Multiclass Classification of Diabetic Retinopathy in Retinal Imaging

Pranav Kanth Anbarasan, Sreeja Vepa, Ahmad El Idrissi Amiri

Abstract

Diabetic retinopathy (DR), a major complication of diabetes that can lead to vision loss, necessitates accurate and early detection. In our study, we explored the effectiveness of three distinct convolutional neural network architectures— a baseline CNN, VGG-16, and DenseNet—on a dataset of approximately 3,600 retinal images for classifying different stages of DR. Each architecture was evaluated for its accuracy, sensitivity, and specificity in detecting the stages of DR from no DR to proliferative DR. The DenseNet model demonstrated the highest accuracy at 82%, followed by VGG-16 with 76%, and the baseline CNN at 73%. This comparative analysis highlights each architecture's capabilities in processing complex image data, offering insights for enhancing diagnostic accuracy in medical imaging. The collected findings may contribute to enhancing DR screening technologies, improving early detection and treatment outcomes for diabetic retinopathy patients.

Introduction

Diabetic retinopathy is a diabetes complication which affects the eyes. Over time, the high blood sugar levels related to diabetes can lead to damage of the blood vessels in the retina. This damage can lead to various visual symptoms like vision loss or blindness. To manage this condition, retinal scans are used for effective management of the condition. Healthcare professionals use these scans to look for key features which indicate the presence and progression of the disease. Some of these features include microaneurysms, retinal hemorrhages, and cotton wool spots. Regular monitoring of the disease through these scans increases the likelihood of timely and accurate intervention to reduce the risk of serious damage.

Aside from the mentioned three features, there exist a great number more which also play key roles in determining stage of disease in DR. By using algorithms to classify the severity levels of retinal scans, there can be improvements in diagnostic accuracy, patient outcomes, and optimization of healthcare resources and care. The use of machine learning models such as neural networks offers significant advantages in image classification. Neural networks excel in handling complex patterns.

Neural networks are adept at analyzing the complex details present in various image features. With appropriate training, these networks can reach high degrees of accuracy and precision in determining the severity levels of diabetic retinopathy from retinal images. Capable of processing vast amounts of data quickly, neural networks enable the swift screening of large datasets. This efficient processing not only enhances the allocation of healthcare resources but also significantly benefits patients by providing prompt and precise diagnoses.

Recent studies have shown that neural networks can outperform traditional methods in analyzing retinal images, often achieving accuracies above 90% with large datasets (Tymchenko et al., 2020). In this paper, we explore whether similar results can be achieved with smaller datasets. We focus on three different neural network architectures: a standard convolutional neural network (CNN), a transfer-learning approach using the well-known VGG-16 model, and the advanced DenseNet model. Our work aims to determine if these high-performance diagnostic tools can still operate effectively in situations where data is limited, opening up possibilities for their use in more restricted or data-sensitive environments.

In this paper, we consider the issue of learning from imbalanced datasets, which is common in medical imaging for diabetic retinopathy. In these datasets, severe cases are far less frequent than mild or normal cases, leading to models that are often biased toward the majority class. To address this problem, we investigate several methods aimed at improving the representation of the minority class. These include adjusting class weights during training to give more importance to these crucial but less frequent cases, and the use of resampling methods to potentially balance the dataset. These approaches help ensure that our models perform well across all disease severity levels. This study demonstrates that even with smaller, imbalanced datasets, high levels of diagnostic accuracy can be achieved through careful application of machine learning techniques.

Background

In our project, we investigated the performance of three CNN architectures: a baseline CNN, VGG-16, and DenseNet, for classifying different stages of DR using retinal images. Each architecture was evaluated based on its accuracy

in distinguishing between five different DR classes, based on severity.

Neural Network Architecture

Neural network architecture is a fundamental framework in deep learning, composed of interconnected layers of nodes or neurons that process input data to produce meaningful output. Convolutional Neural Network (CNNs) represent a specialized form of neural networks, designed primarily for tasks involving image data due to their ability to capture spatial hierarchies of features.

In CNNs, each layer performs specific operations to extract and learn intricate patterns from images. The core components include convolutional layers, pooling layers, and fully connected layers.

- **Convolutional Layers:** These layers apply convolutional operations to the input data, effectively detecting features such as edges, textures, and shapes. The convolution operation involves sliding a filter (also known as kernel) over the input image and computing the dot product between the filter and local regions of the image. Mathematically, the convolution operation can be represented as:

$$(I \cdot K)(i, j) = \Sigma_m \Sigma_n \cdot I(m,n) \cdot K(i - m, j - n)$$

where I represent the input image, K is the filter, and (i, j) denotes the spatial position.

- **Pooling Layers:** Pooling layers reduce the spatial dimensions of the feature maps generated by convolutional layers, helping to preserve essential information while reducing computational complexity. Max pooling, a common pooling technique, retains the maximum value within each local region of the feature map. The pooling operation can be defined as:

$$\text{MaxPooling}(x, y) = \max_{i,j} I(x + i, y + j)$$

where I represent the input feature map, and (x, y) denotes the position of the pooling window.

- **Fully Connected Layers:** Fully connected layers integrate the high-level features extracted by convolutional and pooling layers for classification or regression tasks. These layers connect every neuron in one layer to every neuron in the next layer, enabling the network to learn complex relationships between features and target labels.

Transfer Learning

Transfer learning is a machine learning technique that leverages knowledge gained from solving one problem to address a related but different problem. In the context of deep learning, transfer learning involves using pre-trained models trained on large datasets, such as ImageNet, as a starting point for new tasks. By fine tuning the parameters of those pre-trained models on a smaller dataset specific to the new task, transfer learning allows for faster convergence and improved performance, especially when the dataset is limited. Transfer learning enables researchers to capitalize on the features learned by models trained on diverse datasets and apply them to medical image classification tasks, thereby enhancing the accuracy and reducing the need for extensive data annotation and computational resources.

ADAM Optimization

Optimization techniques play a crucial role in training deep learning models, ensuring that they converge efficiently and produce accurate results. One commonly used optimization technique is the Adam optimizer, which stands for Adaptive Moment Estimation. Adam combines ideas from both momentum optimization and RMSprop to achieve good performance across a wide range of deep learning tasks. It maintains per-parameter learning rates that are adapted based on the first and second moments of the gradients. This adaptive learning rate allows Adam to converge quickly to the optimal solution while also being robust to noisy gradients and variations in the scale of the parameters. Additionally, Adam incorporates bias correction to ensure that the estimates of the first and second moments are unbiased, particularly during the early stages of training when the estimates may be inaccurate. Overall, Adam is a popular choice for optimizing deep neural networks due to its efficiency, simplicity, and ability to handle various types of data and architectures.

Related Work

In the realm of diabetic retinopathy classification, researchers often encounter various challenges during dataset preprocessing. Issues such as dark color illumination, color inversions, uninformative black pixels, and irregular cropping patterns are common hurdles that require meticulous handling. However, in our specific case, the dataset exhibited uniform characteristics, with images consistently sized, free from irregular cropping, and preprocessed with Gaussian blur. Consequently, additional preprocessing steps were deemed unnecessary, streamlining our workflow and focusing efforts on other aspects of model development.

When it comes to transforming output variables, conventional techniques like one-hot encoding and ordinal regression are commonly employed. Despite their utility, we opted against their use in our project. This decision stemmed from the relatively modest size of our dataset, which posed a risk of overfitting with such transformation methods.

In the related work for DR image classification, the weighted kappa score was provided as an evaluation metric (Tymchenko et al., 2020). This metric measures the degree of agreement between raters when dealing with categorical data. Unlike simple accuracy, which calculates the

percentage of true predictions over total predictions, the kappa score offers a more robust assessment by considering the agreement that could occur by chance. The quadratic weighted Cohen's kappa score specifically allows for disagreements to be weighted differently, which is particularly useful when dealing with ordered scores. However, in our approach, we opted not to use Cohen's Kappa as an evaluation metric. Instead, we focused on traditional classification metrics such as accuracy, F1 score, precision, and recall. These metrics provided comprehensive insights into our classification model's performance, aligning more closely with our specific goals for the task at hand.

While data augmentation is a widely adopted technique in image classification tasks, especially when dealing with limited datasets, our approach diverged from this convention. Prior studies have demonstrated the effectiveness of data augmentation in expanding datasets and mitigating overfitting by introducing variations in the image samples through transformations like flipping, rotation, and scaling (Bosch et al., 2007). However, despite its potential benefits, data augmentation comes with computational costs and requires careful parameter tuning to ensure that augmented samples remain representative of the original data distribution.

We chose not to employ data augmentation due to the unique characteristics of our dataset, including uniform image sizes, consistent illumination, and minimal noise. In such cases, data augmentation may not be beneficial. Instead, we prioritized optimizing model architectures, fine tuning hyperparameters, and leveraging transfer learning to achieve robust performance in diabetic retinopathy classification while minimizing computational overhead.

Ensemble methods, such as taking the majority vote of multiple model predictions have been implemented in several related studies and can be effective for improving classification approach, we opted not to implement this approach in our project. While ensemble techniques can offer further performance gains, our decision was influenced by resource constraints and the complexity involved in integrating multiple models. Thus, we prioritized the simplicity and effectiveness of transfer learning to meet our classification objectives.

Project Description

The formulation of our image classification question is as follows, given a retinal image I, and its features Xi, use Xi features to predict the DR stage Y, of the image I.
Traditional methods such as manual classification by specialists are often time-consuming and costly. Additionally, other classic techniques, reliant on basic feature extraction, may lack comprehensive data interpretation. To address both issues and improve in both accuracy and efficiency, we utilized advanced convolutional neural network architectures (CNN) which can capture and analyze the intricate features in retinal images.

Dataset

This paper utilizes the Diabetic Retinopathy 224x224 Gaussian Filtered dataset. This dataset consists of a folder of gaussian filtered retinal scans sorted into five directories. The directories each represent a set of images corresponding to one of five severity stages of DR. The corresponding folders show the five levels are:

- 0 – No_DR
- 1 – Mild
- 2 – Moderate
- 3 – Severe
- 4 – Proliferate_DR

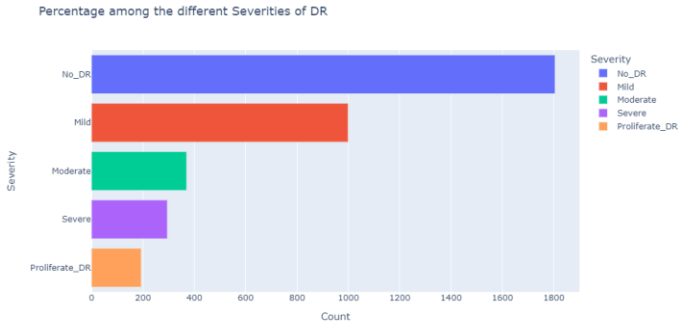

Figure 1: Bar plot indicating class instance counts of dataset

The dataset is highly unbalanced, most samples were in the control class of No_DR.

Statistical Models

Explored three neural architectures in this paper: a baseline convolutional neural network, transfer-learning model with VGG-16, and a transfer learning model with DenseNet.

Each model used an 80:20 split between Train and Test sets. Also, each used categorical cross entropy as the loss function and ADAM as an optimizer.

Categorical cross entropy was used as the primary loss function in all the models due to its primary use in multi-class classification problems. This loss function measures how well the predicted probability distribution matches the actual distribution of the labels. The equation for this:

$$Loss = -\Sigma_{c=1}^{C} y_c \log(p_c)$$

C represents the number of classes, y_c is the true label for class c, and p_c is the predicted probability of class c.

Additionally, the Adaptive Moment Estimation (ADAM) optimizer is particularly effective for tasks like retinal classification of diabetic retinopathy because it dynamically adjusts learning rates based on the data it processes. This feature is essential for dealing with the subtle and complex

features typical of diabetic retinopathy, ensuring robust learning from the imbalanced dataset. By speeding up the learning process and improving model accuracy, ADAM is well-suited for the necessary intricate analyses needed for this dataset. All models used this optimizer for consistency purposes as well.

Baseline CNN

Image preprocessing included reading image files from a disk and then decoding them into grayscale formatting. All images were resized to a uniform size of 224 x 224 pixels and the values were normalized to a range [0, 1].

In the context of retinal scans, it is expected that the first convolutional layer will detect edges of blood vessels, and boundaries of figures in the image. It may also detect basic textures. The following layers focus on more complex features such as clusters of lesions, smaller textural changes, and specific patterns of blood vessels. The deepest layers, the fully connected layers, integrate all previously detected features to make diagnostic classifications as the final output.

The input shape of the model was of size [224, 224, 1] corresponding to resized grayscale images. There were three convolutional layers with 32, 64, and 128 filters respectively, each with a kernel size of (3,3) and ReLU activation. Each convolutional layer is followed by a max pooling layer with a pool size (2,2). Max pooling was chosen due to its effectiveness at highlighting the strongest signals in the feature map as detection of strong features such as edges is crucial in medical image analysis. Pooling reduces the dimensions of the output from the convolutional layer and enhances the detection of features.

A flatten layer transforms the 3D feature maps into 1D vectors. There were two dense layers in this mode, the first having 128 units and ReLU activation, and the second having 5 units. This is because there are five unique diagnosis values possible. ReLU was incorporated into this model because it introduces non-linearity, enabling the network to capture complex patterns effectively. This second layer uses the softmax activation function to output probabilities for each class

To handle class imbalance, class weights were computed to adjust the importance given to each class during training. The equation used to compute the weights is:

$$w_j = n / (k \times n_j)$$

- w_j is the weight for class j
- n is the total number of samples
- k is the total number of classes
- n_j is the number of samples in class j

Transfer Learning: VGG-16 and DenseNet121

In this project, the utilization of transfer learning specifically leveraging the VGG-16 and DenseNet121 architectures, proved imperative due to the limited availability of data and the pronounced class imbalance within the diabetic retinopathy (DR) image dataset. To mitigate these challenges, the dataset underwent under-sampling to ensure uniform representation across all classes. This process involved selecting a subset of 400 images for each class. In cases where a class contained fewer than 400 samples, additional resampling was performed. Specifically, existing images within those classes were duplicated until the target of 400 samples per class was achieved. This strategy fostered a balanced dataset, effective for model training by addressing the

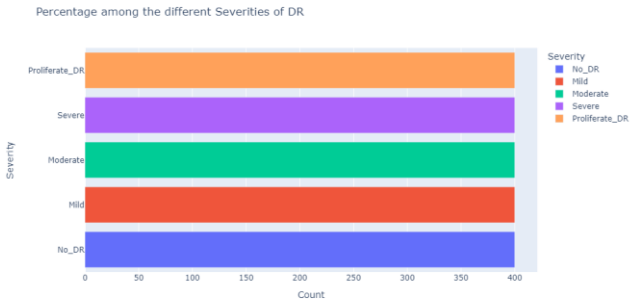

Figure 2: Bar plot of dataset after under sampling

issue of class imbalance.

Subsequently, a preprocessing step was executed to convert all images into RGB pixel values and normalize them. This step was crucial for standardizing the input data and facilitating faster convergence of the optimization algorithm during model training.

Both the architectures are initialized with weights pretrained on ImageNet and then customized to suit our specific problem. For both architectures, the top layers are removed, and custom layers are added to adapt the model to our DR classification task. In the custom layer, a Dense layer with 256 neurons and ReLU activation is incorporated to capture higher-level features from the extracted features of the pretrained network. Additionally, a dropout layer with a dropout rate of 0.4 is included to prevent overfitting by randomly dropping out 40% of the neurons during training.

The output layer consists of a Dense layer with softmax activation, which outputs the probability distribution over the five classes of DR severity. The models are compiled using the Adam optimizer with a learning rate of 0.0001 and categorical cross-entropy loss function.

To monitor the model's performance during training, an early stopping callback is implemented, which halts training if there is no improvement in the testing accuracy for ten consecutive epochs. This helps prevent overfitting and ensures that the models generalize well to unseen data.

Finally, both models are trained for 70 epochs with a batch size of 8, utilizing both training and testing datasets. Thus, ensuring that the models learn to classify DR images accurately while minimizing the risk of overfitting.

Experiments

To properly analyze a highly imbalanced dataset, metrics such as the number of misclassifications, precision, and recall were accounted for rather than solely accuracy. In cases such as the DR dataset, accuracy could be misleading due to possibly reflecting mostly on the accuracy associated with the majority class rather than all the minority classes. By focusing on precision and recall, we gained a better understanding of how the model was working on predicting minority class outcomes along with the majority class outcomes.

Baseline CNN

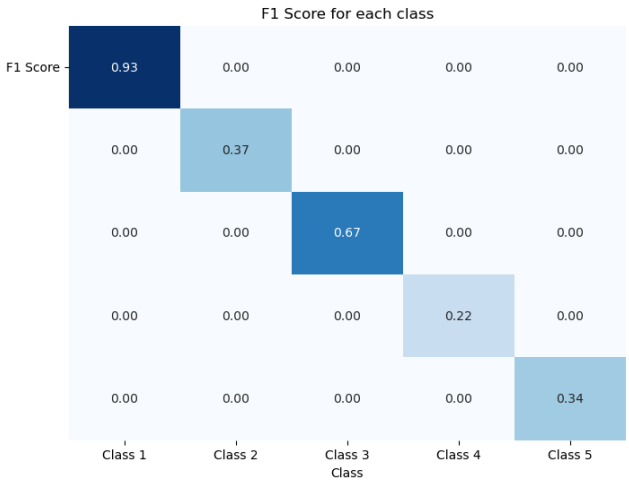

Figure 3: F1 Scores for Baseline CNN

The above diagram illustrates the F1 accuracy of each class. F1 accuracy accounts for the weighted average of precision and recall considering both false positives and negatives. A high score of 0.93 for class 1 indicates the model is effectively balancing both precision and recall, thus indicating the model is reliable for predicting class 1. The F1 score for class 3 is 0.67 as this is a moderate score, the model is shown to have good performance but with room for improvement. This can be further explored by looking into the individual precision and recall scores for the class. Classes 2, 4, and 5 all have a low F1 score showing poor performance. These scores are indicative of high misclassification rates for these classes. This could be due to there being fewer samples for the model to train on for these classes.

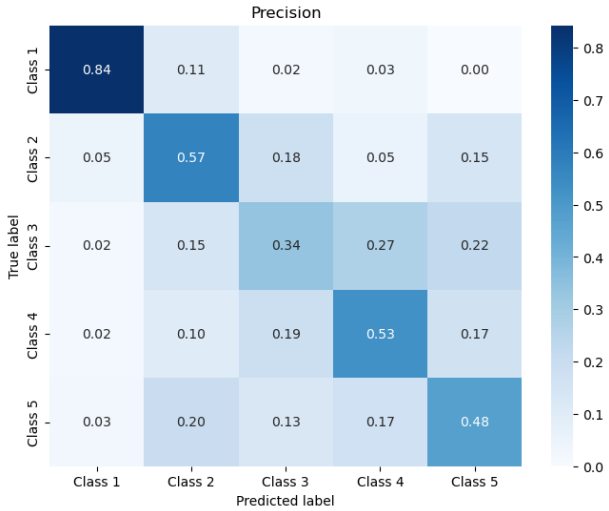

Figure 4: Precision for Baseline CNN

The precision scores from the matrix show that Class 1 has a high precision of 0.84, indicating strong predictive accuracy with few false positives. Conversely, Class 3's precision is notably low at 0.34, which contributes to its moderate F1 score of 0.67, highlighting issues with false positives. Precision for the remaining classes is moderate.

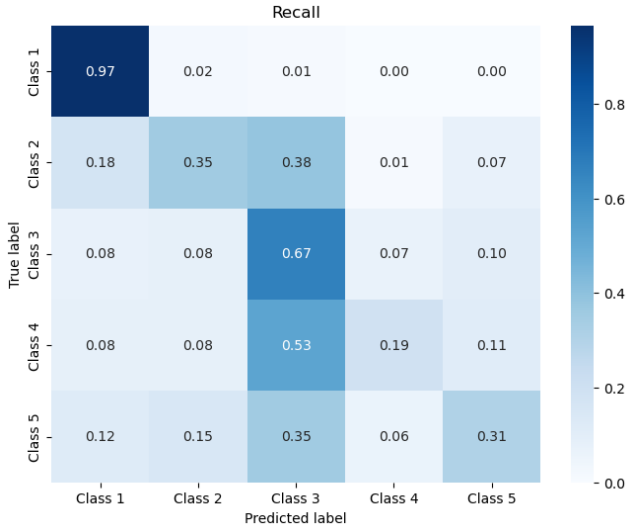

Figure 5: Recall for Baseline CNN

The figure above shows class 1 to have a high recall, contributing to a high F1 score. Class 2, 4, and 5 exhibit low recall scores of 0.35, 0.19, and 0.31, respectively. Class 3 has a moderate recall score of 0.67, suggesting that while the model is reasonably effective at identifying instances of this class, there is room for improvement in capturing most of the cases accurately.

Transfer Learning

In this experiment, DenseNet121 and VGG16 were evaluated for multiclass classification on a balanced dataset, with accuracy as the primary metric. DenseNet121 achieved 93.88% accuracy during training and 82% during testing, while VGG16 attained 90.25% and 75.74% respectively. The slightly higher testing accuracy of DenseNet121 implies better generalization capability compared to VGG16, likely due to its dense connectivity pattern, which promotes feature reuse and facilitates gradient flow, leading to more efficient learning and better representation of complex patters.

These results highlight the effectiveness of both architectures even with a relatively small amount of data, demonstrating their potential for robust multiclass classification tasks. The notable accuracies achieved underscore the viability of DenseNet121 and VGG16 in real-world scenarios where limited data availability is common.

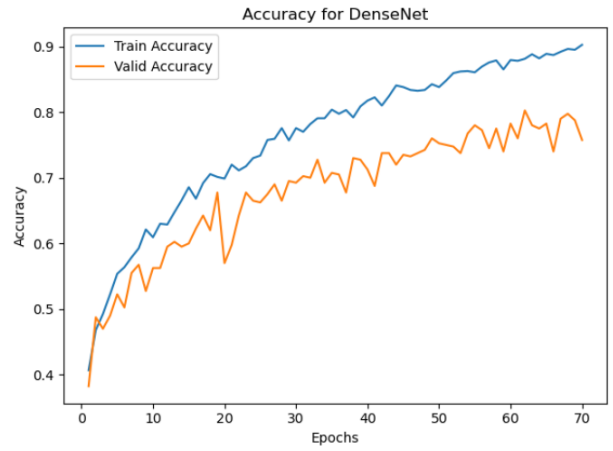

Figure 6: Accuracy for DenseNet Architecture

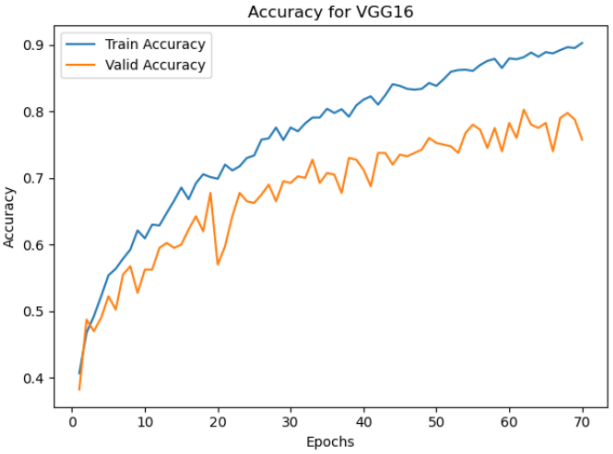

Figure 7: Accuracy for VGG16 Architecture

Conclusion

This research critically evaluated various metrics to assess model performance on the highly imbalanced Diabetic Retinopathy (DR) dataset. Our findings reveal the limitations of traditional accuracy metrics, which often fail to provide a true representation of model effectiveness due to their overemphasis on the majority class. The baseline CNN demonstrated strong performance for Class 1, achieving an F1 score of 0.93, yet it faced challenges with Classes 2, 4, and 5. This underscore substantial disparities in model performance across different disease severities.

Our study highlighted the importance of precision and recall as essential metrics for gaining a more nuanced understanding of the model's capabilities, particularly in accurately predicting outcomes for minority classes. Additionally, employing transfer learning with advanced architectures like DenseNet121 and VGG16 showcased their robust capabilities, with DenseNet121 notably achieving superior performance in both training and testing phases compared to VGG16.

Despite achieving above-human-level accuracy through the integration of multiple modalities within our network, our efforts to fine-tune the models were somewhat constrained.

In future research, we aim to extend fine-tuning beyond the final layers and consider the incorporation of data augmentation techniques. In the related work section, we mentioned the choice to not employ data augmentation in this paper. However, in future research, we would like to compare our results with and without data augmentation. The completed transformations could include rotations, zooming, and flipping. These transformations have the potential to help the model generalize better for underrepresented classes.

By optimizing and refining these three models, we hope to address the pressing need for precise medical diagnostics, where models with accuracy can significantly impact patient care and treatment efficacy.

References

A. Bosch, A. Zisserman and X. Munoz, "Image Classification using Random Forests and Ferns," 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007, pp. 1-8, doi: 10.1109/ICCV.2007.4409066.

D. Cire¸san, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. Arxiv preprint arXiv:1202.2745, 2012.

Garg A (2022a) Image classification using resnet-50 Deep learning model. In: Analytics Vidhya.

Huh, M., Agrawal, P., & Efros, A. A. (2016). What makes ImageNet good for transfer learning? arXiv preprint

arXiv:1608.08614. Retrieved from https://arxiv.org/abs/1608.08614

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks.
L. Breiman. Random forests. Machine learning, 45(1):5–32, 2001.

Tammina, Srikanth. (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. International Journal of Scientific and Research Publications (IJSRP). 9. p9420. 10.29322/IJSRP.9.10. 2019.p9420.

Tymchenko, B., Marchenko, P., & Spodarets, D. (2020). Deep learning approach to diabetic retinopathy detection. arXiv preprint arXiv:2003.02261. Retrieved Rfrom https://arxiv.org/abs/2003.02261

Link to the code:

https://github.com/pranavneu/diabetic_retinopathy