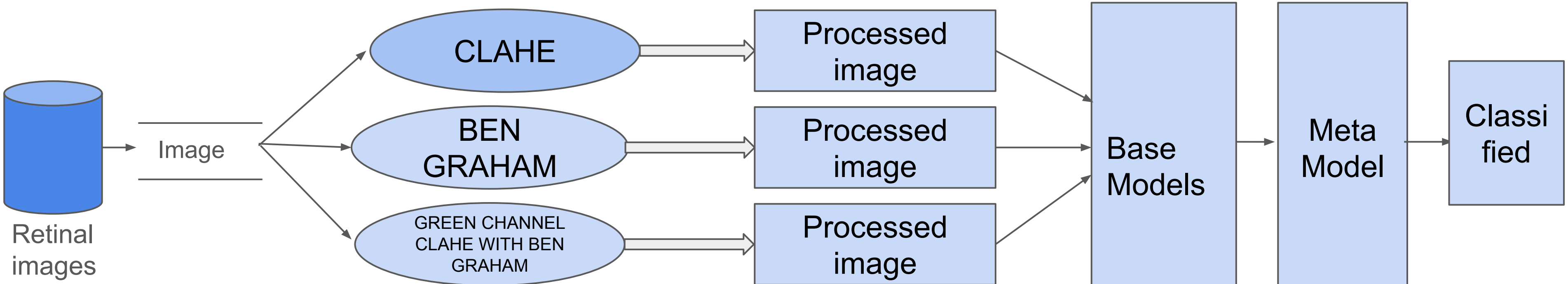
Proposed Approach

Analysis of design requirements

1. DFD 0

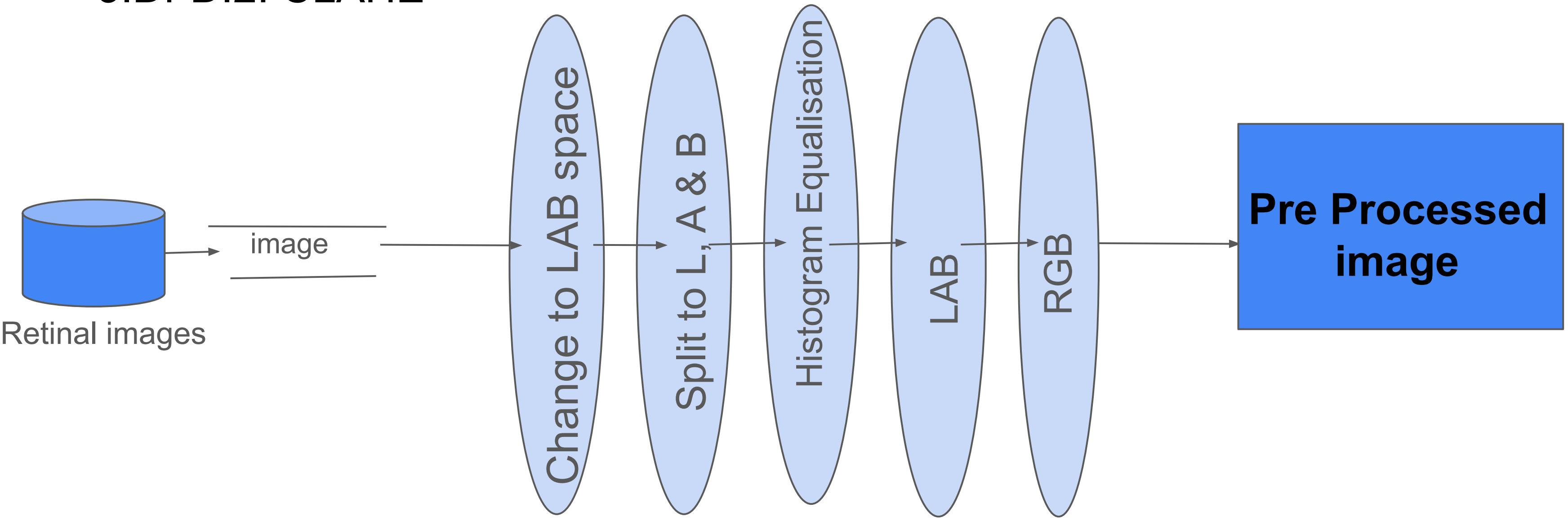


2. DFD 1



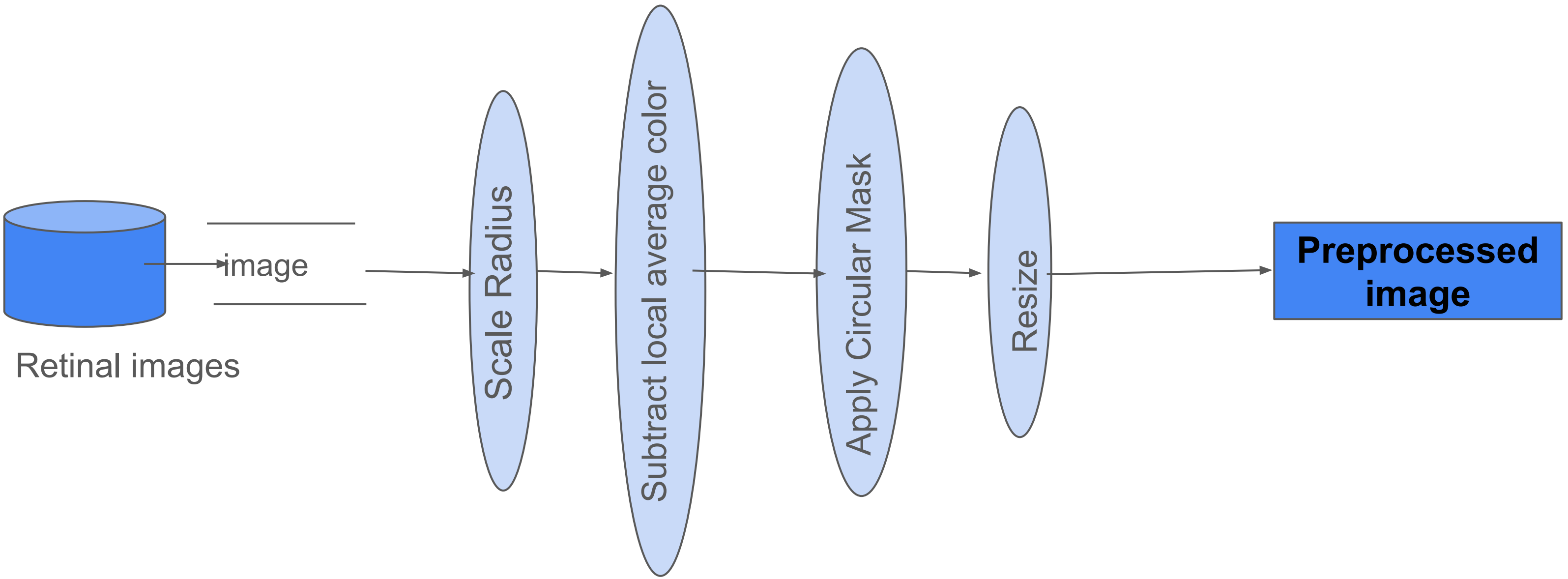
Retinal images
Image
CLAHE
BEN GRAHAM
GREEN CHANNEL CLAHE WITH BEN GRAHAM
Processed image
Processed image
Processed image
Base Models
Meta Model
Classi fied

3.DFD.2: CLAHE

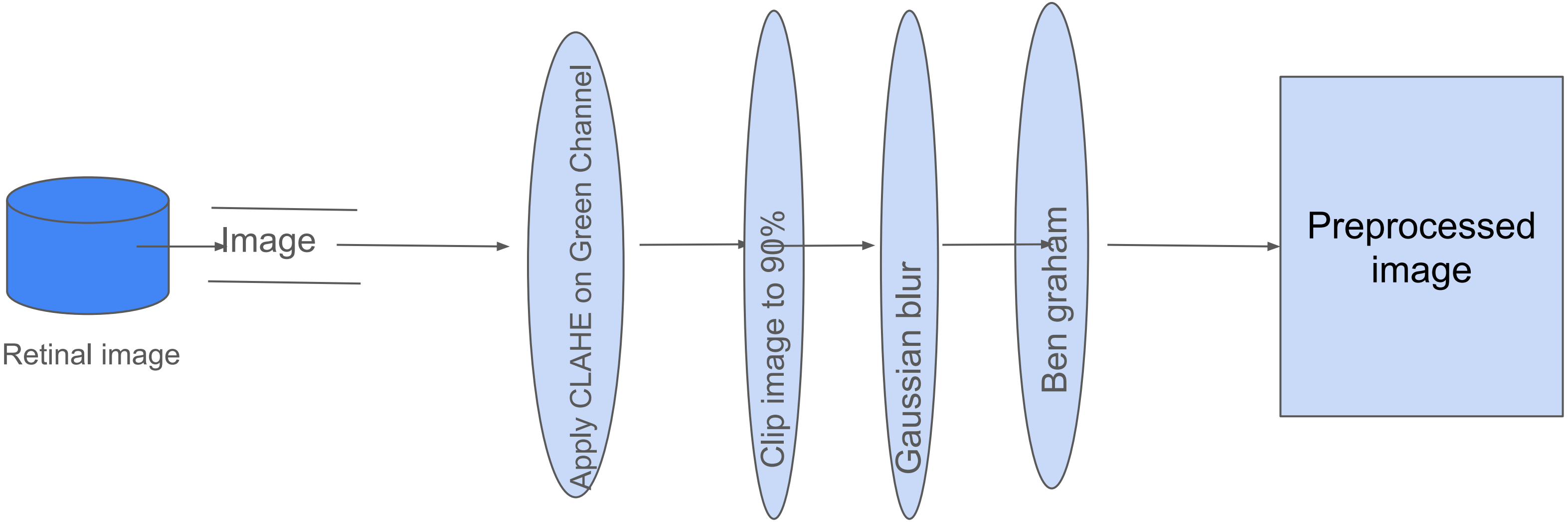


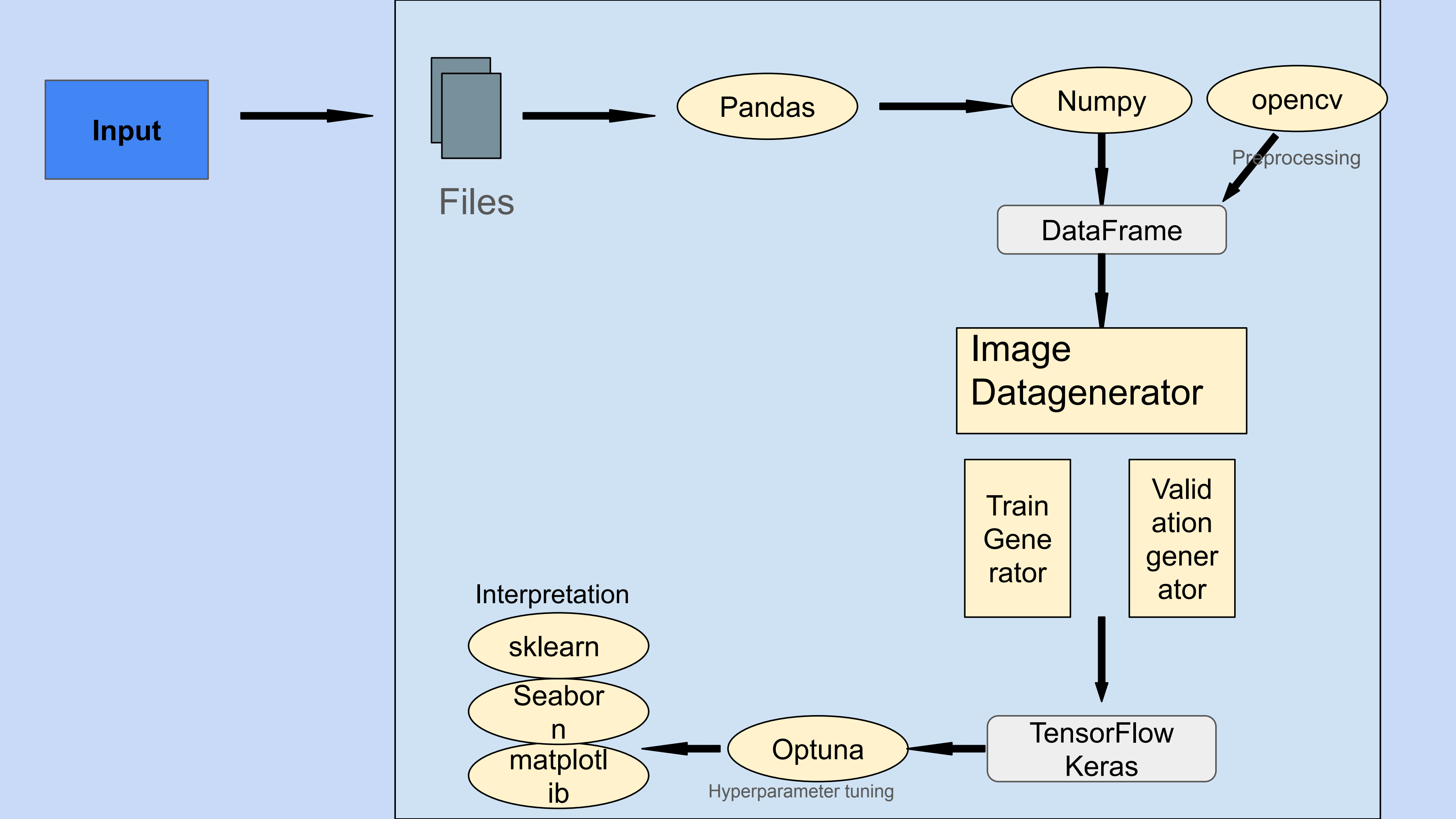
Retinal images
image
Change to LAB space
Split to L, A & B
Histogram Equalisation
LAB
RGB
Pre Processed image

4. DFD.3: Ben Graham:



5. DFD.4: Modified Approach




Input
Files
Pandas
Numpy
opencv
Preprocessing
DataFrame
Image Datagenerator
Train Gene rator
Valid ation gener ator
Interpretation
sklearn
Seabor n
matplotl ib
Optuna
Hyperparameter tuning
TensorFlow Keras

Hardware used:

Local system:

Ram : 8 GB

GPU : NVIDIA GeForce GTX 1650

Software used:

Operating System Type: Microsoft Windows 10
Purpose: Compatible with deep learning framework
Python Version : 3.12.2

Deep Learning Framework:

Framework: TensorFlow and keras
Purpose: Implementing and Training Ensemble Learning Model

Key Libraries: Tensorflow

Keras (Model Implementation)
Opencv-python (image preprocessing)
Numpy, pudus (data manipulation)
Optuna (Hyperparameters tuning)
Matplotlib, Seaborn, sklearn

Development Environment:

Tools : VS Code
Purpose: Writing , debugging and experimenting with code

Environment Identification Development:

Python Version: 3.12.2
Virtual Environment: Created and managed using venv.
Package Management: pip

Methodology:

Data Collection and Preprocessing

Dataset: EyePACS Diabetic Retinopathy dataset

Data Balancing: Combination of random undersampling and random oversampling

Image Preprocessing:

Normal Preprocessing: Standard resizing and normalization

Ben Graham Preprocessing: scaling to a specified radius, subtracting local average color, clipping it to remove boundary effects, and resizing it to a target size.

CLAHE Preprocessing: Contrast Limited Adaptive Histogram Equalization

Combined Preprocessing: Applying a modified technique of CLAHE on green channel, ben graham kernel sequentially

Preprocessing : 1st approach: CLAHE (Contrast Limited Adaptive Histogram Equalization):

Algorithm:

Input: RGB image

1. Convert RGB to Grayscale $I_{\text{gray}}(x,y) = 0.299R(x,y) + 0.587G(x,y) + 0.114B(x,y)$
2. Convert grayscale to 8-bit unsigned integer . $I_{\text{gray\_8u}}(x,y) = \text{round}(I_{\text{gray}}(x,y))$
3. Create CLAHE object clahe with clip limit 2.0 and tile grid size (8, 8)

   The CLAHE transformation $T(i)$ is given by:

   $$T(i) = \frac{\text{clip}(h(i))}{\sum_j h(j)} \times (L-1)$$

4. Apply CLAHE to 8-bit
5. Convert it to 16 bit $I_{\text{CLAHE\_16u}}(x,y) = \text{uint16}(I_{\text{CLAHE}}(x,y))$
6. Convert 16 bit to float32 $I_{\text{CLAHE\_float32}}(x,y) = \text{float32}(I_{\text{CLAHE\_16u}}(x,y))$
7. Convert Float 32 to RGB

   $$I_{\text{RGB}}(x,y) = [I_{\text{CLAHE\_float32}}(x,y), I_{\text{CLAHE\_float32}}(x,y), I_{\text{CLAHE\_float32}}(x,y)]$$

Output: Return RGB image

Preprocessing Steps:	Mathematical formulation
1. Convert RGB to Grayscale	$I_{\text{gray}}(x,y) = 0.299R(x,y) + 0.587G(x,y) + 0.114B(x,y)$
2. Convert grayscale to 8-bit unsigned integer	$I_{\text{gray\_8u}}(x,y) = \text{round}(I_{\text{gray}}(x,y))$
3. Create CLAHE object clahe with clip limit 2.0 and tile grid size (8, 8)	The CLAHE transformation $T(i)$ is given by: $T(i) = \frac{\text{clip}(h(i))}{\sum_j h(j)} \times (L-1)$
4. Apply CLAHE to 8-bit	$I_{\text{LAHE}}(x,y)=\text{CLAHE}(I_{\text{gray\_8u}}(x,y))$
5. Convert to 16 bit	$I_{\text{CLAHE\_16u}}(x,y) = \text{uint16}(I_{\text{CLAHE}}(x,y))$
6. Convert 16 bit to float 32	$I_{\text{CLAHE\_float32}}(x,y) = \text{float32}(I_{\text{CLAHE\_16u}}(x,y))$
7. Convert float 32 to RGB and resize	$I_{\text{RGB}}(x,y) = [I_{\text{CLAHE\_float32}}(x,y), I_{\text{CLAHE\_float32}}(x,y), I_{\text{CLAHE\_float32}}(x,y)]$

Preprocessing : 2nd approach: Ben Graham's Preprocessing

Algorithm:

1. **Calculate Horizontal Sum**: Compute the sum of pixel values along the width at the vertical center. $$x(j) = \sum_{k=0}^{C-1} \text{img}\left(\left\lfloor \frac{H}{2} \right\rfloor, j, k\right)$$

   Determine Radius: $$r = \frac{\sum\left(x > \frac{\text{mean}(x)}{10}\right)}{2}$$

   Compute Scaling Factor: $$s = \frac{\text{scale}}{r}$$

   Resize Image.

2. **Apply Gaussian Blur (0, 0) and standard deviation 10**
3. **Add weights and subtract the local average color**

$$\text{result} = 4 \cdot \text{img} - 4 \cdot \text{blurred\_img} + 128$$

4. **Clip the image to 90% to remove boundary effects by applying circular mask centered at the image center with a radius of 270 pixels.**

$$\text{mask}[x, y] = \begin{cases} 255 & \text{if } (x - \text{center}_x)^2 + (y - \text{center}_y)^2 \leq 270^2 \\ 0 & \text{otherwise} \end{cases}$$

5. **Resize image to (224, 224)**

Preprocessing steps:	Mathematical formulation
1. Compute the sum of the pixel values among the width at the vertical center.	$x(j) = \sum_{k=0}^{C-1} \text{img}\left(\left\lfloor \frac{H}{2} \right\rfloor, j, k\right)$
2. Compute the radius	$r = \frac{\sum(x > \frac{\text{mean}(x)}{10})}{2}$
3. Compute the scaling factor	$s = \frac{\text{scale}}{r}$
4. Resize the image	resize(img, scaleX1/r)
5. Apply Gaussian blur	GaussianBlur(img,(0,0),10)
6. Add weights and subtract the local average	$\text{result} = 4 \cdot \text{img} - 4 \cdot \text{blurred\_img} + 128$
7. Clip the image	$\text{mask}[x,y] = \begin{cases} 255 & \text{if } (x - \text{center}_x)^2 + (y - \text{center}_y)^2 \leq 270^2 \\ 0 & \text{otherwise} \end{cases}$
8. Resize the image	cv2.resize(clip_img, target_size)

Preprocessing : 3rd approach: Modified approach

Algorithm:

1. **Compute the histogram H(k) of the green channel image, where Hk represents pixel intensities ranging from 0 to 255.** $H(k) = \sum_{x=1}^{M} \sum_{y=1}^{N} \delta(I_g(x, y) - k)$ Where M and N are dimension of the image, Ig(x,y) intensity of green channel

2. **Convert Green Channel to 8-bit Unsigned Integer**

$$G_{8u}(x, y) = \mathrm{round}\left(\frac{255 \times G(x,y)}{\max(G)}\right)$$

3. **CLAHE Function**

The CLAHE transformation $T(i)$ is given by:

$$T(i) = \frac{\mathrm{clip}(h(i))}{\sum_j h(j)} \times (L - 1)$$

CLAHE on Green Channel

$$G_{\mathrm{CLAHE}}(x, y) = \begin{cases} \mathrm{HE}(G(x, y)), & \text{if } H(G(x, y)) \leq \mathrm{clipLimit} \\ G(x, y), & \text{otherwise} \end{cases}$$

4. $$G_{\text{CLAHE}}(x, y) = \text{CLAHE}(G_{\text{original}}(x, y), \text{clipLimit} = 2.0, \text{tileGridSize} = (8, 8))$$

5. Convert the CLAHE Output to float32 $I_{\text{CLAHE, float32}} = \text{astype}(I_{\text{CLAHE, 16u}}, \text{float32})$

6. Convert Grayscale Back to RGB $I_{\text{RGB}}(x, y) = [I_{\text{gray}}(x, y), I_{\text{gray}}(x, y), I_{\text{gray}}(x, y)]$

7. Clip the image to 90% by applying circular mask.

8. Ben Graham's Kernel Used : $$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

9. Gaussian Blur : (3 3) kernel with std deviation 10

10. Return processed image.

Preprocessing steps	Mathematical formulation
1. Compute the histogram of the green channel image ranging from 0-255	$H(k) = \sum_{x=1}^{M} \sum_{y=1}^{N} \delta(I_g(x, y) - k)$
2. Convert green channel to 8 bit unsigned integer	$G_{8u}(x, y) = \text{round}\left(\frac{255 \times G(x,y)}{\max(G)}\right)$
3. CLAHE function	The CLAHE transformation $T(i)$ is given by: $T(i) = \frac{\text{clip}(h(i))}{\sum_j h(j)} \times (L - 1)$
4. CLAHE on the green channel	$G_{\text{CLAHE}}(x, y) = \begin{cases} \text{HE}(G(x, y)), & \text{if } H(G(x, y)) \leq \text{clipLimit} \\ G(x, y), & \text{otherwise} \end{cases}$
5. CLAHE apply on the image	$G_{\text{CLAHE}}(x, y) = \text{CLAHE}(G_{\text{original}}(x, y), \text{clipLimit} = 2.0, \text{tileGridSize} = (8, 8))$ $I_{\text{CLAHE, float32}} = \text{astype}(I_{\text{CLAHE, 16u}}, \text{float32})$
6. Convert to float32	$I_{\text{RGB}}(x, y) = [I_{\text{gray}}(x, y), I_{\text{gray}}(x, y), I_{\text{gray}}(x, y)]$
7. Convert Grayscale back to RGB	

2. Data Augmentation:

Normalization: 1./255

Rotation range : 15 degree

Shear range : 0.1

Zoom range: 0.1

Horizontal flip

Vertical flip

Model Selection and Training:

Model Used:

1. InceptionV3
2. ResNet50
3. EfficientNet-B5
4. DenseNet169

Pretrained on ImageNet

Transfer learning : Fine tuning

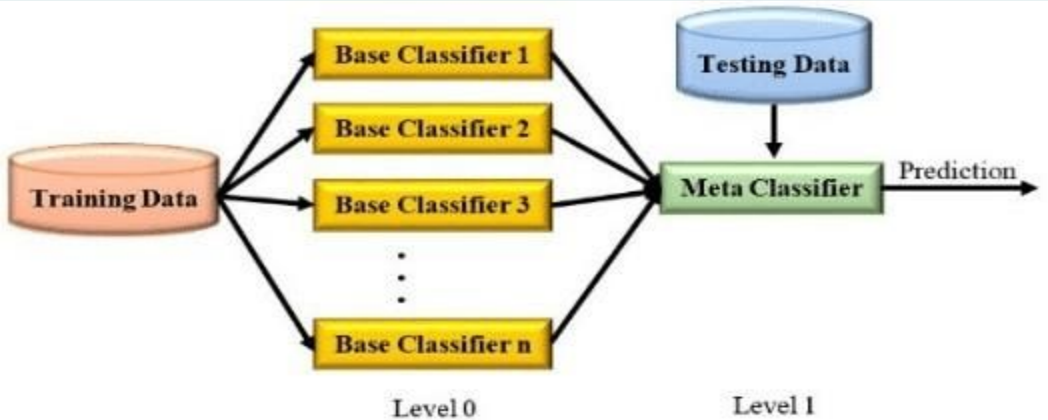
Combined by Ensemble model.

Ensemble technique used : Stacking Ensemble

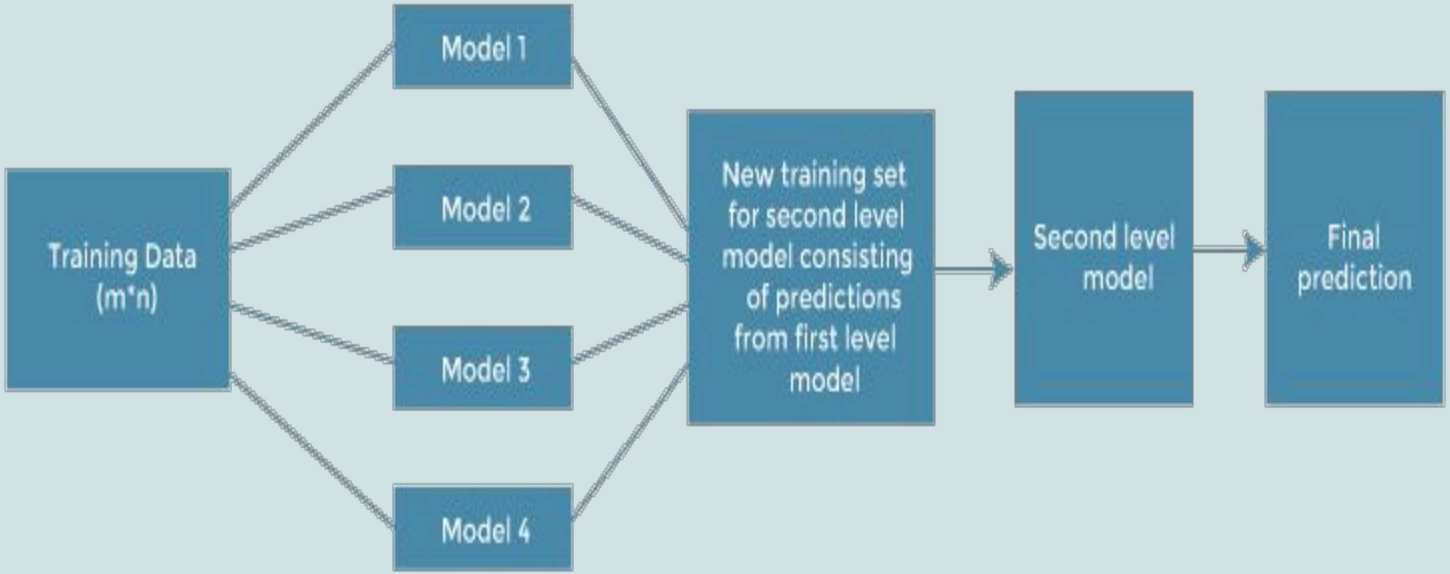
- Combine the prediction of several other learning algorithms.
- All the above algorithms are trained using available data, then a combiner algorithm is trained to make a final prediction using all the prediction of the other algorithms as additional input.

- First step- Train multiple base models, and merged their predicted results called Meta dataset

- Second step- Using meta dataset to build meta learner and do final prediction.



Model prototype:



Ensemble model workflow:

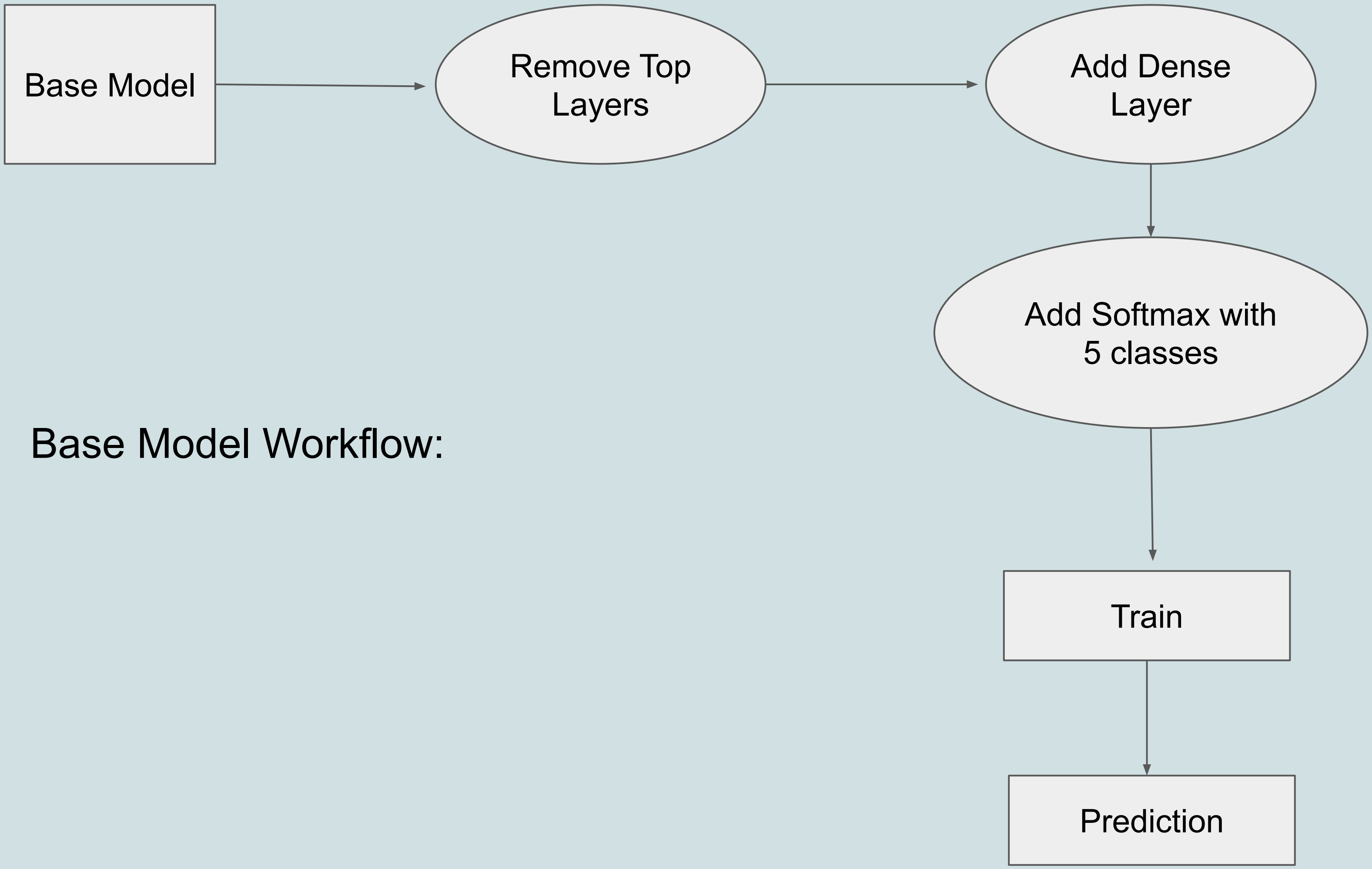
Base Model

1. Load Model
2. Remove the top layer .
3. Add additional Layer With Softmax with 5 Classes . This will personalize the inception model for the problem
4. Train it with the dataset

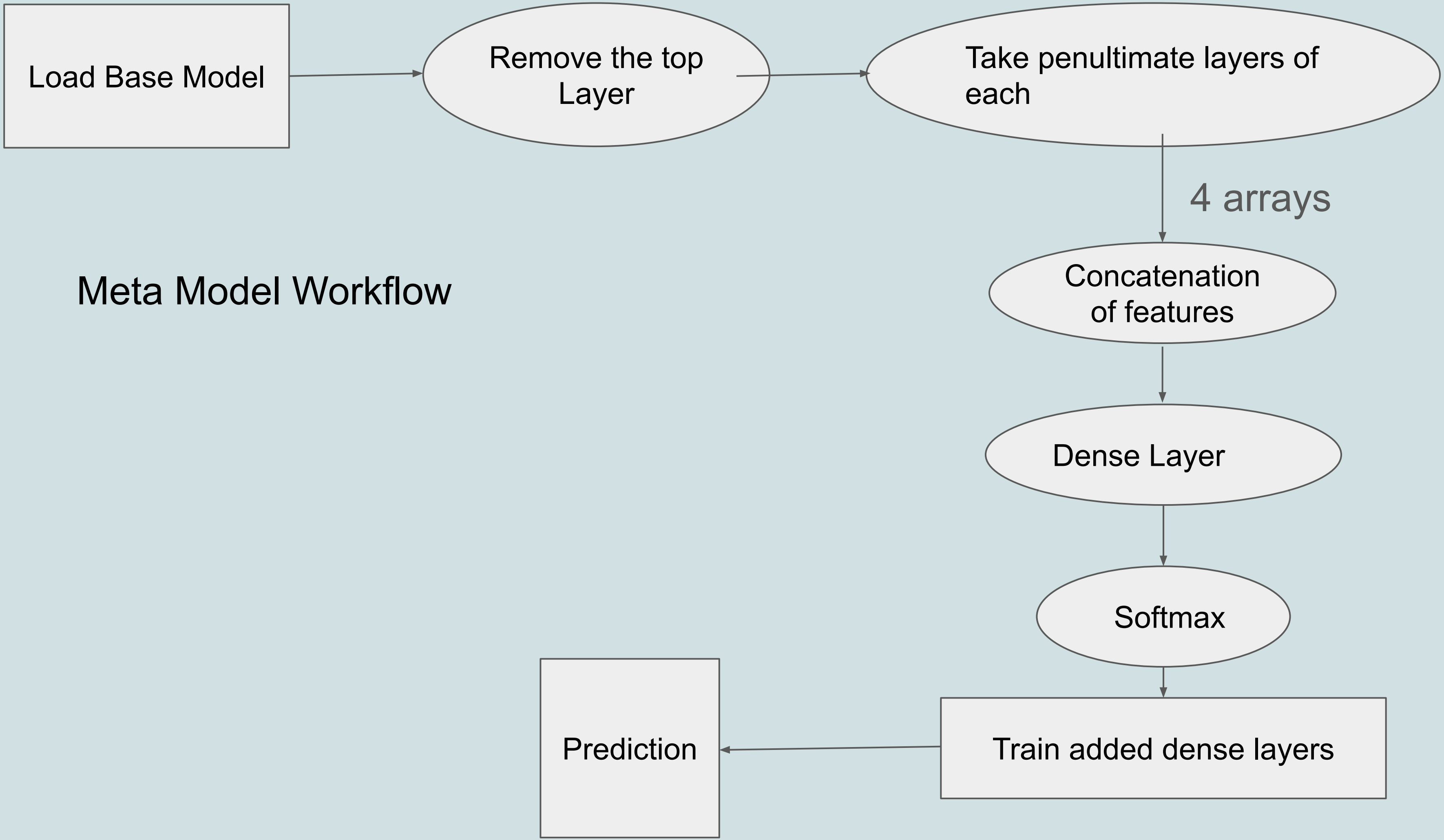
Ensembling

1. Load base models
2. Remove the top layer which previously added.
3. Second last layer output is taken. (Array)
4. For 4 models we will get 4 latent features.
5. Concatenate those output.
6. It is then given to fully connected layer with new soft max layer
7. Train the added dense layer alone (freeze the training of the layers in base model)
8. prediction.

Base Model Workflow:



Meta Model Workflow



Transfer Learning Methodology used:

Inductive Transfer learning: All base models are pretrained.

The pre-trained model learns general features that are useful for a range of tasks. These Pretrained models are then fine tune for specific tasks.

Benefits:

1. Reduced training time and computational resource
2. Improve model performance

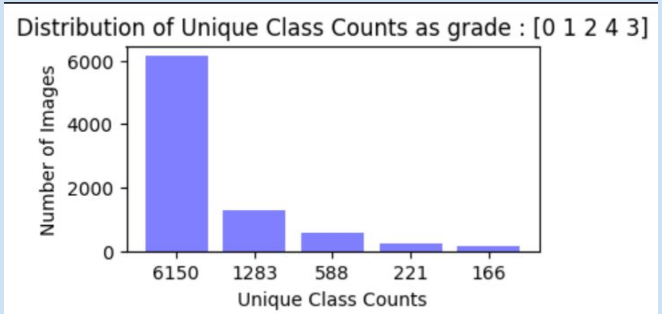
Design and Implementation of Project work

Dataset: Total 35,126 (88 GB)

Image size: Varying size(example: 3168 X 4752 X 2)

Eyepac dataset

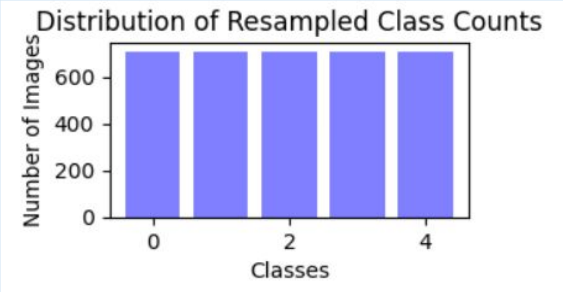
Imbalance



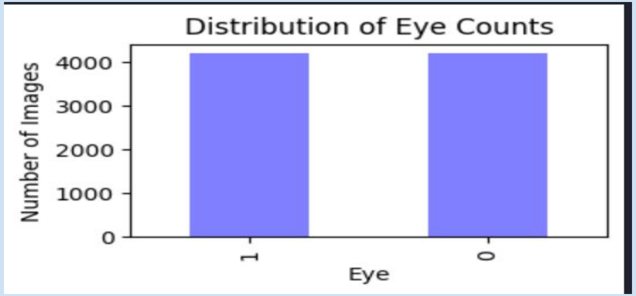
5 classes are there.

After Balancing the dataset:

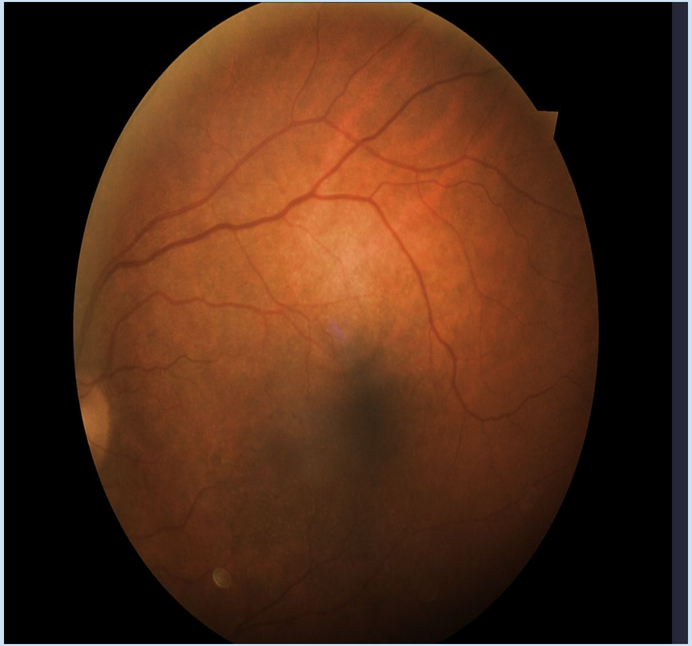
Each class have 708 images.



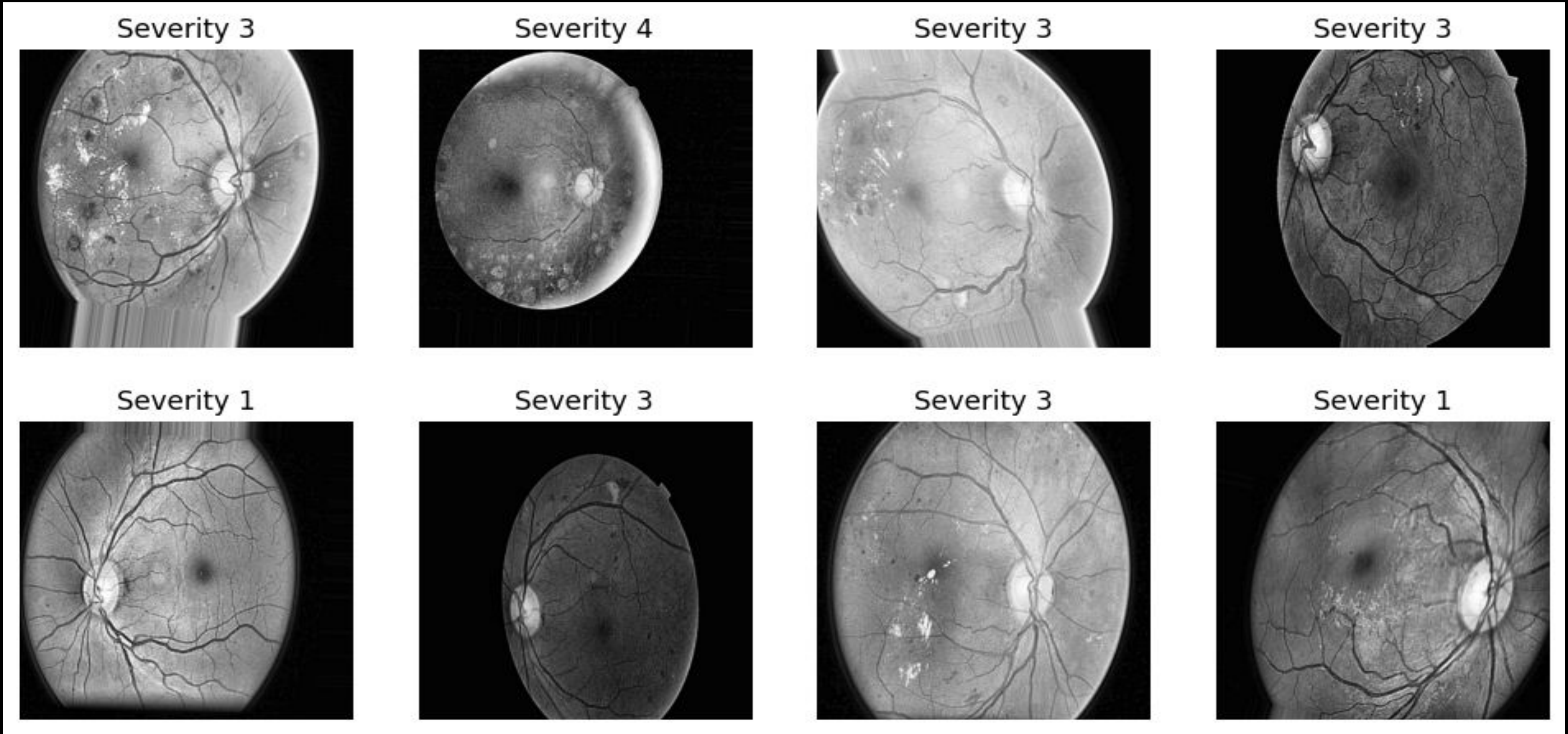
Made sure both eye have equal distribution.



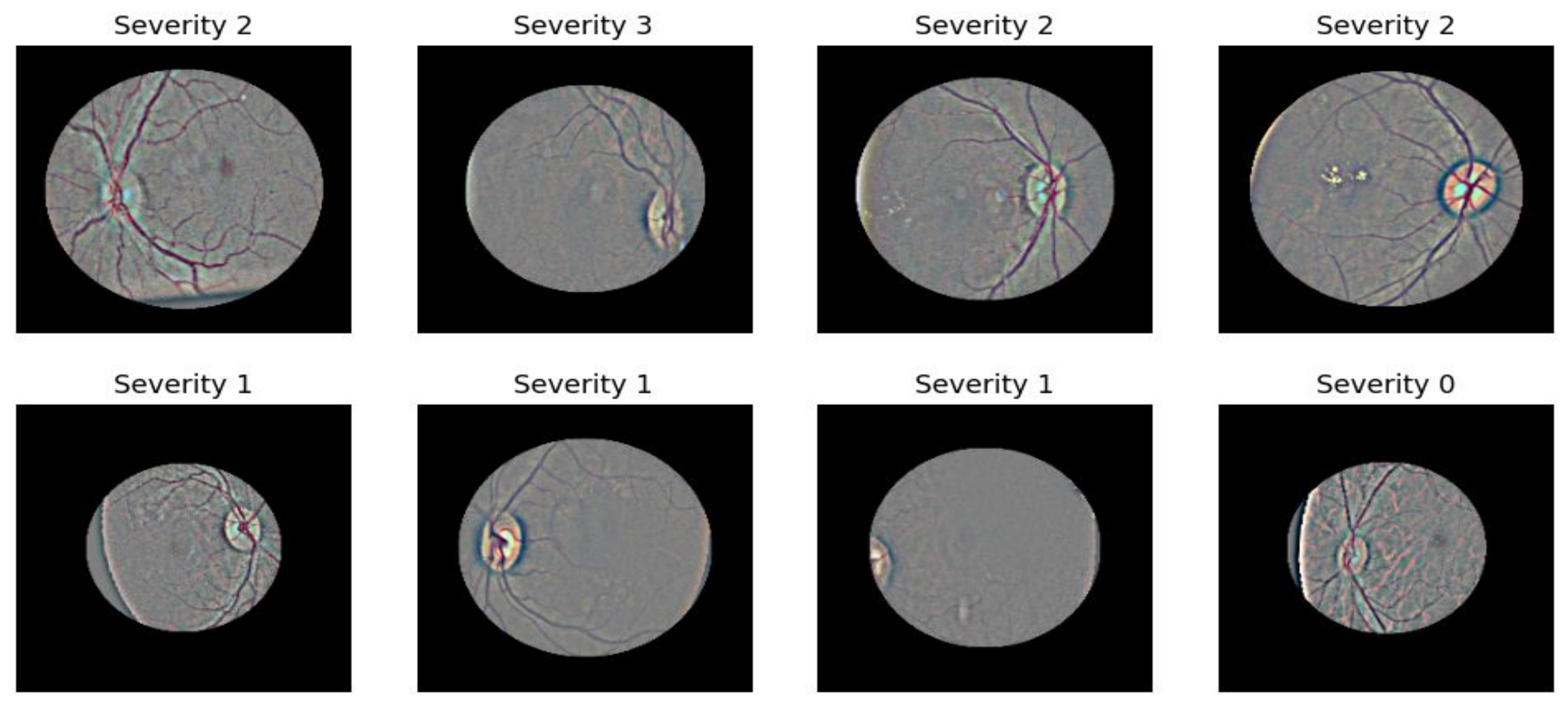
Original sample image:



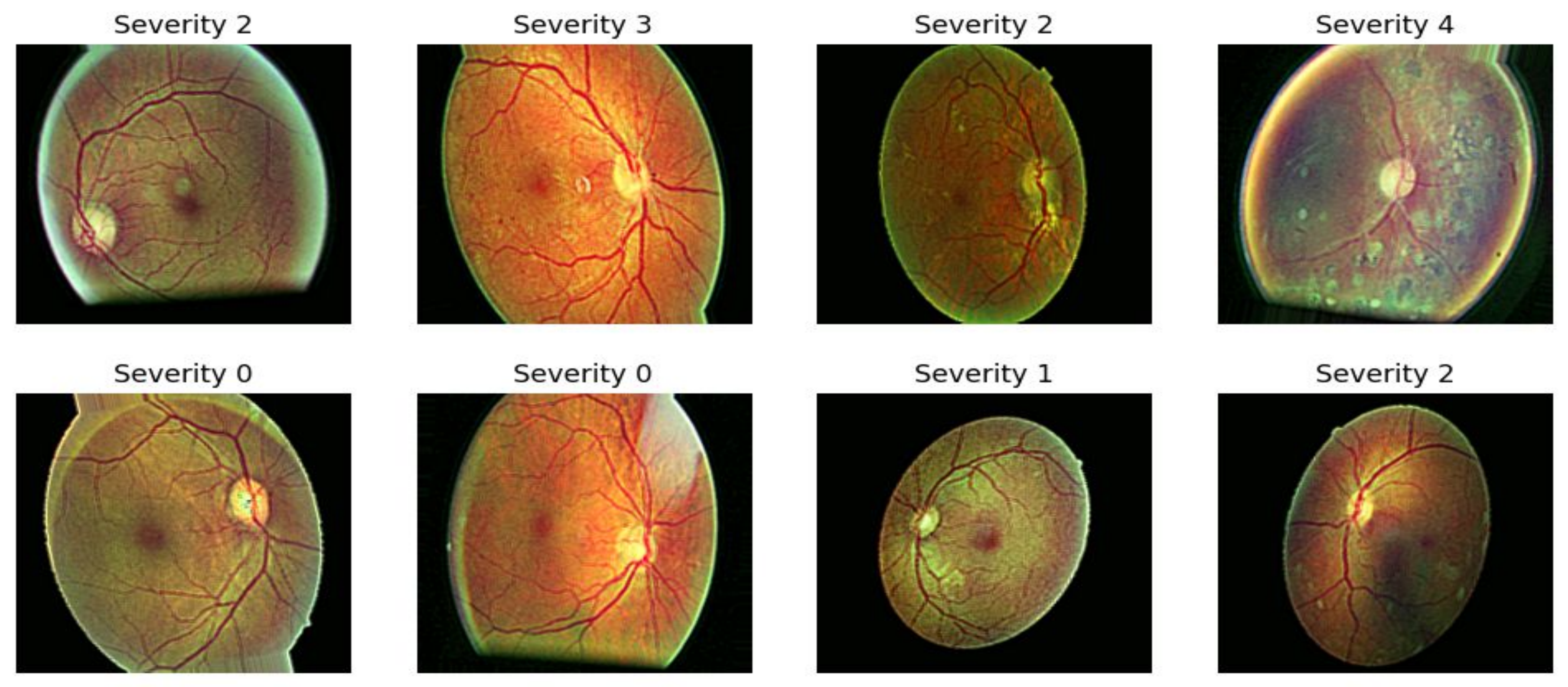
Preprocessing approach 1:

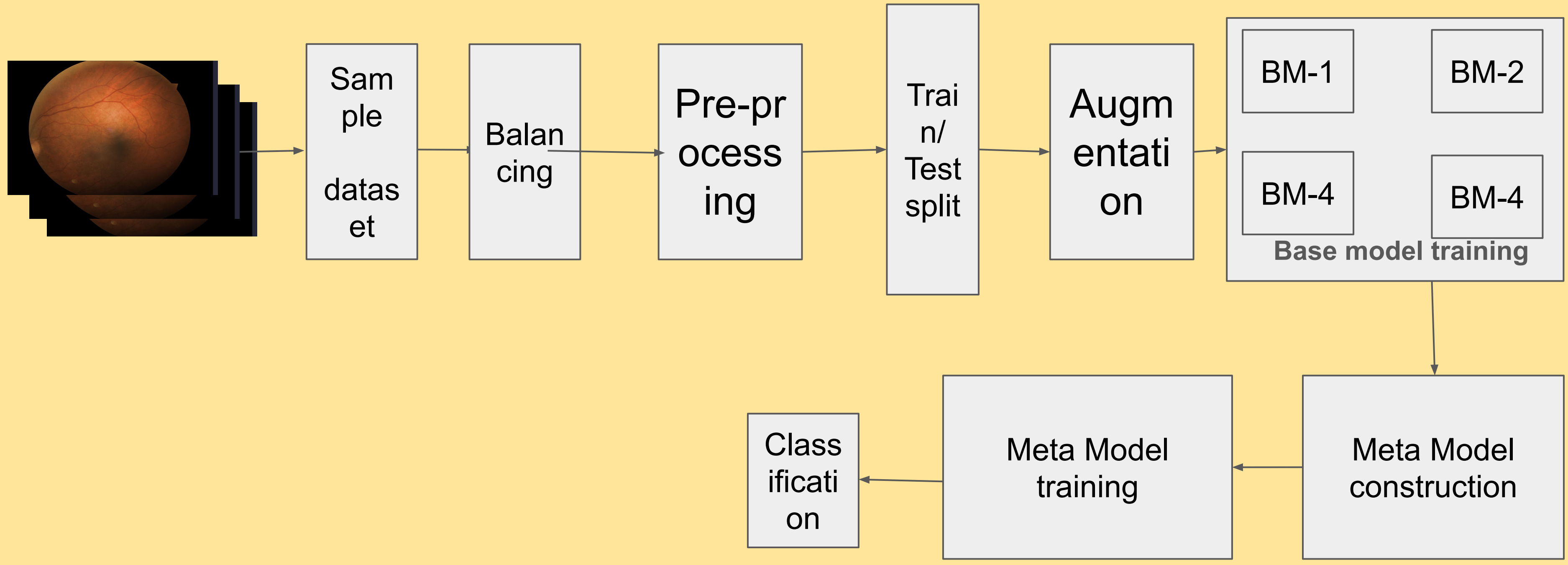


Preprocessing approach 2:

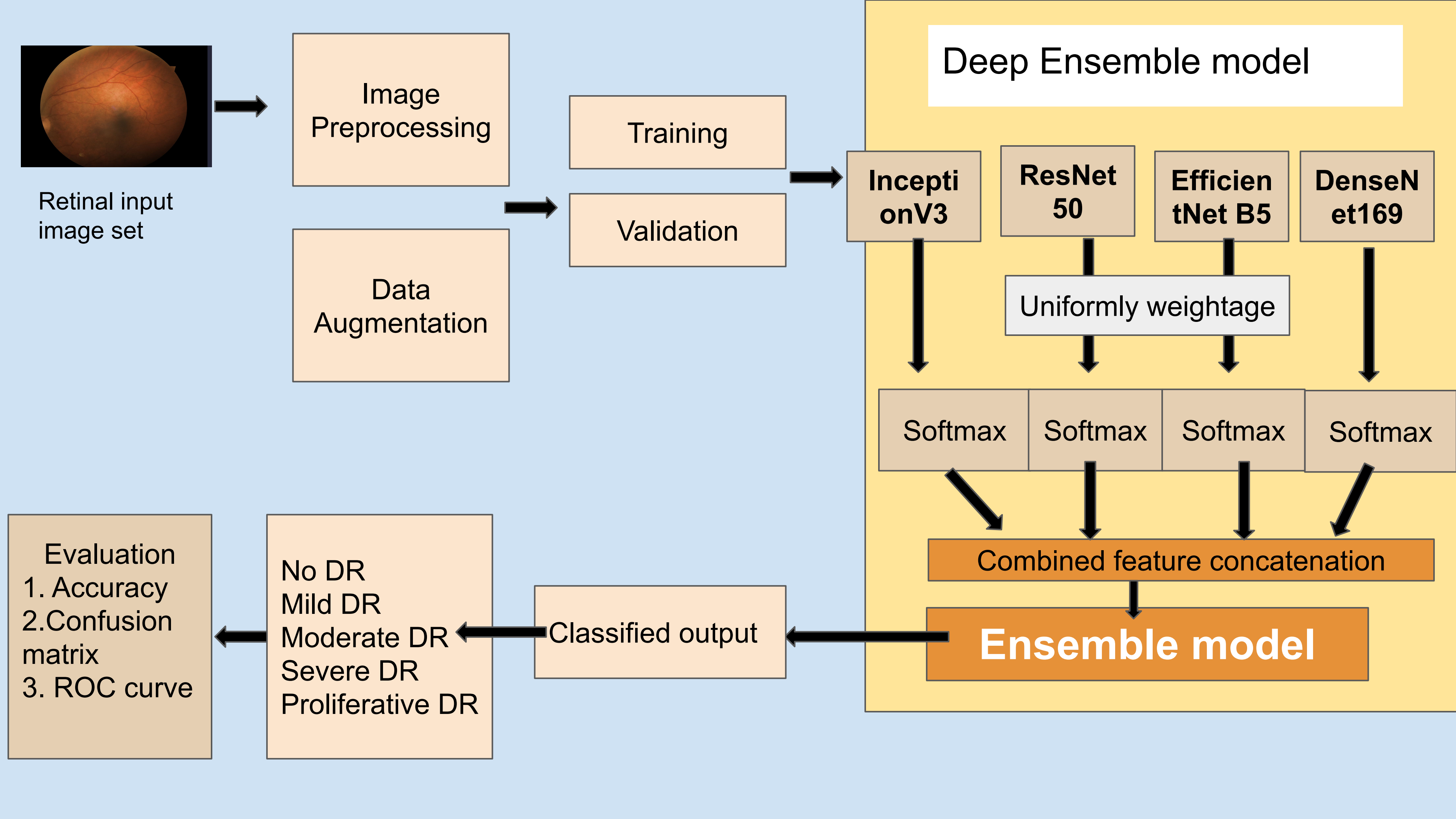


Preprocessing approach 3:





Sample dataset
Balancing
Pre-processing
Train/ Test split
Augmentation
BM-1
BM-2
BM-4
BM-4
Base model training
Meta Model construction
Meta Model training
Classification



Retinal input image set
Image Preprocessing
Data Augmentation
Training
Validation
Deep Ensemble model
InceptionV3
ResNet 50
EfficientNet B5
DenseNet169
Uniformly weightage
Softmax
Softmax
Softmax
Softmax
Combined feature concatenation
Ensemble model
Classified output
No DR
Mild DR
Moderate DR
Severe DR
Proliferative DR
Evaluation
1. Accuracy
2.Confusion matrix
3. ROC curve

1. Experimented with Ben Graham’s preprocessing

Base Model architecture

Load Pre trained model. Input : 224 X 224

Dense layer :

Global average pooling 2D
Dense 1024 (activation= ReLu, Regularization L2= 0.01
BatchNormalization
DropOut = 0.2
Dense 512 (activation= ReLu, Regularization L2= 0.1
BatchNormalization
DropOut = 0.2
Softmax

Batch size: 16, Learning Rate : 0.0001, Optimizer: Adam, Loss function: categorical cross entropy

Epocs: 100 for all

`ReduceLROnPlateau :` Reduce the learning rate when a monitored metric has stopped improving

Meta Model :

Load all 4 models.

Concatenation layer

2 Dense Layer, L2 regularization 0.5, drop out 0.5

Softmax

Multi Input Generator

Training Data

Batch size: 16; Learning rate: 0.001; Optimizer: Adam

Epocs:100

Loss Function: categorical cross entropy

Ensemble model training with ben graham’s preprocessing

Total Data: 35126

After balancing: 3540

Each class:708

Image size=(512,512)

Batch size=32

Number of training set= 2832

Number of validation set=708

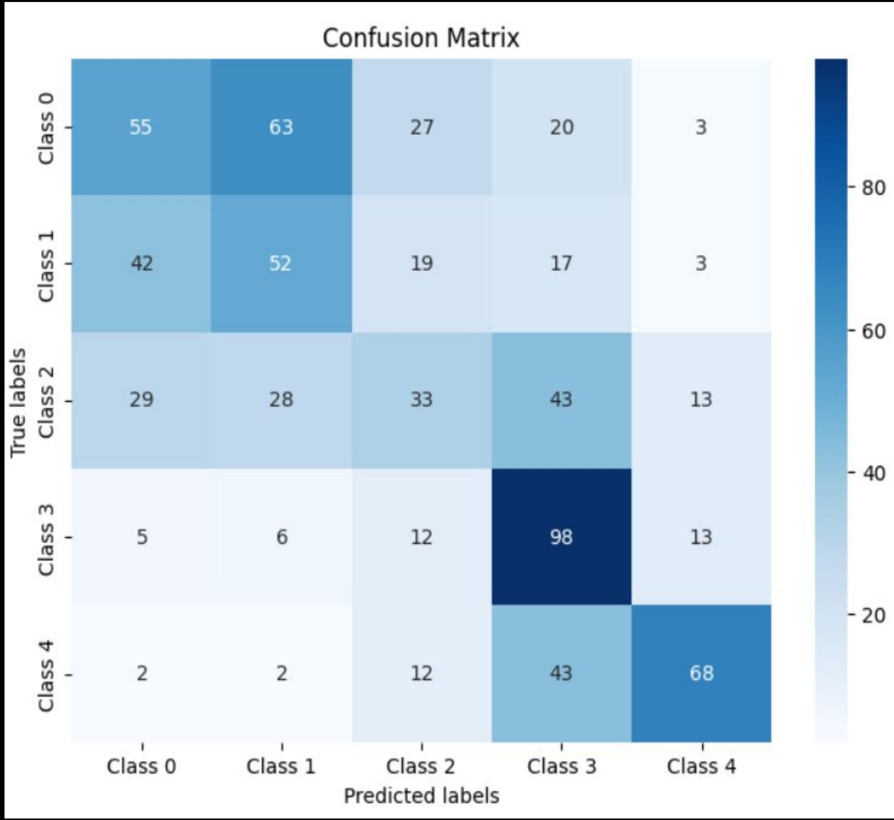
Performance Analysis of classification with pre-processing by BenG approach:

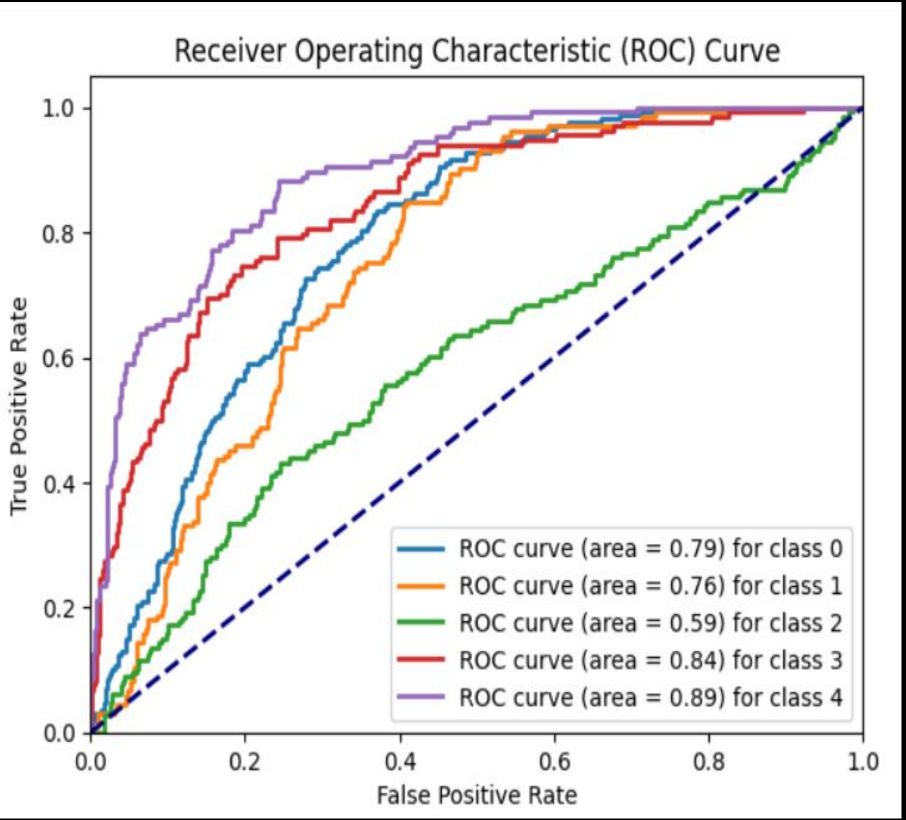
	Validation Accuracy	Validation Loss
InceptionV3	44%	1.88
ResNet50	29%	3.20
EfficientNet-B5	21%	1.77
DenseNet-169	45%	1.70
Ensemble Model	47%	2.00

Accuracy metrics

Precision	Recall	F1 score	Accuracy
0.45	0.45	0.46	0.47

For Preprocessing Approach Ben Graham’s





Modified approach tested with InceptionV3 in 100 epochs

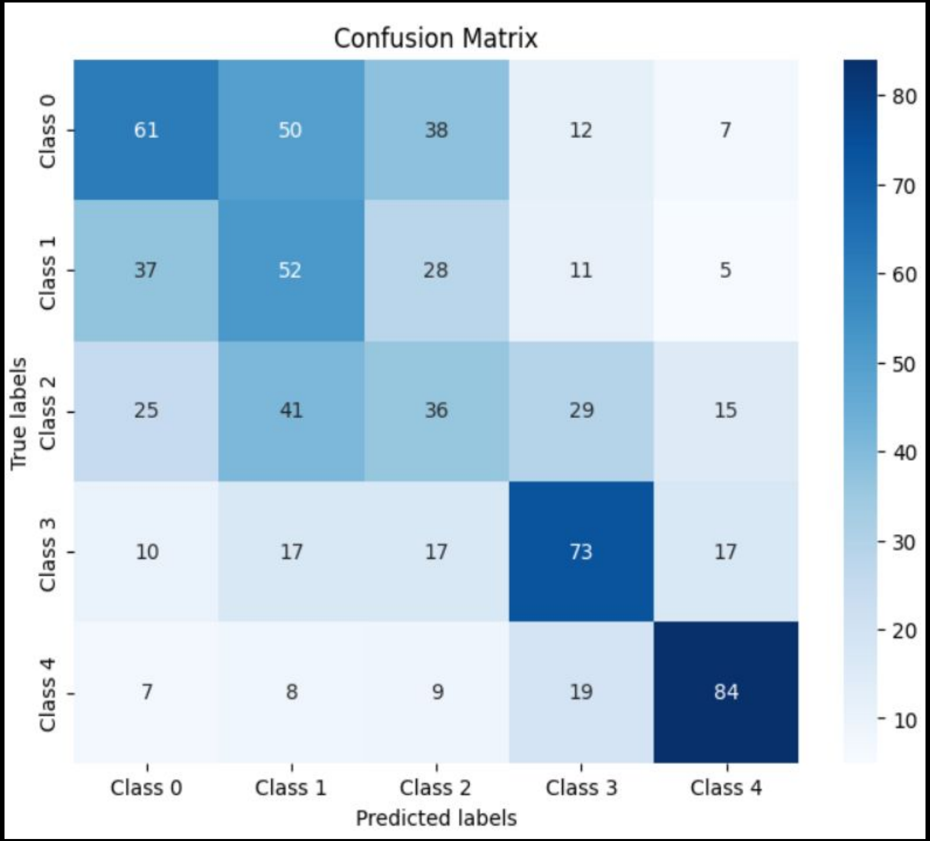
(512X512)

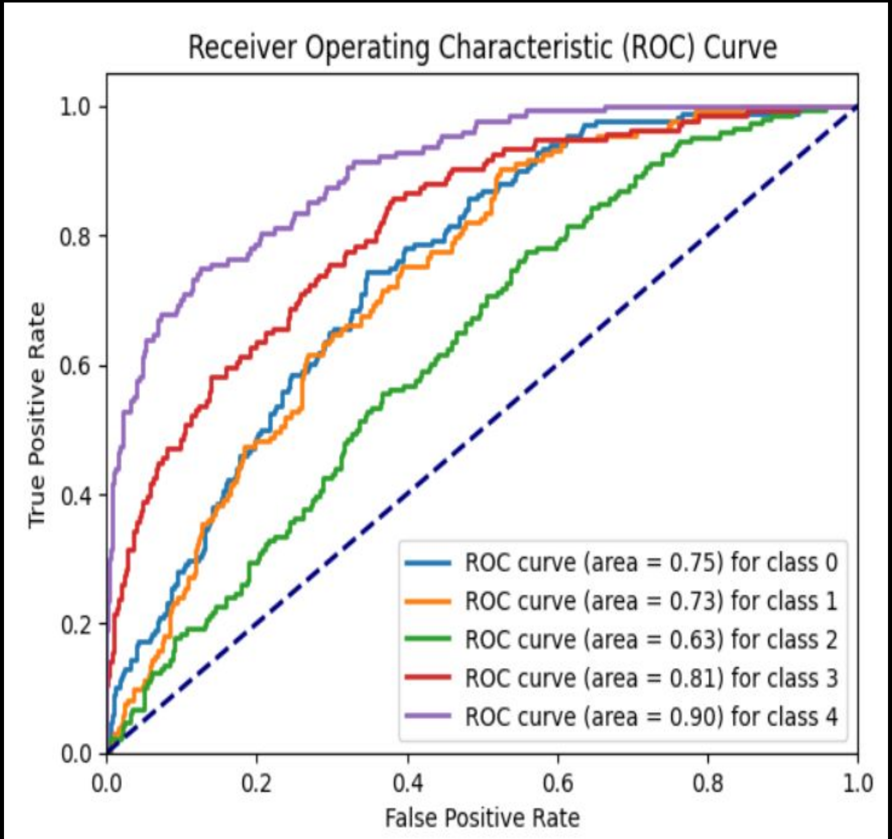
Validation Accuracy: 43%

Validation Loss : 1.84

Precision	Recall	F1 score	Accuracy
0.43	0.43	0.43	0.43

1. Modified Approach





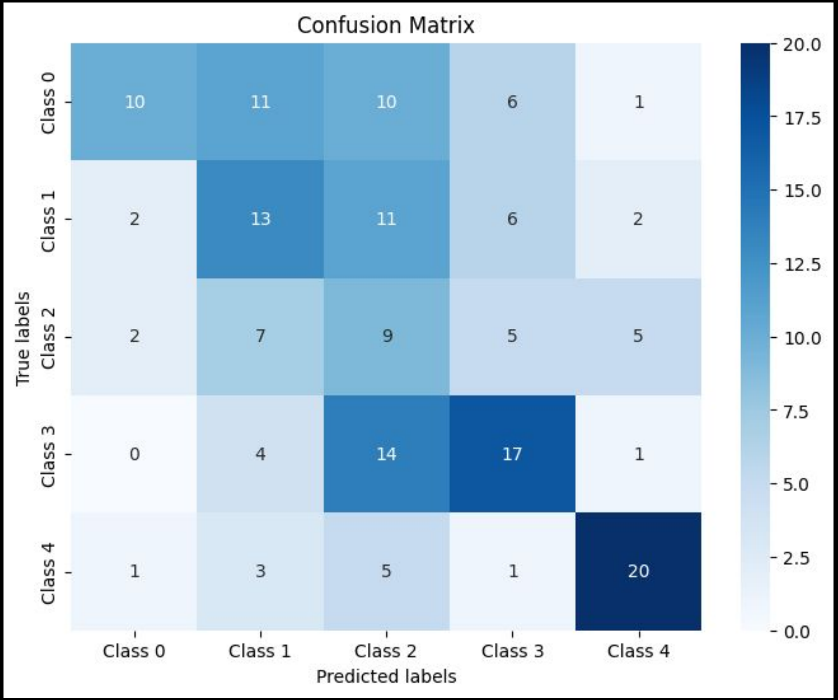
Clahe approach of preprocessing with a sample data(8,408) (224X224)

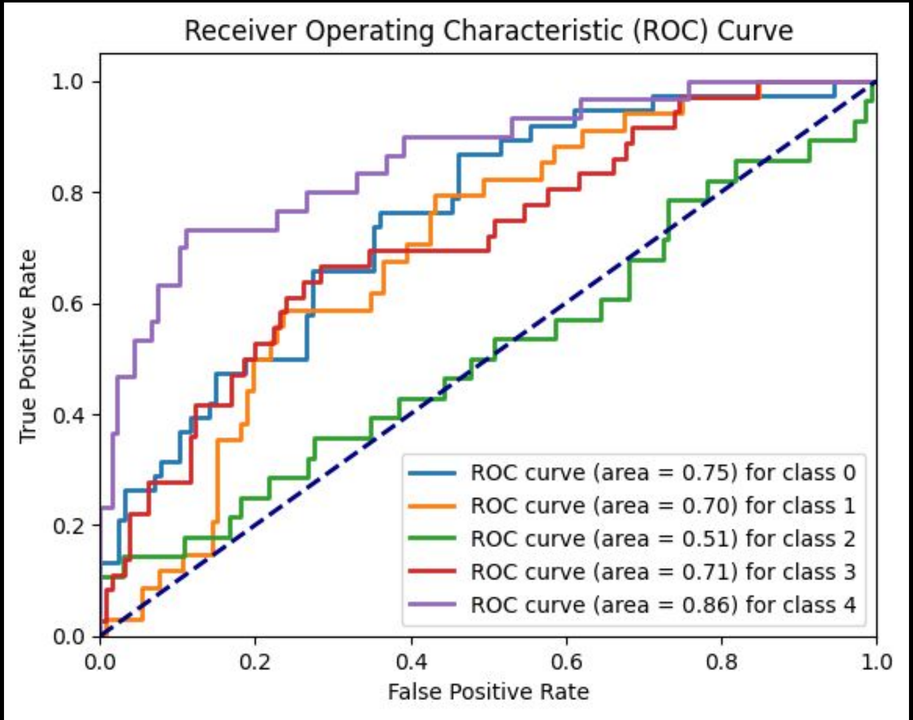
After balancing: each class 166, total 830 retinal image

Evaluation:

Precision	Recall	F1 score	Accuracy
0.48	0.42	0.43	0.42

For Preprocessing Approach CLAHE





Comparison:

Approach 1: Clahe : This approach works better in class 0 and class 1.
Works good for class 4 also but false negative rate is high.

Approach 2: Ben graham's approach: This method shows comparative better results for all classes and for class 3 and 4 it is giving a high yield.

Approach 3: Modified approach: Significantly works better for grade 2, grade 3 and for grade 4 the false detection also spread within severe range.

Scope of Ensemble model:

1. Combining multiple models can lead to more accurate and reliable predictions, have potential to implementation in clinical settings.

2. Ensemble models can integrate various perspectives, enabling a **comprehensive understanding of the relationships** between different regions of medical images.

3. By aggregating the strengths of individual models, ensemble methods can provide more stable and **robust predictions**, **reducing the likelihood of errors in diagnosis.**

4. By implementing transfer learning the we can modify the already proven effective pre trained for our own task resulting development of effective models even with limited medical data.

Limitations of Ensemble model:

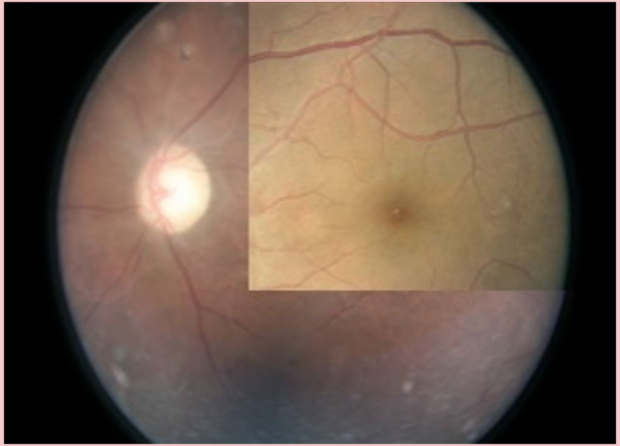
1.Ensemble models often **require a large amount of data** to train effectively, which can be a limitation in the medical field where labeled data is scarce.

2. Training and inference with ensemble models can be **computationally expensive, requiring significant hardware resources.**

3. Ensemble models might **struggle with localizing specific features** in medical images, which is crucial for precise diagnosis.

4. Transformer architecture performs significantly better result.

DR-classification with Data-Augmentation and Ensemble training



1. Sample of Data (8,408) taken
2. Data distribution :

   Class 0: 6150

   Class 1: 221

   Class 2: 1283

   Class 3: 588

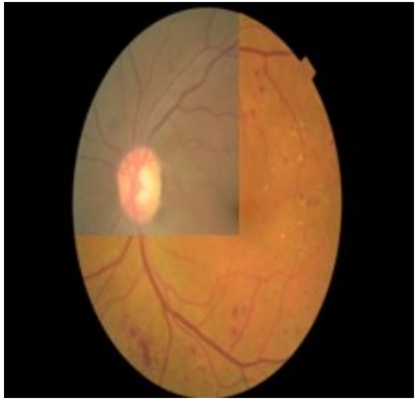
   Class 4: 166

3. Cut-Mix augmentation applied in Class: 1, 2, 3
4. After Cut-Mix augmentation class 1,2,3 each have : 800
5. After Balancing each class having : 800 samples
6. Total: 4000 samples,

Cut-mix augmentation:

Additional images in class 1 : 212, Class 3: 579, Class 4: 634

Sample:

Class 1	Class 3	Class 4
		

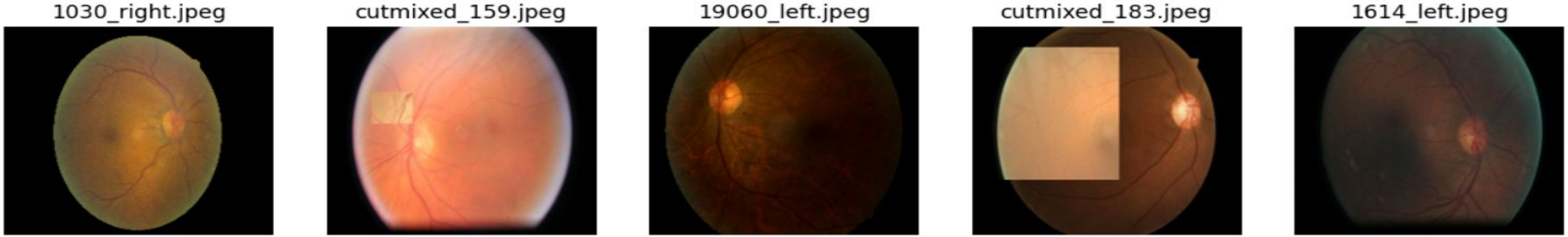
Cut-Mix function:

Step	Operation	Mathematical Formulation
1	Random center point	$x = \mathrm{random}(0, W), y = \mathrm{random}(0, H)$
2	Random value for mixing	$b = \mathrm{random}(0, 1)$
3	Compute width of region	$\mathrm{WIDTH} = \sqrt{1 - b} \times H$
4	Compute boundaries of region	$xa = \max(0, x - \frac{\mathrm{WIDTH}}{2}), xb = \min(W, x + \frac{\mathrm{WIDTH}}{2}), ya = \max(0, y - \frac{\mathrm{WIDTH}}{2}), yb = \min(H, y + \frac{\mathrm{WIDTH}}{2})$
5	Extract left part from `image1`	$\mathrm{one} = \mathrm{image1}[ya : yb, 0 : xa, :]$
6	Extract patch from `image2`	$\mathrm{two} = \mathrm{image2}[ya : yb, xa : xb, :]$
7	Extract right part from `image1`	$\mathrm{three} = \mathrm{image1}[ya : yb, xb : W, :]$
8	Concatenate horizontally	$\mathrm{middle} = \mathrm{concat}(\mathrm{one}, \mathrm{two}, \mathrm{three}, \mathrm{axis} = 1)$
9	Concatenate vertically	$\mathrm{img} = \mathrm{concat}(\mathrm{image1}[0 : ya, :, :], \mathrm{middle}, \mathrm{image1}[yb : H, :, :], \mathrm{axis} = 0)$

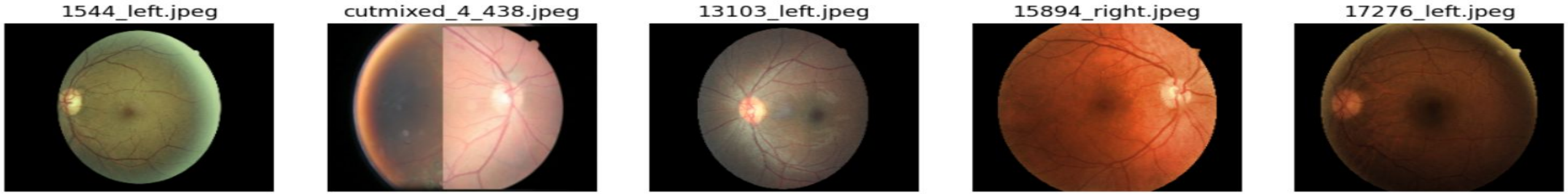
Training set: 3200

Validation set: 800

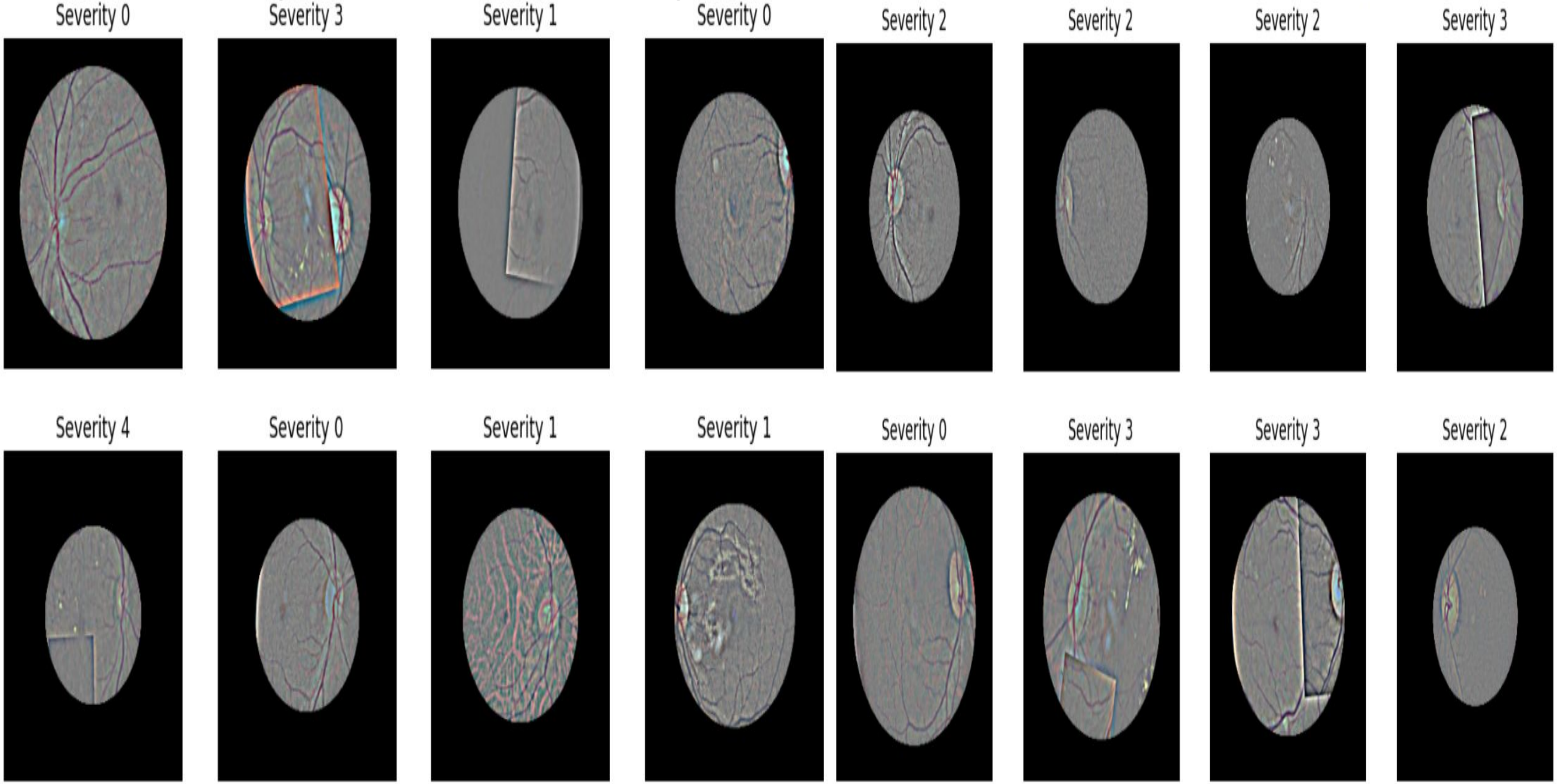
Sample validated images:



Sample Validated Train Images



Cut mix + Ben graham’s preprocessing:



Model training

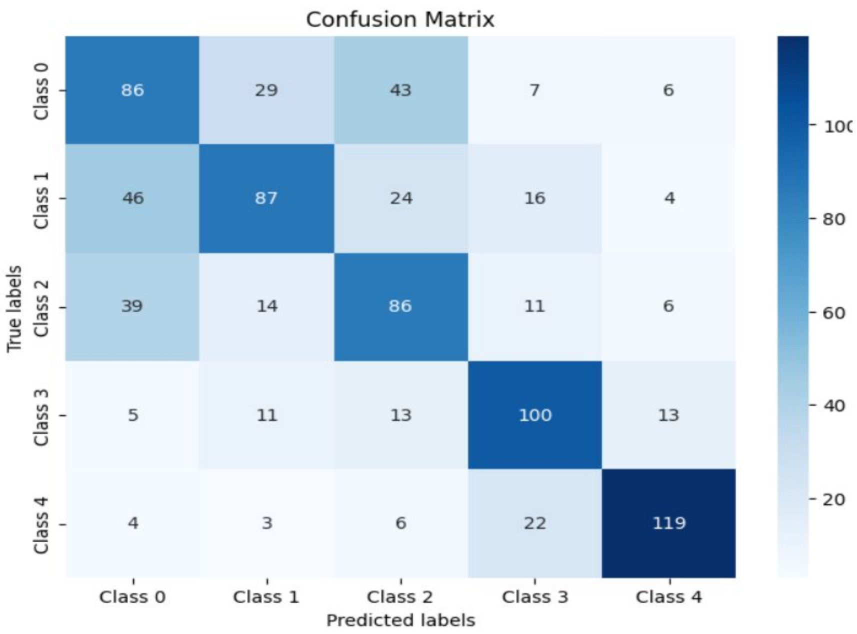
Model trained with ReduceLRonPlateau and early stopping call backs.

Learning rate starting point: 0.001

Model	Loss Function	Epochs
InceptionV3	Categorical cross entropy	150
ResNet152	Kappa loss + categorical cross entropy	150
Efficientnet-B7 + RA block	Categorical cross entropy	180
DenseNet169	Categorical cross entropy	150

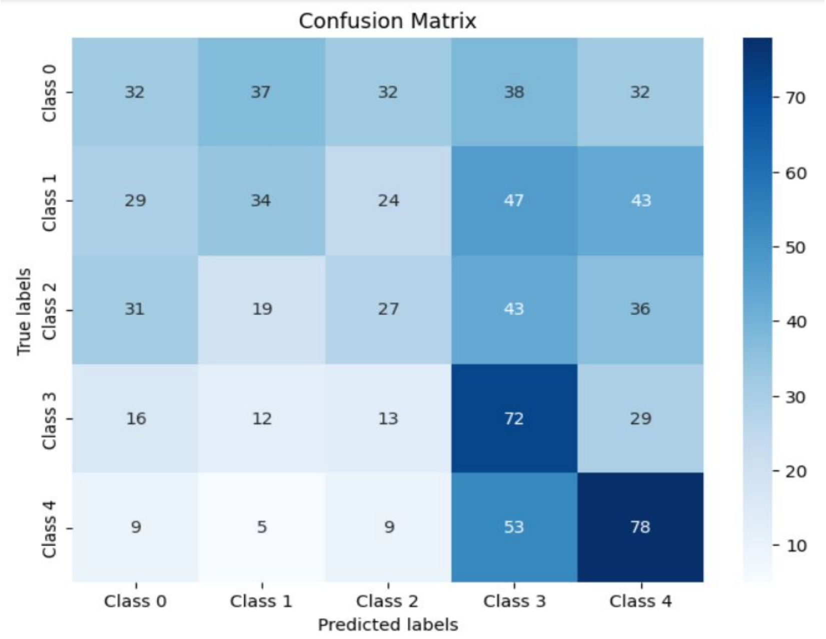
InceptionV3 score

Validation accuracy	Validation Loss	Precision	Recall	F1 score	Accuracy
60%	1.51	0.61	0.60	0.60	0.60



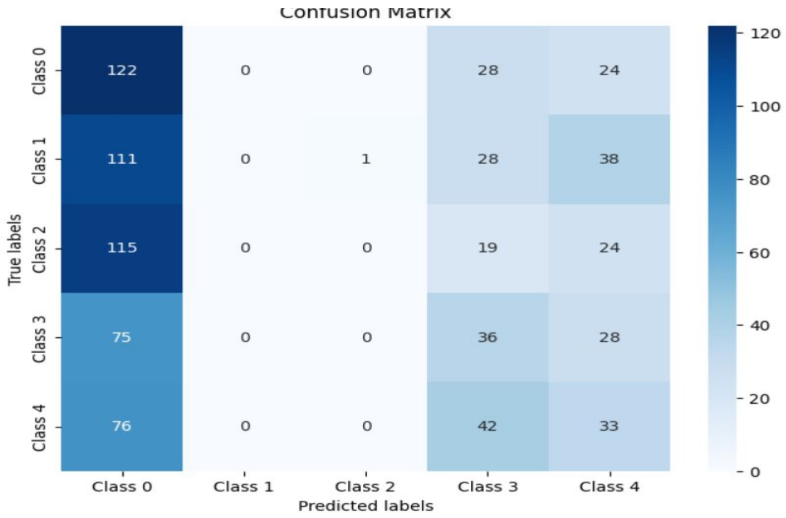
ResNet152 with custom loss (kappa + Categorical cross entropy) score

Validation accuracy	Validation Loss	Precision	Recall	F1 score	Accuracy
32%	1.68	0.30	0.32	0.31	0.32



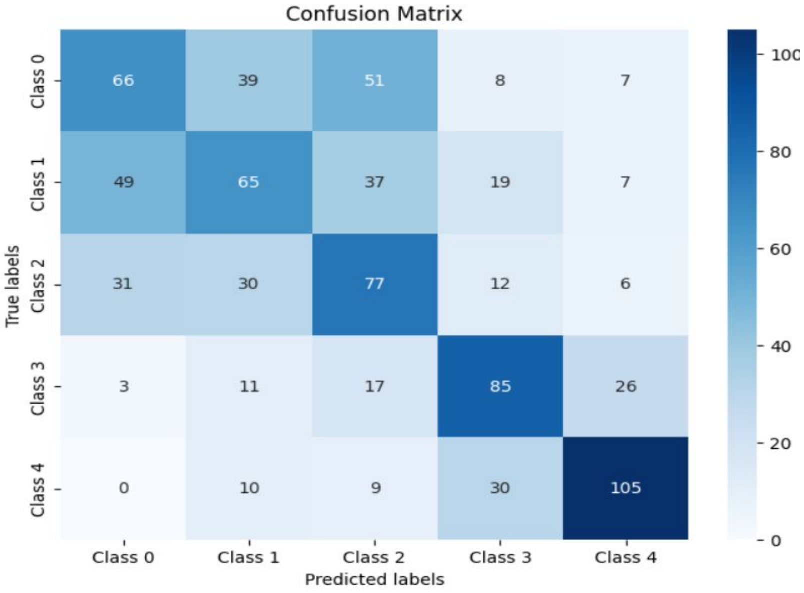
EfficientNet-B7 + RA block score

Validation accuracy	Validation Loss	Precision	Recall	F1 score	Accuracy
25%	3.77	0.20	0.24	0.23	0.24



DenseNet169 score

Validation accuracy	Validation Loss	Precision	Recall	F1 score	Accuracy
54%	1.62	0.52	0.51	0.54	0.54

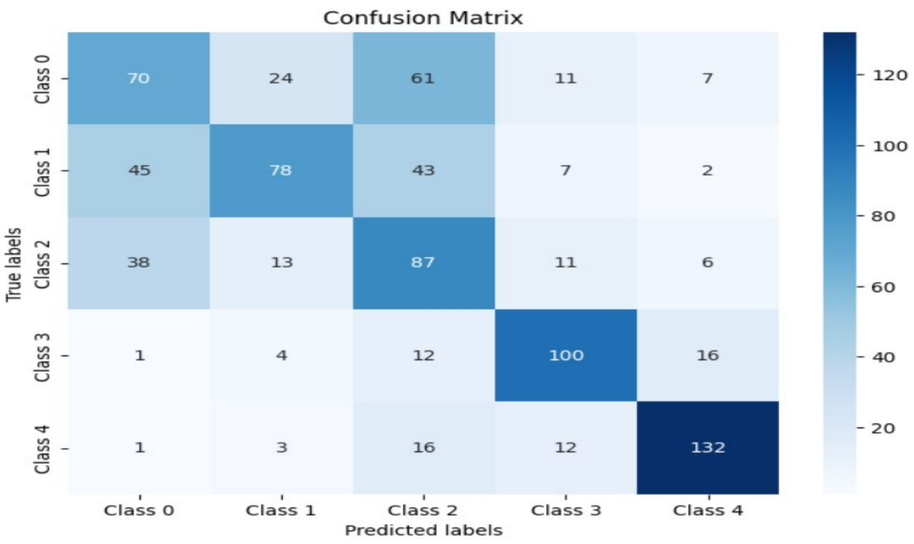


Ensemble Model

Stacking + weightage to best models

Weightage: InceptionV3 highest with 0.4 and densenet with 0.3, resnet with 0.2 and efficientnet with 0.1

Validation accuracy	Validation Loss	Precision	Recall	F1 score	Accuracy
64%	1.62	0.62	0.64	0.65	0.65



Conclusion:

1.This research project introduced an innovative ensemble learning approach for diabetic retinopathy (DR) screening, leveraging multiple machine learning models to enhance accuracy and reliability in disease classification.

2.Through the utilization of deep learning techniques, particularly convolutional neural networks (CNNs), implementing computer vision to automate the detection process. This will reduces the reliance on highly trained experts and helps to efficient management of diabetic retinopathy cases.

3.Future Directions includes to further refine and optimize the proposed techniques, particularly in handling class imbalances and improving the efficiency of detecting proliferative data collection.

References:

1.https://ieeexplore.ieee.org/document/9091167

2..https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8869883

3. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9125868

4. https://ieeexplore.ieee.org/document/8055572

5. https://ieeexplore.ieee.org/document/9214404

6.https://ieeexplore.ieee.org/document/9729867

Thank-you