TABLE OF CONTENTS

TOPIC : DEEP LEARNING FUNDUS IMAGE ANALYSIS FOR EARLY DETECTION OF DIABETIC RETINOPATHY

1. INTRODUCTION:

1.1 PROJECT OVERVIEW:

Diabetic Retinopathy is a retina disease caused by diabetes mellitus and it is the leading cause of blindness globally. Early detection and treatment are necessary in order to delay or avoid vision deterioration and vision loss. To that end, many artificial-intelligence-powered methods have been proposed by the research community for the detection and classification of diabetic retinopathy on fundus retina images. This review article provides a thorough analysis of the use of deep learning methods at the various steps of the diabetic retinopathy detection pipeline based on fundus images. We discuss several aspects of that pipeline, ranging from the datasets that are widely used by the research community, the preprocessing techniques employed and how these accelerate and improve the models' performance, to the development of such deep learning models for the diagnosis and grading of the disease as well as the localization of the disease's lesions. We also discuss certain models that have been applied in real clinical settings. Finally, we conclude with some important insights and provide future research directions. Diabetic retinopathy is a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina).

At first, diabetic retinopathy might cause no symptoms or only mild vision problems. But it can lead to blindness.

The condition can develop in anyone who has type 1 or type 2 diabetes. The longer you have diabetes and the less controlled your blood sugar is, the more likely you are to develop this eye complication.

1.2 PURPOSE

Diabetic eye screening is important as it helps to prevent sight loss. As someone with diabetes, your eyes are at risk of damage from diabetic retinopathy. Screening can detect the condition early before you notice any changes to your vision. Current diabetic retinopathy screening guidelines recommend a retinal examination in type 1 diabetics 5 years after diagnosis and at least annually thereafter. Type 2 diabetes patients should be examined immediately at the time of diagnosis and at least annually thereafter. Diabetic retinopathy is best diagnosed with a comprehensive dilated eye exam. For this exam, drops placed in your eyes widen (dilate) your pupils to allow your doctor a better view inside your eyes. The drops can cause your close vision to blur until they wear off, several hours later.This labor-intensive task could greatly benefit from automatic detection using deep learning technique. Here we present a deep learning system that identifies referable diabetic retinopathy comparably or better than presented in the previous studies, although we use only a small fraction of images (<1/4) in training but are aided with higher image resolutions. We also provide novel results for five different screening and clinical grading systems for diabetic retinopathy and macular edema classification, including state-of-the-art results for accurately classifying images according to clinical five-grade diabetic retinopathy and for the first time for the four-grade diabetic macular edema scales. These results suggest, that a deep learning system could increase the cost-effectiveness of screening and diagnosis, while attaining higher than recommended performance, and that the system could be applied in clinical examinations requiring finer grading. Here we present a deep learning system that identifies referable diabetic retinopathy comparably or better than presented in the previous. In this study, we present a diabetic retinopathy detection system based on ultra-wide-field fundus photography and deep learning. This project is a part of the whole process of identifying Diabetic Retinopathy in its early stages. In this project, we'll extract basic features which can help us in identifying Diabetic Retinopathy in its early stages.

2. LITERATURE SURVEY

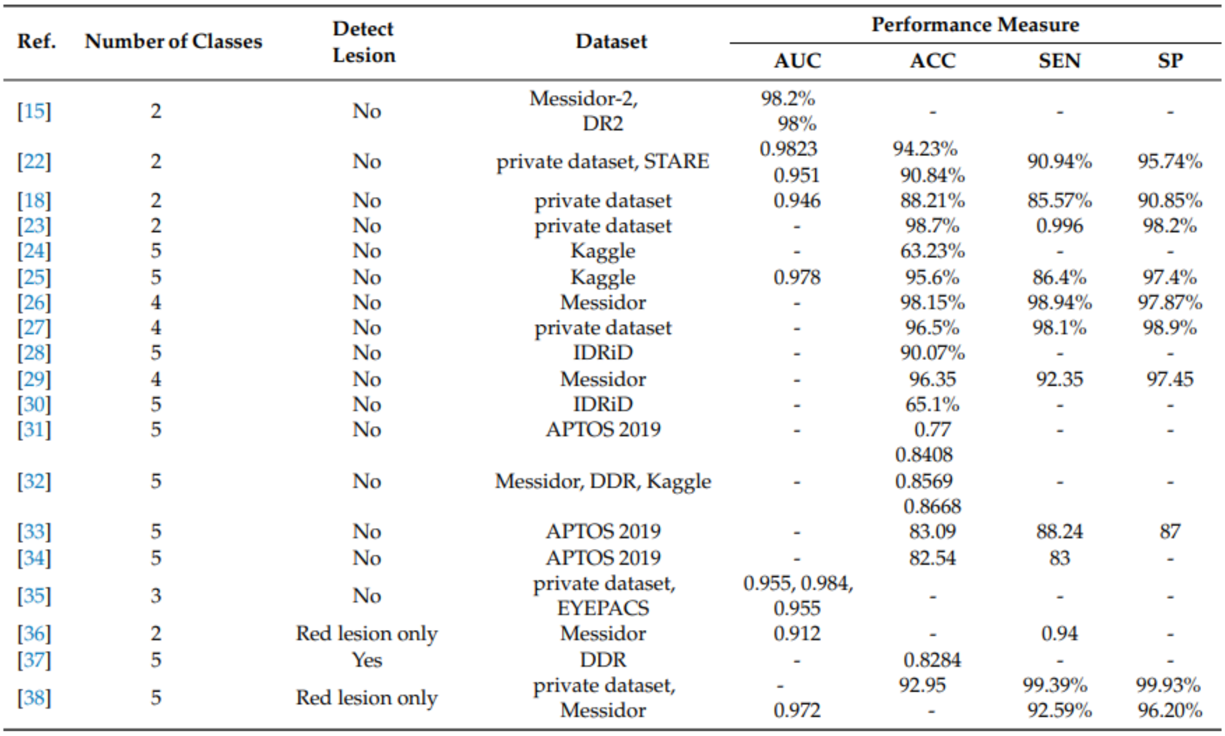
2.1 EXISTING PROBLEM:

CNN has been used widely in the classification and localisation of retinal fundus images. The DR detection works using DL can be categorized into three main categories: binary DR classification, multi-level DR classification and hybrid classification. In the following sections, we will summarise the recent efforts in DR classification in these three categories. A comparison between the related works is presented in Table 2.

2.1.1. Binary Classification

This section looks at the studies that have classified DR images into two categories. Pires et al. proposed a custom CNN to detect referable DR images and non-referable DR images. Their CNN were trained on the Kaggle and achieved an AUC of 98.2% on the Messidor-2. Jiang et al. created a new dataset to classify DR images to referable DR or not using three pretrained CNNs; Inception-Resnet-V2, Inception V and Resnet152 . These CNNs were combined using the Adaboost algorithm. They obtained an AUC of 0.946. Liu et al. created a weighted paths CNN called WP-CNN to classify referable DR images in a private dataset. They reported an accuracy (ACC) of 94.23%. Das et al. proposed two independent CNN to classify the images into normal or DR images. Their CNNs obtained an ACC of 98.7% on the DIARETDB1 dataset. Although the previous studies achieved good results in detecting DR, they did not take the five DR stages and the various lesions into account. The main drawback of the binary classification method is that it only classifies the DR images into two categories, without considering the five DR stages. The identification of the exact DR stages is essential in selecting a suitable treatment process and preventing retina deterioration.

Table 2. Comparison between the related works that used DL to classify DR Images.

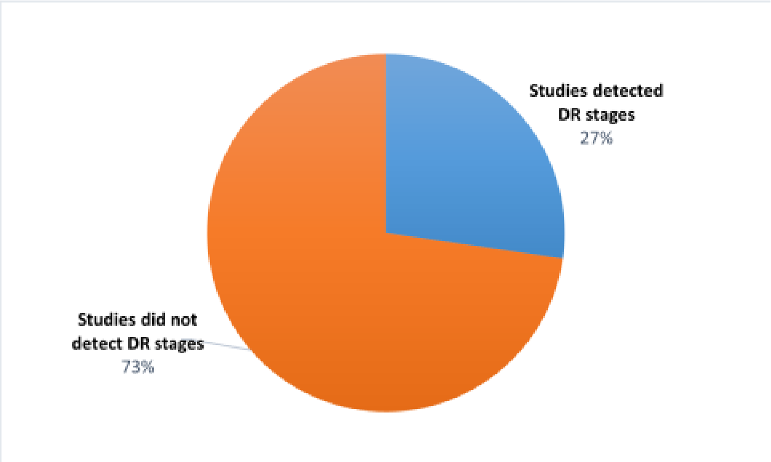


Ref.	Number of Classes	Detect Lesion	Dataset	Performance Measure			
				AUC	ACC	SEN	SP
[15]	2	No	Messidor-2, DR2	98.2% 98%	-	-	-
[22]	2	No	private dataset, STARE	0.9823 0.951	94.23% 90.84%	90.94%	95.74%
[18]	2	No	private dataset	0.946	88.21%	85.57%	90.85%
[23]	2	No	private dataset	-	98.7%	0.996	98.2%
[24]	5	No	Kaggle	-	63.23%	-	-
[25]	5	No	Kaggle	0.978	95.6%	86.4%	97.4%
[26]	4	No	Messidor	-	98.15%	98.94%	97.87%
[27]	4	No	private dataset	-	96.5%	98.1%	98.9%
[28]	5	No	IDRiD	-	90.07%	-	-
[29]	4	No	Messidor	-	96.35	92.35	97.45
[30]	5	No	IDRiD	-	65.1%	-	-
[31]	5	No	APTOS 2019	-	0.77	-	-
[32]	5	No	Messidor, DDR, Kaggle	-	0.8408 0.8569 0.8668	-	-
[33]	5	No	APTOS 2019	-	83.09	88.24	87
[34]	5	No	APTOS 2019	-	82.54	83	-
[35]	3	No	private dataset, EYEPACS	0.955, 0.984, 0.955	-	-	-
[36]	2	Red lesion only	Messidor	0.912	-	0.94	-
[37]	5	Yes	DDR	-	0.8284	-	-
[38]	5	Red lesion only	private dataset, Messidor	- 0.972	92.95 -	99.39% 92.59%	99.93% 96.20%

2.1.2. Multi-Level Classification

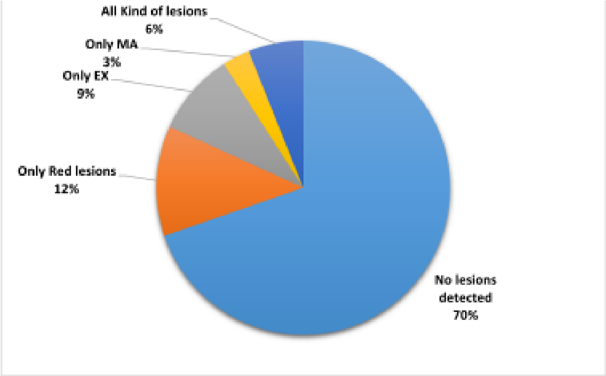
This section reviews the works that have classified DR images into various stages. Wang et al. examined the performance of three pre-trained CNNs in the Kaggle dataset to classify all the stages of the DR images. The three CNN architectures used were InceptionNet V3 , AlexNet and VGG16 . The best average ACC of 63.23% was obtained by InceptionNet V3. The work of transferred learning pre-trained AlexNet, VggNet , GoogleNet and ResNet to detect the different DR stages in the Kaggle dataset. Their results showed that VggNet achieved the higher ACC, with a value of 95.68%. Mobeen-ur-Rehman et al. proposed a simple CNN to detect the DR stages of the Messidor dataset. Their CNN obtained an excellent ACC of 98.15%. Zhang et al. proposed a method to detect the DR stages of their private dataset. They fine-tuned InceptionV3 , ResNet50 , Xception , InceptionResNetV2 , and DenseNets and then combined the strongest CNNs. This method obtained an ACC of 96.5%. Harangi et al. classified the DR stages by integrating hand-crafted features and AlexNet . They used the Kaggle dataset for training and the IDRiD dataset for testing. This method achieved an ACC of 90.07%. Shanthi and Sabeenian used Alexnet to classify the DR stages of the Messidor dataset . Their ACC was 96.35%. Li et al. used ResNet50 with attention modules to classify the stages in the IDRiD dataset, resulting in a 65.1% joint ACC. Dekhil et al. transferred learning.

VGG16 to classify the DR stages in the APTOS 2019 Kaggle dataset , and they achieved an ACC of 77%. He et al. proposed a CABNet network to classify DR images into stages, achieving an ACC of 85.69% in the DDR . Kassani et al. modified Xception model to classify the DR stages in the APTOS 2019 Kaggle dataset , resulting in a 83.09% ACC. Bodapati et al. proposed a composite network with gated attention to classify DR images into stages, achieving an ACC of 82.54% in the APTOS 2019 Kaggle dataset . Hsieh et al. trained the modified Inception-v4 and the modified ResNet to detect any DR, proliferative DR and referable DR in their private dataset and the EYEPACS dataset. They obtained an AUC of 0.955, 0.984 and 0.955, respectively in detecting any DR, proliferative DR and referable DR. These previous studies demonstrated that CNN is effective in classifying DR images. However, localising DR lesions with DR image classification is more efficient for ophthalmologists at diagnosis. Moreover, Alyoubi et al. reported that most of the studies, almost 70%, classified the fundus images using binary classifiers such as DR or non-DR, while only 27% classified the input to one or more stages, as shown in Figure 3.



2.1.3. Hybrid Classification

This section presents the studies that classified DR images and localised lesions at the same time. Zago et al. used VGG16 to detect red lesion patches of the DR images, and then they classified the image to DR or no-DR based on the detected red lesions. Their best results were achieved in the Messidor dataset with an AUC of 0.912. Li et al. created a dataset called the DDR to classify images into five DR stages and to localise lesions. For the stages classification, they achieved the best ACC of 82.84% using SE-BN-Inception , while for localisation, they achieved a mAP of 9.2 using Faster RCNN . Wang et al. used two modified RFCN to detect the stages of DR and localise the MA and HM. Then the results from the two RFCN were merged. In their private dataset, this method achieved a mAP of 92.15 in localizing, while in classification, they achieved a 92.95% ACC. Many studies, such as those by W. Alyoubi et al. and T. Li et al. , show that the main limitation of the DR classification systems is that only a limited number of the studies detected and localized the types of DR lesions on the fundus image, as shown in Figure 4. Furthermore, there are limited studies that detected the DR stages, grading and lesions together. Lesions localization with high accuracy helps with grading the cases and the patients' follow-up, which is considered a critical requirement for DR patients.



2.2 REFERENCES

1. American Academy of Ophthalmology-What Is Diabetic Retinopathy. Available online: https://www.aao.org/eye-health/

diseases/what-is-diabetic-retinopathy (accessed on 1 January 2019).

2. Bourne, R.R.; Stevens, G.A.; White, R.A.; Smith, J.L.; Flaxman, S.R.; Price, H.; Jonas, J.B.; Keeffe, J.; Leasher, J.; Naidoo, K.; et al.

Causes of vision loss worldwide, 1990-2010: A systematic analysis. Lancet Glob. Health 2013, 1, 339–349. [CrossRef]

3. Taylor, R.; Batey, D. Handbook of Retinal Screening in Diabetes:Diagnosis and Management, 2nd ed.; John Wiley & Sons, Ltd.,

Wiley-Blackwell: Hoboken, NJ, USA, 2012; pp. 1–173. [CrossRef]

4. Wilkinson, C.P.; Ferris, F.L.; Klein, R.E.; Lee, P.P.; Agardh, C.D.; Davis, M.; Dills, D.; Kampik, A.; Pararajasegaram, R.; Verdaguer,

J.T.; Lum, F. Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. Am. Acad.

Ophthalmol. 2003, 110, 1677–1682. [CrossRef]

5. Deng, L.; Yu, D. Deep learning: Methods and applications. Found. Trends Signal Process. 2014, 7, 197–387. [CrossRef]

6. Vega, R.; Sanchez-Ante, G.; Falcon-Morales, L.E.; Sossa, H.; Guevara, E. Retinal vessel extraction using lattice neural networks

with dendritic processing. Comput. Biol. Med. 2015, 58, 20–30. [CrossRef] [PubMed]

7. Al Zaid, E.; Shalash, W.M.; Abulkhair, M.F. Retinal blood vessels segmentation using Gabor filters. In Proceedings of the 2018 1st

International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 4–6 April 2018;

pp. 1–6.

8. Sikder, N.; Masud, M.; Bairagi, A.K.; Arif, A.S.M.; Nahid, A.A.; Alhumyani, H.A. Severity Classification of Diabetic Retinopathy

Using an Ensemble Learning Algorithm through Analyzing Retinal Images. Symmetry 2021, 13, 670. [CrossRef]

9. Bakator, M.; Radosav, D. Deep learning and medical diagnosis: A review of literature. Multimodal Technol. Interact. 2018, 2, 47.

[CrossRef]

10. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J.A.; van Ginneken, B.; Sánchez, C.I. A

survey on deep learning in medical image analysis. Med. Image Anal. 2017, 42, 60–88. [CrossRef] [PubMed]

11. Lu, L.; Zheng, Y.; Carneiro, G.; Yang, L. Deep Learning and Convolutional Neural Networks for Medical Image Computing; Springer:

Berlin/Heidelberg, Germany, 2017; [CrossRef]

12. Tan, M.; Le, Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th

International Conference on Machine Learning, ICML 2019, Beach, CA, USA, 10–15 June 2019; pp. 6105–6114.

13. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. arXiv 2018, arXiv:1804.02767.

14. Pal, P.; Kundu, S.; Dhara, A.K. Detection of red lesions in retinal fundus images using YOLO V3. Curr. Indian Eye Res. J.

Ophthalmic Res. Group 2020, 7, 49.

15. Pires, R.; Avila, S.; Wainer, J.; Valle, E.; Abramoff, M.D.; Rocha, A. A data-driven approach to referable diabetic retinopathy

detection. Artif. Intell. Med. 2019, 96, 93–106. [CrossRef]

16. Kaggle 2015 Dataset. Available online: https://kaggle.com/c/diabetic-retinopathy-detection (accessed on 1 April 2019).

17. Decenciere, E.; Zhang, X.; Cazuguel, G.; Lay, B.; Cochener, B.; Trone, C.; Gain, P.; Ordonez, R.; Massin, P.; Erginay, A.; et al.

Feedback on a publicly distributed image database: The messidor database. Image Anal. Stereol. 2014, 33, 231–234. [CrossRef]

18. Jiang, H.; Yang, K.; Gao, M.; Zhang, D.; Ma, H.; Qian, W. An Interpretable Ensemble Deep Learning Model for Diabetic

Retinopathy Disease Classification. In Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in

Medicine and Biology Society (EMBC), Berlin, Germany, 23–27 July 2019; pp. 2045–2048. [CrossRef]

19. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on

learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February

2017; pp. 4278–4284.

20. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016;

pp. 2818–2826. [CrossRef]

Sensors 2021, 21, 3704 21 of 22

21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on

Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. [CrossRef]

22. Liu, Y.P.; Li, Z.; Xu, C.; Li, J.; Liang, R. Referable diabetic retinopathy identification from eye fundus images with weighted path

for convolutional neural network. Artif. Intell. Med. 2019, 99, 101694. [CrossRef] [PubMed]

23. Das, S.; Kharbanda, K.; Suchetha, M.; Raman, R.; Dhas, E. Deep learning architecture based on segmented fundus image features

for classification of diabetic retinopathy. Biomed. Signal Process. Control 2021, 68, 102600. [CrossRef]

24. Wang, X.; Lu, Y.; Wang, Y.; Chen, W.B. Diabetic retinopathy stage classification using convolutional neural networks. In

Proceedings of the International Conference on Information Reuse and Integration for Data Science, Salt Lake City, UT, USA, 6–9

July 2018; pp. 465–471. [CrossRef]

25. Wan, S.; Liang, Y.; Zhang, Y. Deep convolutional neural networks for diabetic retinopathy detection by image classification.

Comput. Electr. Eng. 2018, 72, 274–282. [CrossRef]

26. Mobeen-Ur-Rehman.; Khan, S.H.; Abbas, Z.; Danish Rizvi, S.M. Classification of Diabetic Retinopathy Images Based on

Customised CNN Architecture. In Proceedings of the 2019 Amity International Conference on Artificial Intelligence, AICAI 2019, Dubai, United Arab Emirates, 4–6 February 2019; pp. 244–248. [CrossRef]

27. Zhang, W.; Zhong, J.; Yang, S.; Gao, Z.; Hu, J.; Chen, Y.; Yi, Z. Automated identification and grading system of diabetic retinopathy using deep neural networks. Knowl. Based Syst. 2019, 175, 12–25. [CrossRef]

28. Harangi, B.; Toth, J.; Baran, A.; Hajdu, A. Automatic screening of fundus images using a combination of convolutional neural network and hand-crafted features. In Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 23–27 July 2019; pp. 2699–2702. [CrossRef]

29. Shanthi, T.; Sabeenian, R.S. Modified Alexnet architecture for classification of diabetic retinopathy images. Comput. Electr. Eng. 2019, 76, 56–64. [CrossRef]

30. Li, X.; Hu, X.; Yu, L.; Zhu, L.; Fu, C.W.; Heng, P.A. CANet: Cross-disease Attention Network for Joint Diabetic Retinopathy and Diabetic Macular Edema Grading. IEEE Trans. Med Imaging 2020, 39, 1483–1493. [CrossRef] [PubMed]

31. Dekhil, O.; Naglah, A.; Shaban, M.; Ghazal, M.; Taher, F.; Elbaz, A. Deep Learning Based Method for Computer Aided Diagnosis of Diabetic Retinopathy. In Proceedings of the IST 2019—IEEE International Conference on Imaging Systems and Techniques, Abu Dhabi, United Arab Emirates, 9–10 December 2019; pp. 1–4. [CrossRef]

32. He, A.; Li, T.; Li, N.; Wang, K.; Fu, H. CABNet: Category Attention Block for Imbalanced Diabetic Retinopathy Grading. IEEE Trans. Med. Imaging 2020, 40, 143–153. [CrossRef] [PubMed]

33. Kassani, S.H.; Kassani, P.H.; Khazaeinezhad, R.; Wesolowski, M.J.; Schneider, K.A.; Deters, R. Diabetic retinopathy classification

using a modified xception architecture. In Proceedings of the 2019 IEEE International Symposium on Signal Processing and

Information Technology (ISSPIT), Ajman, United Arab Emirates, 10–12 December 2019; pp. 1–6.

34. Bodapati, J.D.; Shaik, N.S.; Naralasetti, V. Composite deep neural network with gated-attention mechanism for diabetic

retinopathy severity classification. J. Ambient. Intell. Humaniz. Comput. 2021, 1–15. [CrossRef]

35. Hsieh, Y.T.; Chuang, L.M.; Jiang, Y.D.; Chang, T.J.; Yang, C.M.; Yang, C.H.; Chan, L.W.; Kao, T.Y.; Chen, T.C.; Lin, H.C.; et al.

Application of deep learning image assessment software VeriSee™ for diabetic retinopathy screening. J. Formos. Med Assoc. 2021,

120, 165–171. [CrossRef]

36. Zago, G.T.; Andreão, R.V.; Dorizzi, B.; Teatini Salles, E.O. Diabetic retinopathy detection using red lesion localization and

convolutional neural networks. Comput. Biol. Med. 2019, 116, 103537. [CrossRef] [PubMed]

37. Li, T.; Gao, Y.; Wang, K.; Guo, S.; Liu, H.; Kang, H. Diagnostic assessment of deep learning algorithms for diabetic retinopathy

screening. Inf. Sci. 2019, 501, 511–522. [CrossRef]

38. Wang, J.; Luo, J.; Liu, B.; Feng, R.; Lu, L.; Zou, H. Automated diabetic retinopathy grading and lesion detection based on the

modified R-FCN object-detection algorithm. IET Comput. Vis. 2020, 14, 1–8. [CrossRef]

39. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. Commun. ACM

2017, 60, 84–90. [CrossRef]

40. Zisserman, K.S.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the

International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015

41. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with

Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA,

7–12 June 2015; pp. 1–9. [CrossRef]

42. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Trans.

Pattern Anal. Mach. Intell. 2015, 37, 1904–1916. [CrossRef] [PubMed]

43. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer

Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.

44. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.

45. Porwal, P.; Pachade, S.; Kamble, R.; Kokare, M.; Deshmukh, G.; Sahasrabuddhe, V.; Meriaudeau, F. Indian Diabetic Retinopathy

Image Dataset (IDRiD): A Database for Diabetic Retinopathy Screening Research. Data 2018, 3, 25. [CrossRef]

46. APTOS 2019 Blindness Detection. Available online: https://www.kaggle.com/c/aptos2019-blindness-detection/overview/

evaluation (accessed on 1 January 2020).

Sensors 2021, 21, 3704 22 of 22

47. Alyoubi, W.L.; Shalash, W.M.; Abulkhair, M.F. Diabetic Retinopathy Detection through Deep Learning Technique: A Review.

Inform. Med. Unlocked 2020, 20, 1–11. [CrossRef]

48. Hu, J.; Shen, L.; Sun;, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and

Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.

49. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans.

Pattern Anal. Mach. Intell. 2017, 39, 1137–1149. [CrossRef]

50. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks. arXiv 2016, arXiv:1605.06409.

51. Li, T.; Bo, W.; Hu, C.; Kang, H.; Liu, H.; Wang, K.; Fu, H. Applications of Deep Learning in Fundus Images: A Review. Med. Image

Anal. 2021, 69, 101971. [CrossRef]

52. Esfahani, M.T.; Ghaderi, M.; Kafiyeh, R. Classification of diabetic and normal fundus images using new deep learning method.

Leonardo Electron. J. Pract. Technol. 2018, 17, 233–248.

53. Dutta, S.; Manideep, B.C.; Basha, S.M.; Caytiles, R.D.; Iyengar, N.C.S.N. Classification of Diabetic Retinopathy Images by Using

Deep Learning Models. Int. J. Grid Distrib. Comput. 2018, 11, 99–106. [CrossRef]

54. Zhou, M.; Jin, K.; Wang, S.; Ye, J.; Qian, D. Color Retinal Image Enhancement Based on Luminosity and Contrast Adjustment.

IEEE Trans. Biomed. Eng. 2018, 65, 521–527. [CrossRef] [PubMed]

55. Pisano, E.D.; Zong, S.; Hemminger, B.M.; Deluca, M.; Johnston, R.E.; Muller, K.; Braeuning, M.P.; Pizer, S.M. Contrast Limited

Adaptive Histogram Equalization Image Processing to Improve the Detection of Simulated Spiculations in Dense Mammograms.

J. Digit. Imaging 1998, 11, 193–200. [CrossRef]

56. Zuiderveld, K. Contrast Limited Adaptive Histogram Equalization. Graph. Gems IV 1994, 474–485. [CrossRef]

57. Sonali.; Sahu, S.; Singh, A.K.; Ghrera, S.; Elhoseny, M. An approach for de-noising and contrast enhancement of retinal fundus image using CLAHE. Optics Laser Technol. 2019, 110, 87–98. [CrossRef]

58. Pratt, H.; Coenen, F.; Broadbent, D.M.; Harding, S.P.; Zheng, Y. Convolutional Neural Networks for Diabetic Retinopathy. Procedia Comput. Sci. 2016, 90, 200–205. [CrossRef]

59. Ketkar, N. Deep Learning with Python; Springer: Berlin/Heidelberg, Germany, 2017; pp. 1–235. [CrossRef]

60. Shalash, W.M. Driver Fatigue Detection with Single EEG Channel Using Transfer Learning. In Proceedings of the 2019 IEEE International Conference on Imaging Systems and Techniques (IST), Abu Dhabi, United Arab Emirates, 9–10 December 2019.

61. COVER, T.; HART, P.. Nearest Neighbor Pattern Classfication. IEEE Trans. Inf. Theory 1967, 13, 21–27. [CrossRef]

62. Keras. Available online: https://keras.io/ (accessed on 1 January 2019).

63. Lee, W.Y.; Park, S.M.; Sim, K.B. Optimal hyperparameter tuning of convolutional neural networks based on the parameter-settingfree harmony search algorithm. Optik 2018, 172, 359–367. [CrossRef]

64. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. J. Mach. Learn. Res. 2012, 13, 281–305

65. Huang, Q.; Mao, J.; Liu, Y. An improved grid search algorithm of SVR parameters optimization. In Proceedings of the 2012 IEEE 14th International Conference on Communication Technology, Chengdu, China, 9–11 November 2012; pp. 1022–1026.

66. Maclaurin, D.; Duvenaud, D.; Adams, R. Gradient-based hyperparameter optimization through reversible learning. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 7–9 July 2015; pp. 2113–2122.

67. Smith, L.N. Cyclical learning rates for training neural networks. In Proceedings of the IEEE Winter Conference on Applications

of Computer Vision, WACV 2017, Santa Rosa, CA, USA, 24–31 March 2017; pp. 464–472. [CrossRef]

68. Ahn, S.; Pham, Q.; Shin, J.; Song, S.J. Future Image Synthesis for Diabetic Retinopathy Based on the Lesion Occurrence Probability.

Electronics 2021, 10, 726. [CrossRef]

69. Anton, N.; Dragoi, E.N.; Tarcoveanu, F.; Ciuntu, R.E.; Lisa, C.; Curteanu, S.; Doroftei, B.; Ciuntu, B.M.; Chiseli¸t˘a, D.; Bogd˘anici,

C.M. Assessing Changes in Diabetic Retinopathy Caused by Diabetes Mellitus and Glaucoma Using Support Vector Machines in

Combination with Differential Evolution Algorithm. Appl. Sci. 2021, 11, 3944. [CrossRef]

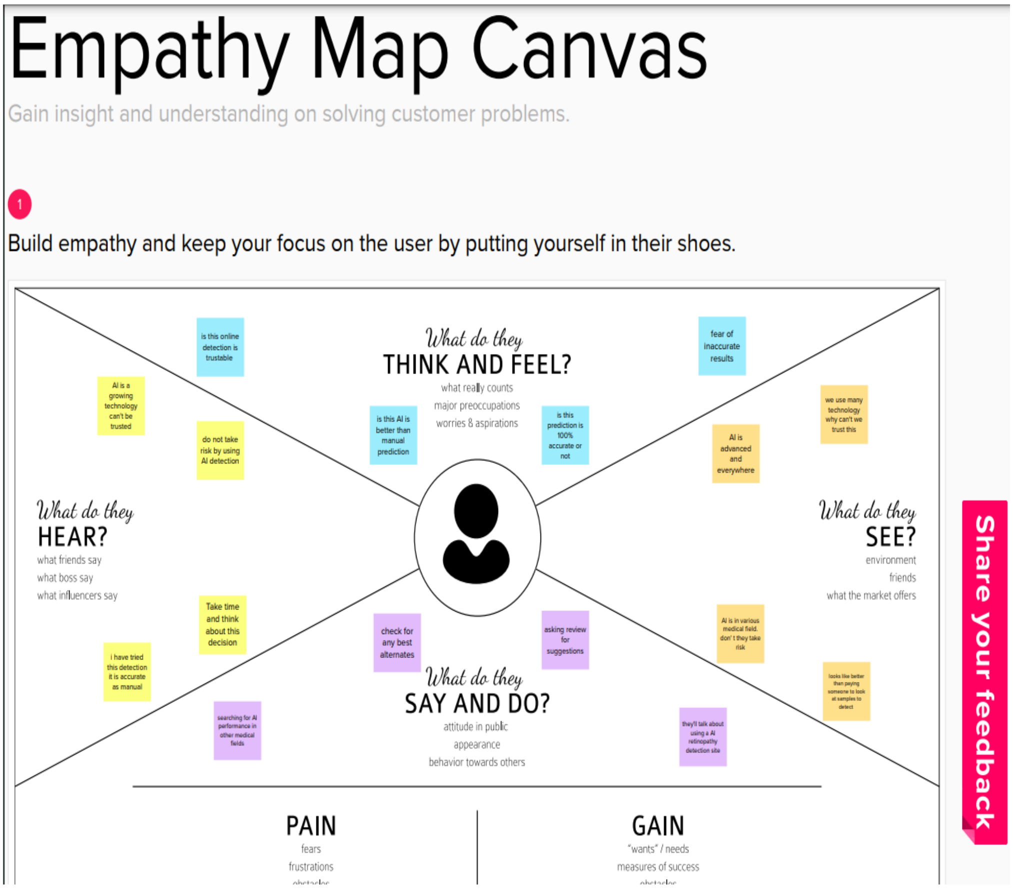
70. Aziz Computer. Available online: http://hpc.kau.edu.sa (accessed on 1 January 2019)

2.3 PROBLEM STATEMENT DEFINITION:

Diabetic retinopathy is one of the most threatening complications of diabetes that leads to permanent blindness if left untreated. One of the essential challenges is early detection, which is very important for treatment success. Unfortunately, the exact identification of the diabetic retinopathy stage is notoriously tricky and requires expert human interpretation of fundus images. Simplification of the detection step is crucial and can help millions of people. Convolutional neural networks (CNN) have been successfully applied in many adjacent subjects, and for diagnosis of diabetic retinopathy itself. However, the high cost of big labeled datasets, as well as inconsistency between different doctors, impede the performance of these methods. In this paper, we propose an automatic deep-learning-based method for stage detection of diabetic retinopathy by single photography of the human fundus. Additionally, we propose the multistage approach to transfer learning, which makes use of similar datasets with different labeling. The presented method can be used as a screening method for early detection of diabetic retinopathy with sensitivity. Primarily occurs when the blood sugar level is unmanageable. Therefore, the person with diabetes mellitus is always at a high risk of acquiring this disease. The early detection can deter the contingency of complete and permanent blindness. Thus, requires an efficient screening system. The present work considers a deep learning methodology specifically a Densely Connected Convolutional Network DenseNet-169, which is applied for the early detection of diabetic retinopathy. It classifies the fundus images based on its severity levels as No DR, Mild, Moderate, Severe and Proliferative DR. The datasets that are taken into consideration are Diabetic Retinopathy Detection 2015 and Aptos 2019 Blindness Detection which are both obtained from Kaggle. The proposed method is accomplished through various steps: Data Collection, Preprocessing, Augmentation and modelling. Our proposed model achieved 90% of accuracy. The Regression model was also employed, manifested up an accuracy of 78%. The main aim of this work is to develop a robust system for detecting DR automatically.

3. IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHY MAP CANVAS:



Empathy Map Canvas

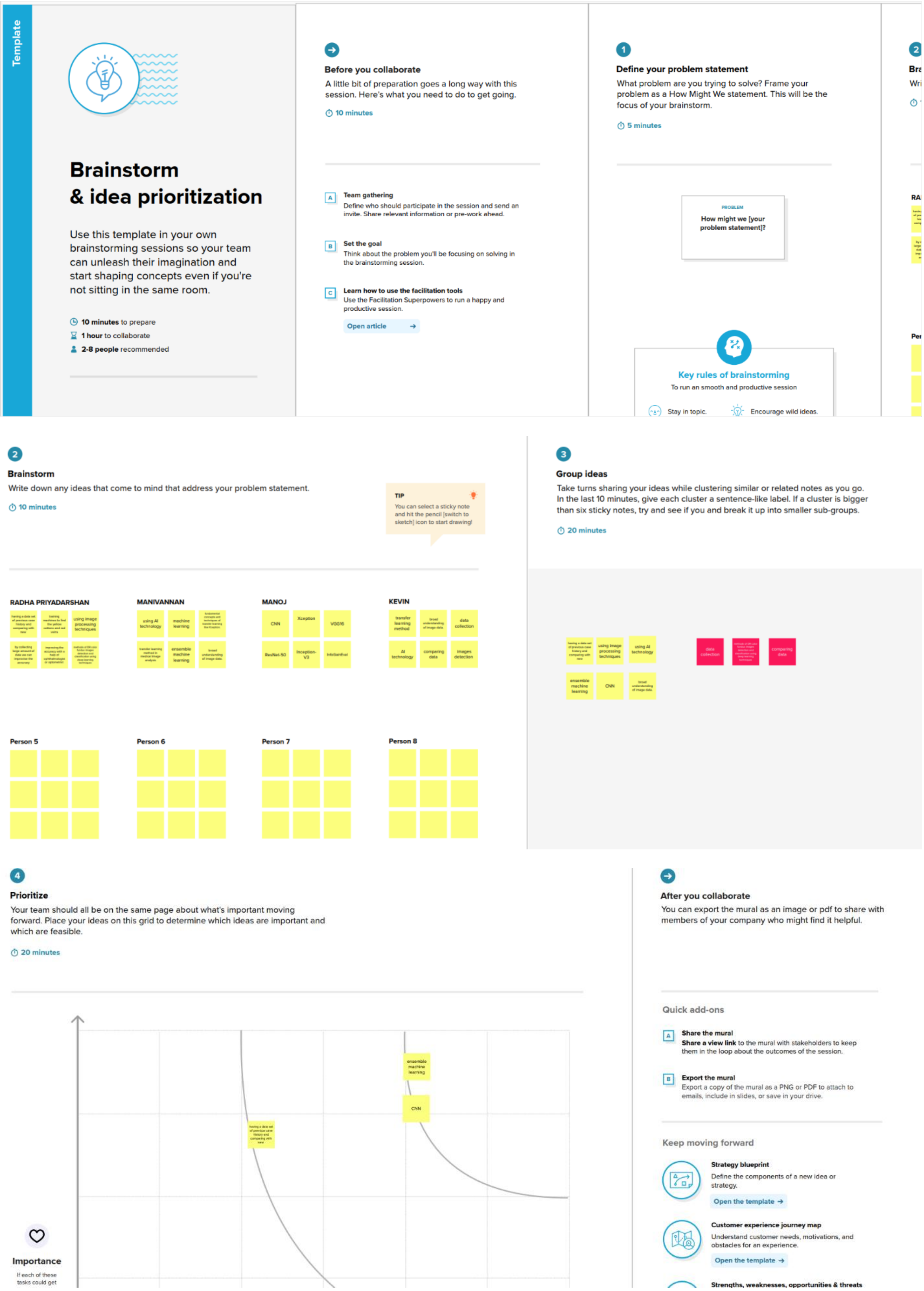
Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
is this online detection is trustable
is this AI is better than manual prediction
is this prediction is 100% accurate or not
fear of inaccurate results
What do they
HEAR?
what friends say
what boss say
what influencers say
AI is a growing technology can't be trusted
do not take risk by using AI detection
Take time and think about this decision
i have tried this detection it is accurate as manual
What do they
SEE?
environment
friends
what the market offers
we use many technology why can't we trust this
AI is advanced and everywhere
AI is in various medical field. don' t they take risk
looks like better than paying someone to look at samples to detect
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
check for any best alternates
asking review for suggestions
searching for AI performance in other medical fields
they'll talk about using a AI retinopathy detection site
PAIN
fears
frustrations
obstacles
GAIN
"wants" / needs
measures of success
obstacles
Share your feedback

3.2 IDEATION AND BRAINSTORMING:



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
How might we [your problem statement]?

Key rules of brainstorming
To run an smooth and productive session

Stay in topic. Encourage wild ideas.

2 **Bra**
Wri

RA

Per

2 **Brainstorm**

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

RADHA PRIYADARSHAN

having a data set of previous case history and comparing with new
training machines to find the yellow [illegible] and red [illegible]
using image processing techniques
by collecting large amount of data we can improvise the accuracy
improving the accuracy with a help of ophthalmologist or optometrist
methods of DR color fundus images detection and classification using deep learning techniques

MANIVANNAN

using AI technology
machine learning
fundamental concepts and techniques of transfer learning like Xception
transfer learning method in medical image analysis
ensemble machine learning
broad understanding of image data.

MANOJ

CNN
Xception
VGG16
ResNet-50
Inception-V3
InfoGainEval

KEVIN

transfer learning method
broad understanding of image data
data collection
AI technology
comparing data
images detection

Person 5

Person 6

Person 7

Person 8

3 **Group ideas**

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

having a data set of previous case history and comparing with new
using image processing techniques
using AI technology
ensemble machine learning
CNN
broad understanding of image data

data collection
methods of DR color fundus images detection and classification using deep learning techniques
comparing data

4 **Prioritize**

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

ensemble machine learning
CNN
having a data set of previous case history and comparing with new
Importance
If each of these tasks could get

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

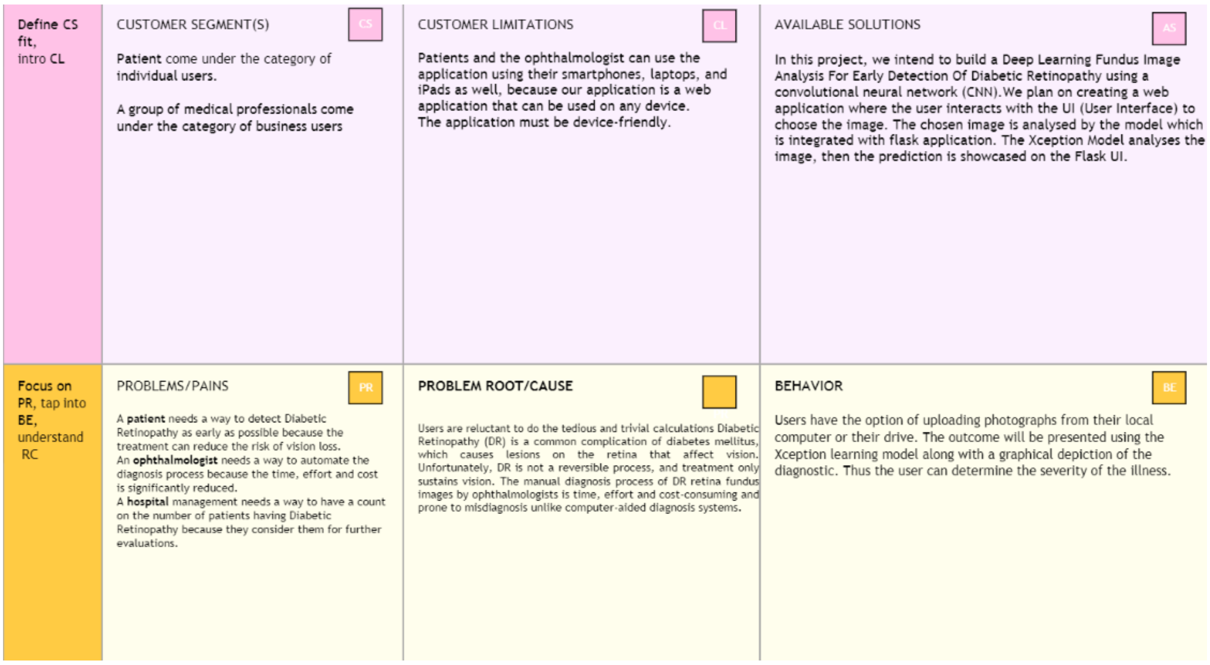
Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats

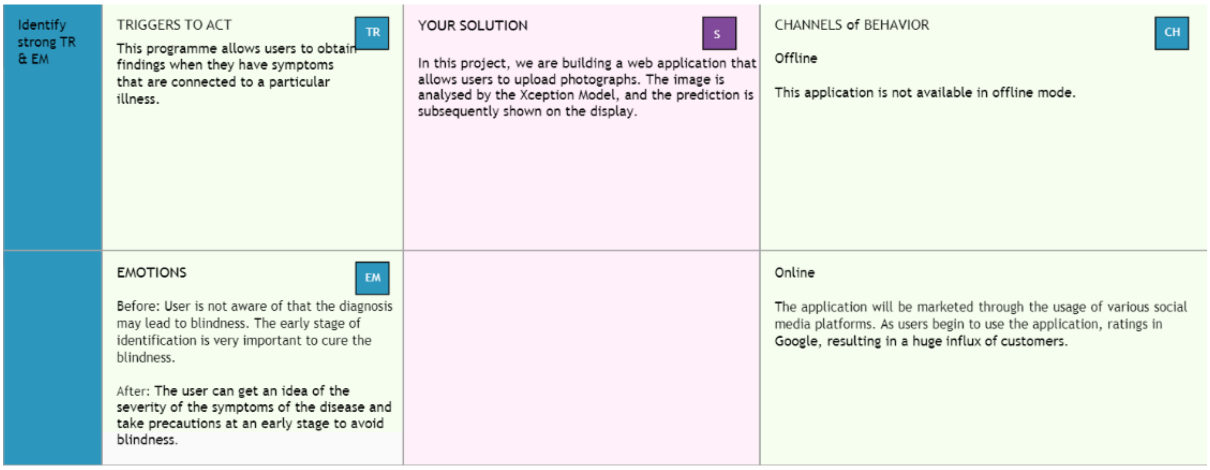
3.3 PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To build a Deep learning model that will extract basic features which can help us in identifying diabetic retinopathy in its early stages. This deep learning system helps in rectifying the damage caused to the blood vessels of the light sensitive tissue at the back of the eye(retina). This method is capable of processing retinal fundus images for early detection of diabetic retinopathy and its degree which also improves the model accuracy.
2.	Idea / Solution description	To accomplish this, the first step is to create Train and Test path folders and then the second step is the image preprocessing in which Import the imagedatagenerator library and applyimagedatagenerator functionality to Trainset and Testset. The third step is Model Building in which Import the model building Libraries,Adding Flatten layers then Adding Output Layer then Creating Model Object then Configure the Learning Process then Train,Save,Test The Model. Step four is Cloudant DB in which Register & Login to IBM Cloud then Create Service Instance and Credentials then Launch Cloudant DB then Create Database. The last step is Application Building in which Building HTML Pages then Build Python Code finally Run The Application
3.	Novelty / Uniqueness	▪ Deep Learning based Diabetic retinopathy detection. ▪ Image processing
4.	Social Impact	A deep learning system could increase the cost effectiveness of screening and diagnosis, while attaining higher than recommended performance, and the system could be applied in clinical examination requiring finer grading.
5.	Business Model	1)A platform to provide details about the disease and its origin and helpful in determining a solution to cure this disease. 2)Subscription based model
6.	Scalability of the Solution	It allows the doctor to accurately identify the severity grades of diabetic retinopathy and macular edema using the high resolution and quality images.

3.4 PROBLEM SOLUTION FIT:



Define CS fit, intro CL	CUSTOMER SEGMENT(S) CS Patient come under the category of individual users. A group of medical professionals come under the category of business users	CUSTOMER LIMITATIONS CL Patients and the ophthalmologist can use the application using their smartphones, laptops, and iPads as well, because our application is a web application that can be used on any device. The application must be device-friendly.	AVAILABLE SOLUTIONS AS In this project, we intend to build a Deep Learning Fundus Image Analysis For Early Detection Of Diabetic Retinopathy using a convolutional neural network (CNN).We plan on creating a web application where the user interacts with the UI (User Interface) to choose the image. The chosen image is analysed by the model which is integrated with flask application. The Xception Model analyses the image, then the prediction is showcased on the Flask UI.
Focus on PR, tap into BE, understand RC	PROBLEMS/PAINS PR A **patient** needs a way to detect Diabetic Retinopathy as early as possible because the treatment can reduce the risk of vision loss. An **ophthalmologist** needs a way to automate the diagnosis process because the time, effort and cost is significantly reduced. A **hospital** management needs a way to have a count on the number of patients having Diabetic Retinopathy because they consider them for further evaluations.	**PROBLEM ROOT/CAUSE** Users are reluctant to do the tedious and trivial calculations Diabetic Retinopathy (DR) is a common complication of diabetes mellitus, which causes lesions on the retina that affect vision. Unfortunately, DR is not a reversible process, and treatment only sustains vision. The manual diagnosis process of DR retina fundus images by ophthalmologists is time, effort and cost-consuming and prone to misdiagnosis unlike computer-aided diagnosis systems.	BEHAVIOR BE Users have the option of uploading photographs from their local computer or their drive. The outcome will be presented using the Xception learning model along with a graphical depiction of the diagnostic. Thus the user can determine the severity of the illness.



Identify strong TR & EM	TRIGGERS TO ACT TR This programme allows users to obtain findings when they have symptoms that are connected to a particular illness.	YOUR SOLUTION S In this project, we are building a web application that allows users to upload photographs. The image is analysed by the Xception Model, and the prediction is subsequently shown on the display.	CHANNELS of BEHAVIOR CH Offline This application is not available in offline mode.
	EMOTIONS EM Before: User is not aware of that the diagnosis may lead to blindness. The early stage of identification is very important to cure the blindness. After: The user can get an idea of the severity of the symptoms of the disease and take precautions at an early stage to avoid blindness.		Online The application will be marketed through the usage of various social media platforms. As users begin to use the application, ratings in Google, resulting in a huge influx of customers.

4.REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENTS:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Separate login for doctor and patients along with username and password
FR-4	Contact details	The contact details of the health care specialists are displayed.
FR-5	Input	User will be able to upload the image from there personal system into the website
FR-6	Output	The accuracy of the situation and the relevant information is displayed according to the obtained result from the prediction
FR-7	Training	The model should be able to be trained with new datasets to increase the accuracy.
FR-8	Image processing accuracy	The prediction accuracy should be correct and there should not be any discrepancy
FR-9	Feedback input	The feedback from the user of the system is required to make the system more efficient.

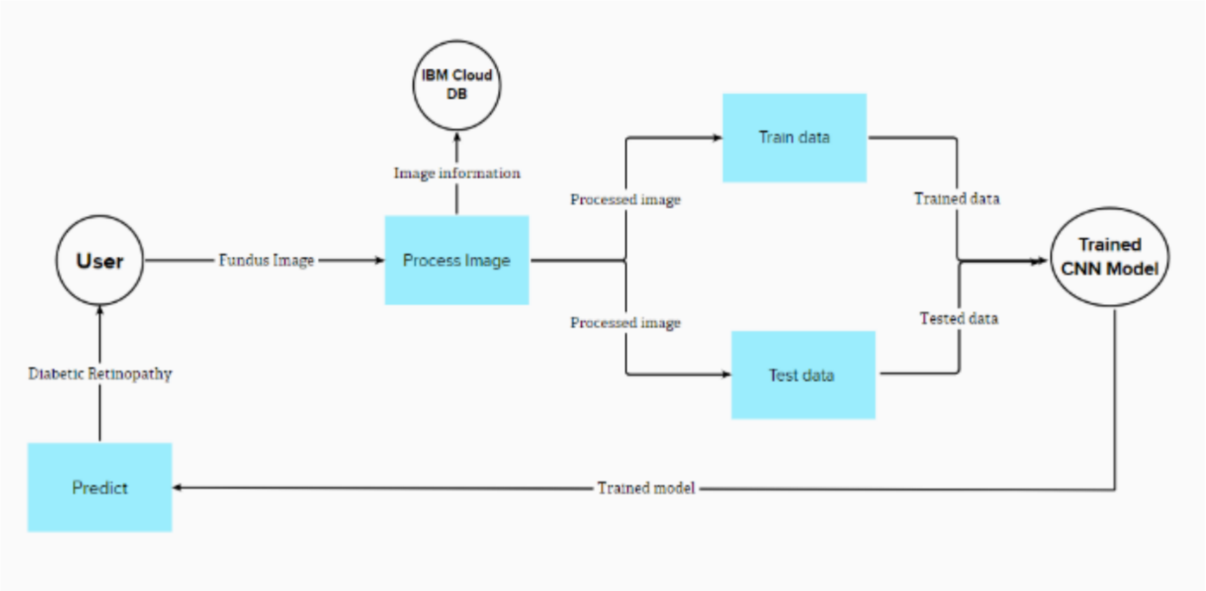
4.2 NON-FUNCTIONAL REQUIREMENTS:

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The website must built with simple English vocabulary so that the users can understand The input dialog box should mention the type of image and maximum size of the image which is permitted to be uploaded
NFR-2	Security	Only the admin should have the permission to access the whole system and have the privilege to update the model with datasets.
NFR-3	Reliability	All the data should be securely stored in the cloud for backup Should be available even during update or rollback phases.
NFR-4	Performance	The loading time for each page should be less than 2 sec to provide the user a better experience The prediction time should be less and the be able to achieve better accuracy.
NFR-5	Availability	New module deployment mustn’t impact website pages availability and mustn't take longer than one hour to be live. The pages that may experience problems must display a notification with a timer showing when the system is going to be up again.
NFR-6	Scalability	The database size should be able to be increased without affecting the existing records. The website should be able to handle up to 5000 users at a time

5. PROJECT DESIGN:

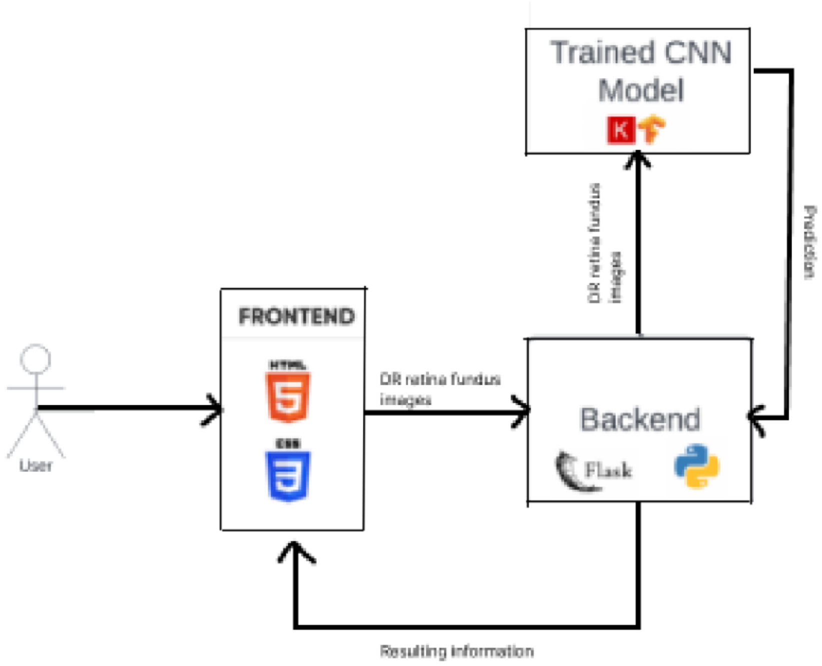
5.1 DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

Solution Architecture
MINIMUM VIABLE ARCHITECTURE FOR MVP



Technologies needed for Minimum Viable Product deployment

Upon research it was found that we need require the following software technologies

for the systematic development and deployment of the project:

- HTML/CSS/JavaScript/Bootstrap – Front End Development
- Python
- TensorFlow
- Image Processing Basics
- Flask – Backend Development
- Git & GitHub – Project Management
- IBM Cloud – Hosting

● IBM Watson – Training the Deep Learning Model

Technical Architecture:

The Technical Architecture has the following blocks:

□ Data Collection.

o Create a Train and Test path.

□ Data Pre-processing.

□ Import the required library

□ Configure ImageDataGenerator class

□ Apply ImageDataGenerator functionality to Trainset and Testset

□ Model Building

o Pre-trained CNN model as a Feature Extractor

o Adding Dense Layer

o Configure the Learning Process

o Train the model

o Save the Model

o Test the model

□ Cloudant DB

o Register & Login to IBM Cloud

o Create Service Instance

o Creating Service Credentials

o Launch Cloudant DB

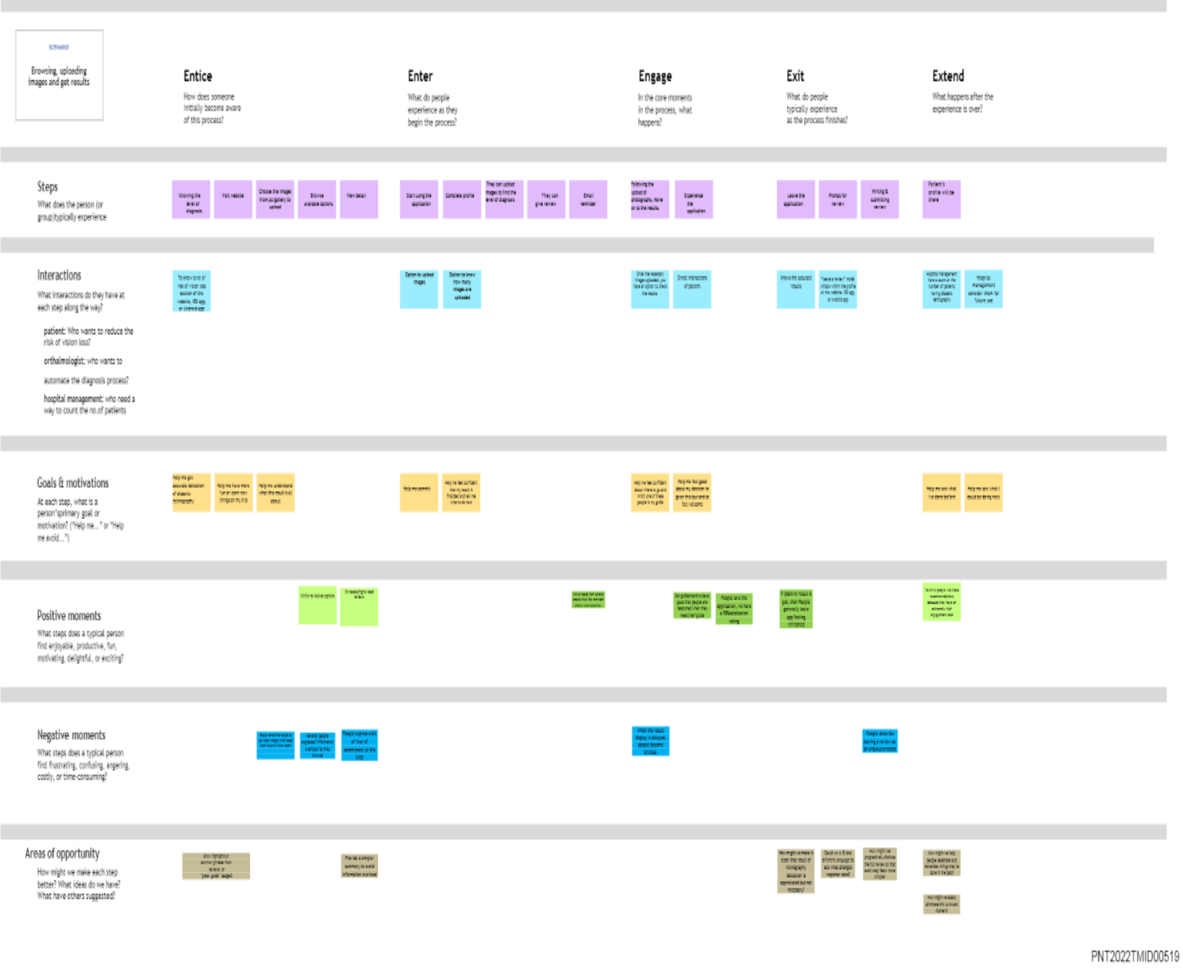
o Create Database

□ Application Building

o Create an HTML file

o Build Python Code

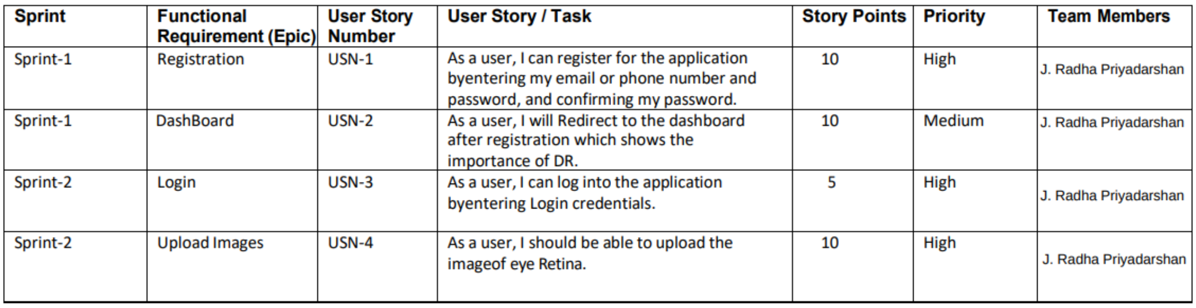
5.3 USER STORIES:

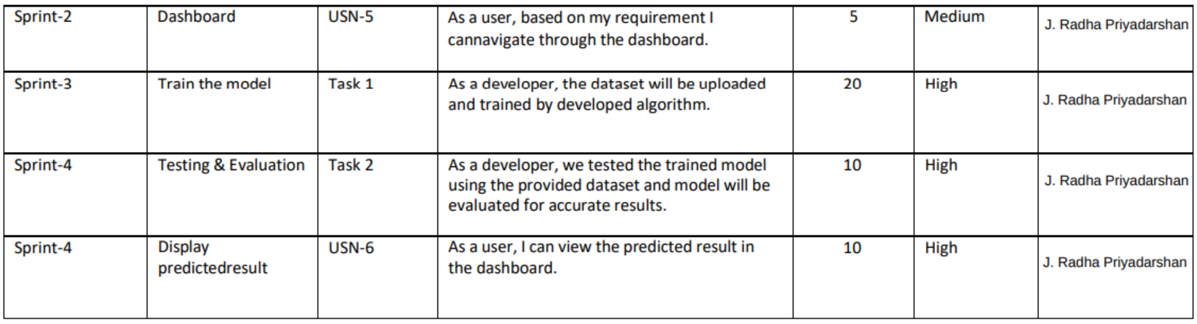

SCENARIO
Browsing, uploading images and get results
Entice
How does someone initially become aware of this process?
Enter
What do people experience as they begin the process?
Engage
In the core moments in the process, what happens?
Exit
What do people typically experience as the process finishes?
Extend
What happens after the experience is over?
Steps
What does the person (or group)typically experience
Knowing the level of diagnosis
Visit website
Choose the images from gallery to upload
Browse available options
View detail
Start using the application
Complete profile
They can upload images to find the level of diagnosis
They can give review
Email reminder
Following the upload of photographs, move on to the results.
Experience the application
Leave the application
Prompt for review
Writing & submitting review
Patient's profile will be there
Interactions
What interactions do they have at each step along the way?
patient: Who wants to reduce the risk of vision loss?
orthalmologist: who wants to automate the diagnosis process?
hospital management: who need a way to count the no.of patients
Option to upload images
Option to know how many images are uploaded
Direct interactions of patient
Knows the accurate results
Goals & motivations
At each step, what is a person'sprimary goal or motivation? ("Help me..." or "Help me avoid...")
Help me get accurate detection of diabetic retinopathy
Help me understand what this result is all about
Help me commit
Positive moments
What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?
It's fun to look at options
Negative moments
What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?
People express a lot of fear of commitment at this step
Areas of opportunity
How might we make each step better? What ideas do we have? What have others suggested?
Provide a simpler summary to avoid information overload
PNT2022TMID00519

6.PROJECT PLANNING AND SCHEDULING:

6.1 SPRINT PLANNING AND ESTIMATION:

To create product backlog and sprint schedule.





Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application byentering my email or phone number and password, and confirming my password.	10	High	J. Radha Priyadarshan
Sprint-1	DashBoard	USN-2	As a user, I will Redirect to the dashboard after registration which shows the importance of DR.	10	Medium	J. Radha Priyadarshan
Sprint-2	Login	USN-3	As a user, I can log into the application byentering Login credentials.	5	High	J. Radha Priyadarshan
Sprint-2	Upload Images	USN-4	As a user, I should be able to upload the imageof eye Retina.	10	High	J. Radha Priyadarshan
Sprint-2	Dashboard	USN-5	As a user, based on my requirement I cannavigate through the dashboard.	5	Medium	J. Radha Priyadarshan
Sprint-3	Train the model	Task 1	As a developer, the dataset will be uploaded and trained by developed algorithm.	20	High	J. Radha Priyadarshan
Sprint-4	Testing & Evaluation	Task 2	As a developer, we tested the trained model using the provided dataset and model will be evaluated for accurate results.	10	High	J. Radha Priyadarshan
Sprint-4	Display predictedresult	USN-6	As a user, I can view the predicted result in the dashboard.	10	High	J. Radha Priyadarshan

6.2 SPRINT DELIVERY SCHEDULE:

Sprint	Total story point	Duration	Sprint Start Date	Sprint EndDate (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV)per iteration unit (story points per day).

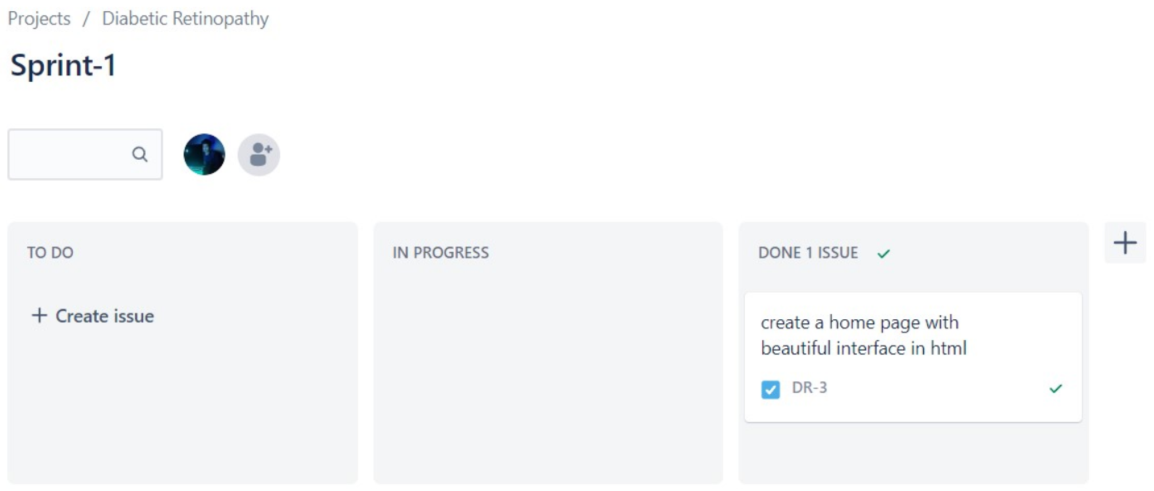
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

AV=20/6=3.33points per day.

6.3 REPORTS FROM JIRA:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

JIRA Folder is created to show the Scrum methodologies and Burn Down chart progress.



Projects / Diabetic Retinopathy

Sprint-1

TO DO

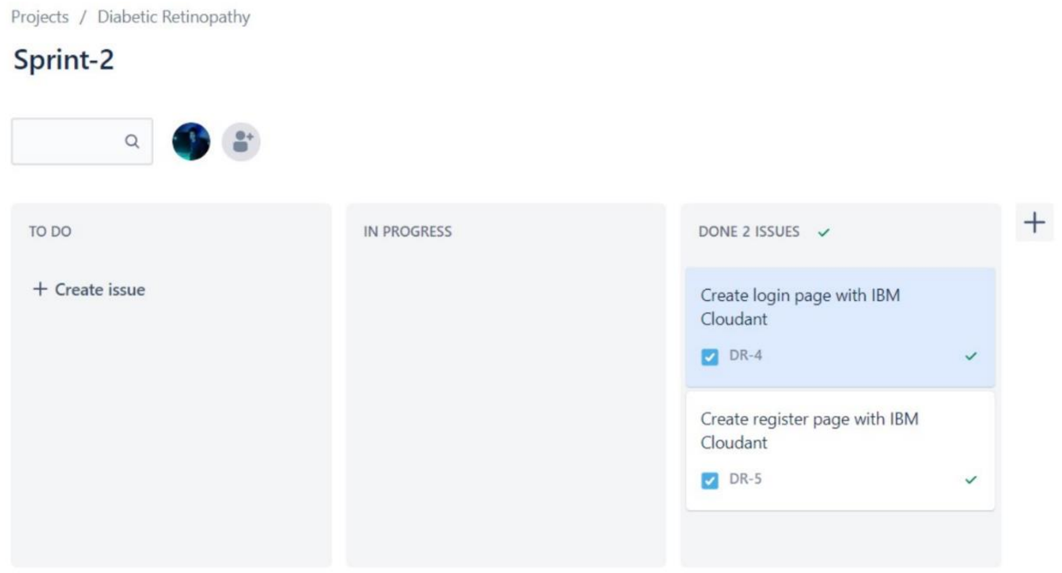
+ Create issue

IN PROGRESS

DONE 1 ISSUE ✓

create a home page with beautiful interface in html

☑ DR-3 ✓



Projects / Diabetic Retinopathy

Sprint-2

TO DO

+ Create issue

IN PROGRESS

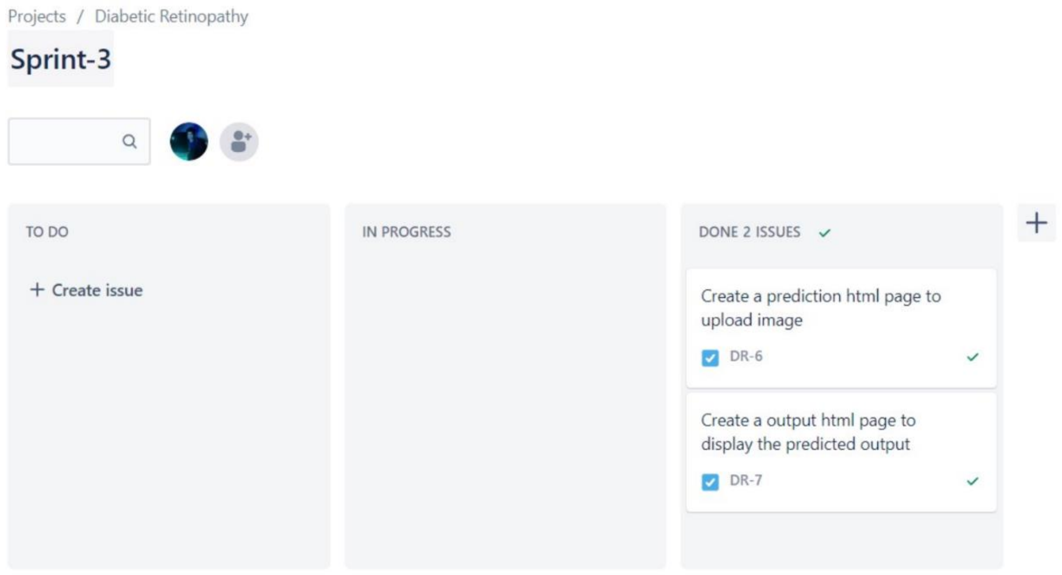
DONE 2 ISSUES ✓

Create login page with IBM Cloudant

☑ DR-4 ✓

Create register page with IBM Cloudant

☑ DR-5 ✓



Projects / Diabetic Retinopathy

Sprint-3

TO DO

+ Create issue

IN PROGRESS

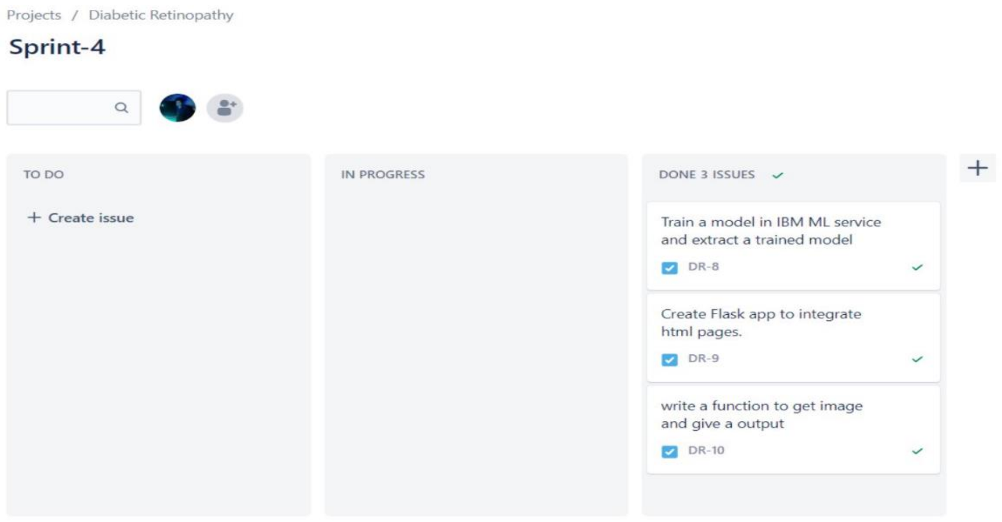
DONE 2 ISSUES ✓

Create a prediction html page to upload image

☑ DR-6 ✓

Create a output html page to display the predicted output

☑ DR-7 ✓



Projects / Diabetic Retinopathy

Sprint-4

TO DO

+ Create issue

IN PROGRESS

DONE 3 ISSUES ✓

Train a model in IBM ML service and extract a trained model

☑ DR-8 ✓

Create Flask app to integrate html pages.

☑ DR-9 ✓

write a function to get image and give a output

☑ DR-10 ✓

7.CODING AND SOLUTIONING:

7.1 FEATURE 1:

```
client=Cloudant.iam('6b2497c0-c725-4938-b8a1-1c3e458f5e77-
bluemix','aXe9BTFF_q8czjIKToN3ALOYUhXda38k37bW2maLgO
AF', connect=True)
my_database=client.create_database('my_database')
def afterlogin():
  user = request.form['_id']
  passw= request.form['psw']
  print(user,passw)

  query={'_id':{'$eq': user}}

  docs= my_database.get_query_result(query)
  print(docs)
  print(len(docs.all()))

  if (len(docs.all())==0):
    return render_template('login.html')
  else:
    if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
      return render_template('prediction.html')
    else:
      print('invalid user')
```

Serverless

Instantly deploy an instance, create databases and independently scale throughput capacity and data storage to meet your application requirements.

Global availability

Get continuous availability as Cloudant distributes data across availability zones and 6 regions for app performance and disaster recovery requirements.

7.2 FEATURE 2:

```
client=Cloudant.iam('6b2497c0-c725-4938-b8a1-1c3e458f5e77-bluemix','aXe9BTFF_q8czjIKToN3ALOYUhXda38k37bW2maLgOAF', connect=True)

my_database=client.create_database('my_database')


def afterreg():
   username=request.form['_id']
   password=request.form['psw']
   data={
   '_id':username,
   # 'name':x[0],
   'psw':password
   }
   print(data)
```

```
 query={'_id': {'$eq':username}}

 docs=my_database.get_query_result(query)
 print(docs)
 print(len(docs.all()))

 if(len(docs.all())==0):
   url=my_database.create_document(data)
   return render_template('prediction.html')
 else:
   return render_template('register.html')
```

Secure

Encrypt all data, with optional user-defined encryption key management through IBM Key Protect, and integrate with IBM Identity and Access Management.

Compliant

Ensure compliance as Cloudant is ISO 27001 compatible and SOC 2 Type 2, PCI, GDPR and HIPAA compliant.

8.TESTING:

8.1 TEST CASES:

1. Verify that as soon as the login page opens, by default the cursor should remain on the username textbox.
2. Verify that the user is able to navigate or access the different controls by pressing the ‘Tab’ key on the keyboard.
3. Check if the password is in masked form when typed in the password field.
4. Check if the password can be copy-pasted or not.
5. Verify that the user is able to login by entering valid credentials and clicking on the ‘Login’ button.
6. Verify that the user is able to login by entering valid credentials and pressing Enter key.
7. Check that the user is not able to login with an invalid username and password.
8. Verify that the validation message gets displayed in case the user leaves the username or password field as blank.
9. Check that the validation message is displayed in the case the user exceeds the character limit of the user name and password fields.
10. Verify that reset button functionality on the login page. Clicking on it should clear the textbox’s content.
11. Verify if there is a checkbox with the label “remember password” on the login page.

12. Verify that closing the browser should not log-out an authenticated user. Launching the application should lead the user to login state only.

REGISTRATION PAGE

1. Verify that all the required fields – username, email, password, confirm password, etc are present on the registration page.

2. Verify that on passing valid values, a user should get registered and the same should be allowed to login to the application.

3. Verify that if a user tries to register an existing username then an error message should get displayed.

4. Verify that the required/mandatory fields are marked with the '*' symbol.

5. Verify that for a better user interface – dropdowns, radio buttons, checkboxes, etc fields are displayed wherever possible instead of just text boxes.

6. Verify the page has both submit and cancel/reset buttons at the end.

7. Verify that clicking submits button after entering all the required fields, submits the data to the server.

8. Verify that clicking the cancel/reset button after entering all the required fields, cancels the submit request, and reset all the fields.

9. Verify that if no value is passed to the mandatory fields and submit button is clicked then it leads to a validation error.

10. Verify that the user can leave the optional fields blank and on clicking the submit button no validation error appears.

11. Verify that whenever possible validation should take place on the client side. For example, if a user presses submit button without entering a username, and password then this validation should take place on the client side instead of sending blank entries to the server.

12. Check the upper limit of the different textbox fields.

13. Verify validation on the date and email fields. Only valid dates and valid email Ids should be allowed.

14. Check validation on numeric fields by entering alphabets and special characters.

15. Check that leading and trailing spaces are trimmed i.e. in case, the user appends space before and after a field, then the same should get trimmed before getting stored on the server.

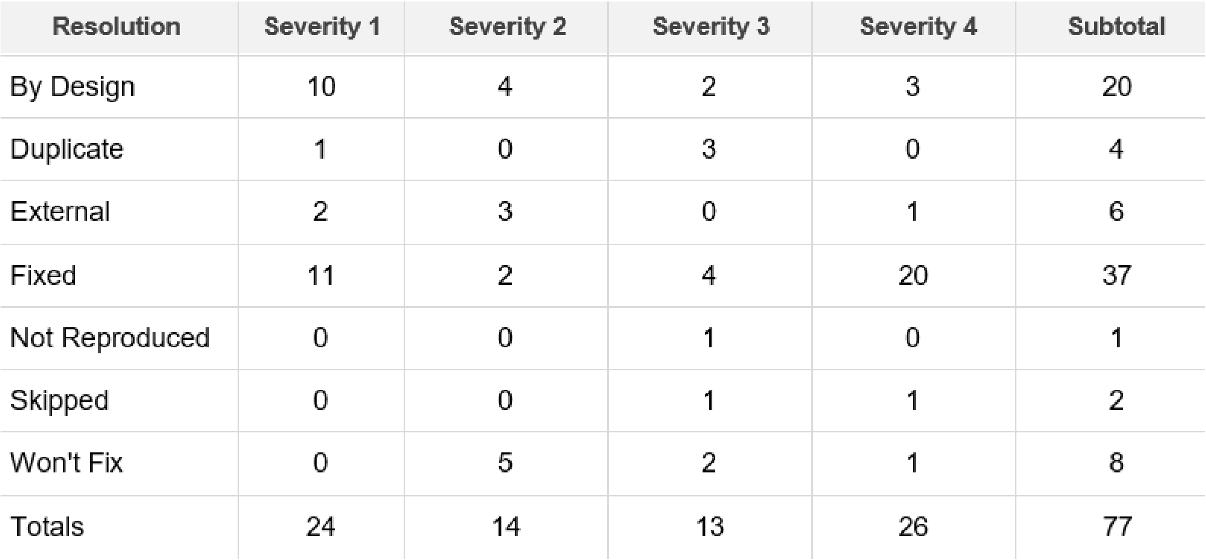
8.2 USER ACCEPTANCE TESTING:

1.Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2.Defect Analysis

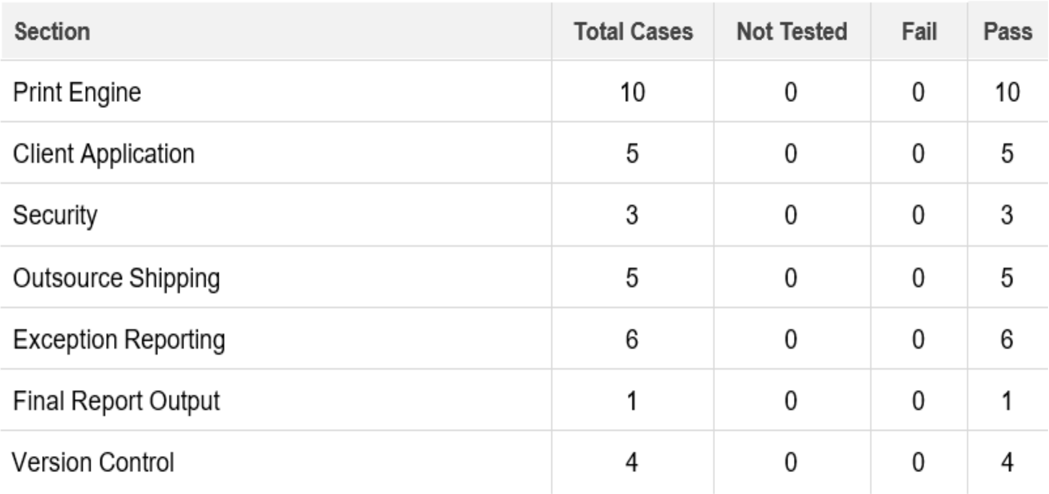
This report shows the number of resolved or closed bugs at each severity level, and how they were resolved



Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

3.Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

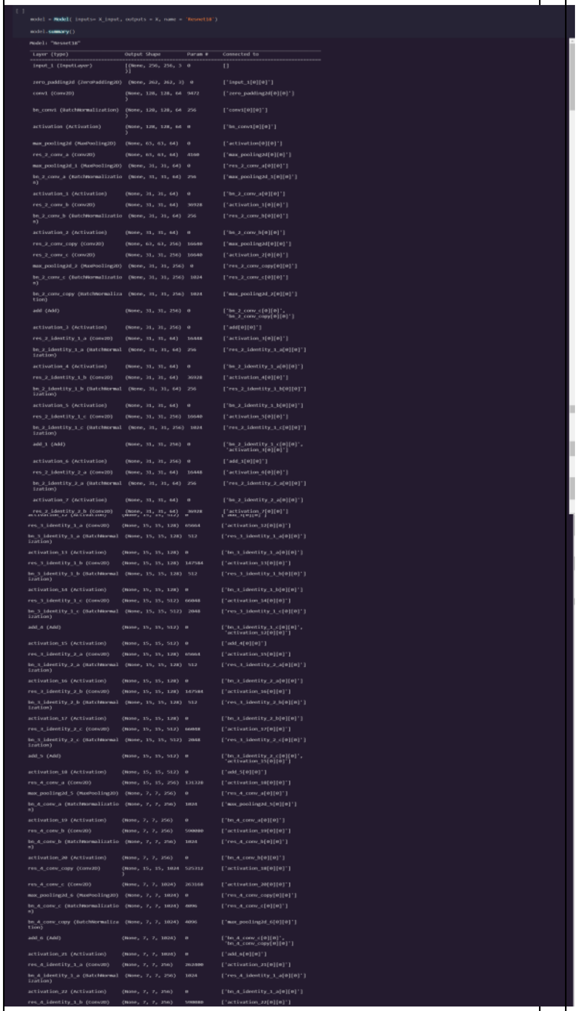


Section	Total Cases	Not Tested	Fail	Pass
Print Engine	10	0	0	10
Client Application	5	0	0	5
Security	3	0	0	3
Outsource Shipping	5	0	0	5
Exception Reporting	6	0	0	6
Final Report Output	1	0	0	1
Version Control	4	0	0	4

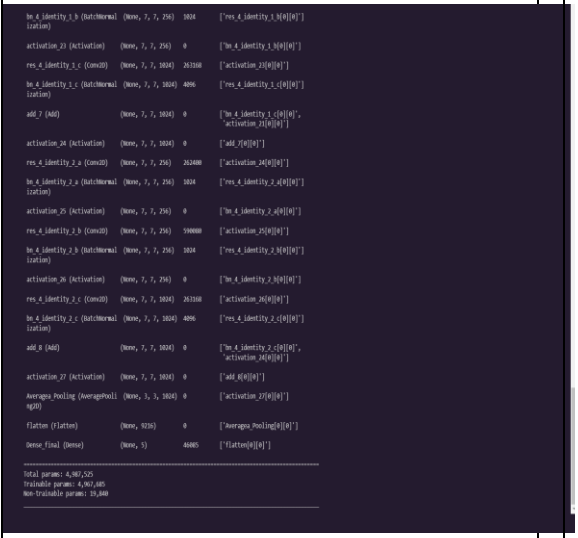
9. RESULTS:

9.1 PERFORMANCE METRICS:

Project team shall fill the following information in model performance testing template.

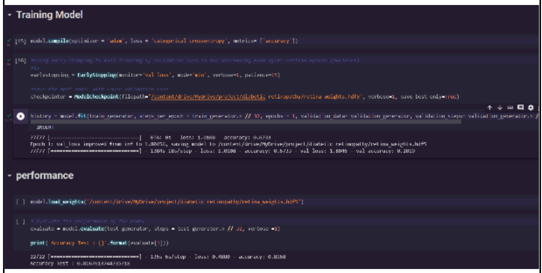
S. No.	Parameter	Values	Screenshot
1	Model Summary	Model: "Resnet18" ____ Layer (type) Output Shape Param # Connected to == input_1 (InputLayer) [(None, 256, 256, 3 0 [])] zero_padding2d (ZeroPadding2D) (None, 262, 262, 3) 0 ['input_1[0][0]'] conv1 (Conv2D) (None, 128, 128, 64 9472 ['zero_padding2d[0][0]']) bn_conv1 (BatchNormalization) (None, 128, 128, 64 256 ['conv1[0][0]']) activation (Activation) (None, 128, 128, 64 0 ['bn_conv1[0][0]']) max_pooling2d (MaxPooling2D) (None, 63, 63, 64) 0 ['activation[0][0]'] res_2_conv_a (Conv2D) (None, 63, 63, 64) 4160 ['max_pooling2d[0][0]'] max_pooling2d_1 (MaxPooling2D) (None, 31, 31, 64) 0 ['res_2_conv_a[0][0]'] bn_2_conv_a (BatchNormalizatio (None, 31, 31, 64) 256 ['max_pooling2d_1[0][0]'] n) activation_1 (Activation) (None, 31, 31, 64) 0 ['bn_2_conv_a[0][0]'] res_2_conv_b (Conv2D) (None, 31, 31, 64) 36928 ['activation_1[0][0]'] bn_2_conv_b (BatchNormalizatio (None, 31, 31, 64) 256 ['res_2_conv_b[0][0]'] n) activation_2 (Activation) (None, 31, 31, 64) 0 ['bn_2_conv_b[0][0]'] res_2_conv_copy (Conv2D) (None, 63, 63, 256) 16640 ['max_pooling2d[0][0]'] res_2_conv_c (Conv2D) (None, 31, 31, 256) 16640 ['activation_2[0][0]'] max_pooling2d_2 (MaxPooling2D) (None, 31, 31, 256) 0 ['res_2_conv_copy[0][0]'] bn_2_conv_c (BatchNormalizatio (None, 31, 31, 256) 1024 ['res_2_conv_c[0][0]'] n) bn_2_conv_copy (BatchNormaliza (None, 31, 31, 256) 1024 ['max_pooling2d_2[0][0]'] tion) add (Add) (None, 31, 31, 256) 0 ['bn_2_conv_c[0][0]', 'bn_2_conv_copy[0][0]'] activation_3 (Activation) (None, 31, 31, 256) 0 ['add[0][0]'] res_2_identity_1_a (Conv2D) (None, 31, 31, 64) 16448 ['activation_3[0][0]'] bn_2_identity_1_a (BatchNormal (None, 31, 31, 64) 256 ['res_2_identity_1_a[0][0]'] ization) activation_4 (Activation) (None, 31, 31, 64) 0 ['bn_2_identity_1_a[0][0]'] res_2_identity_1_b (Conv2D) (None, 31, 31, 64) 36928 ['activation_4[0][0]'] bn_2_identity_1_b (BatchNormal (None, 31, 31, 64) 256 ['res_2_identity_1_b[0][0]'] ization) activation_5 (Activation) (None, 31, 31, 64) 0 ['bn_2_identity_1_b[0][0]']	

```
res_2_identity_1_c (Conv2D)    (None, 31, 31, 256)  16640      ['activation_5[0][0]']

bn_2_identity_1_c (BatchNormal  (None, 31, 31, 256)  1024       ['res_2_identity_1_c[0][0]']
ization)

add_1 (Add)                    (None, 31, 31, 256)  0          ['bn_2_identity_1_c[0][0]',
                                                                'activation_3[0][0]']

activation_6 (Activation)      (None, 31, 31, 256)  0          ['add_1[0][0]']

res_2_identity_2_a (Conv2D)    (None, 31, 31, 64)   16448      ['activation_6[0][0]']

bn_2_identity_2_a (BatchNormal  (None, 31, 31, 64)  256        ['res_2_identity_2_a[0][0]']
ization)

activation_7 (Activation)      (None, 31, 31, 64)   0          ['bn_2_identity_2_a[0][0]']

res_2_identity_2_b (Conv2D)    (None, 31, 31, 64)   36928      ['activation_7[0][0]']

bn_2_identity_2_b (BatchNormal  (None, 31, 31, 64)  256        ['res_2_identity_2_b[0][0]']
ization)

activation_8 (Activation)      (None, 31, 31, 64)   0          ['bn_2_identity_2_b[0][0]']

res_2_identity_2_c (Conv2D)    (None, 31, 31, 256)  16640      ['activation_8[0][0]']

bn_2_identity_2_c (BatchNormal  (None, 31, 31, 256)  1024       ['res_2_identity_2_c[0][0]']
ization)

add_2 (Add)                    (None, 31, 31, 256)  0          ['bn_2_identity_2_c[0][0]',
                                                                'activation_6[0][0]']

activation_9 (Activation)      (None, 31, 31, 256)  0          ['add_2[0][0]']

res_3_conv_a (Conv2D)          (None, 31, 31, 128)  32896      ['activation_9[0][0]']

max_pooling2d_3 (MaxPooling2D)  (None, 15, 15, 128)  0         ['res_3_conv_a[0][0]']

bn_3_conv_a (BatchNormalizatio  (None, 15, 15, 128)  512       ['max_pooling2d_3[0][0]']
n)

activation_10 (Activation)     (None, 15, 15, 128)  0          ['bn_3_conv_a[0][0]']

res_3_conv_b (Conv2D)          (None, 15, 15, 128)  147584     ['activation_10[0][0]']

bn_3_conv_b (BatchNormalizatio  (None, 15, 15, 128)  512       ['res_3_conv_b[0][0]']
n)

activation_11 (Activation)     (None, 15, 15, 128)  0          ['bn_3_conv_b[0][0]']

res_3_conv_copy (Conv2D)       (None, 31, 31, 512)  131584     ['activation_9[0][0]']

res_3_conv_c (Conv2D)          (None, 15, 15, 512)  66048      ['activation_11[0][0]']

max_pooling2d_4 (MaxPooling2D)  (None, 15, 15, 512)  0         ['res_3_conv_copy[0][0]']

bn_3_conv_c (BatchNormalizatio  (None, 15, 15, 512)  2048      ['res_3_conv_c[0][0]']
n)

bn_3_conv_copy (BatchNormaliza  (None, 15, 15, 512)  2048      ['max_pooling2d_4[0][0]']
tion)

add_3 (Add)                    (None, 15, 15, 512)  0          ['bn_3_conv_c[0][0]',
                                                                'bn_3_conv_copy[0][0]']

activation_12 (Activation)     (None, 15, 15, 512)  0          ['add_3[0][0]']

res_3_identity_1_a (Conv2D)    (None, 15, 15, 128)  65664      ['activation_12[0][0]']

bn_3_identity_1_a (BatchNormal  (None, 15, 15, 128)  512       ['res_3_identity_1_a[0][0]']
ization)

activation_13 (Activation)     (None, 15, 15, 128)  0          ['bn_3_identity_1_a[0][0]']

res_3_identity_1_b (Conv2D)    (None, 15, 15, 128)  147584     ['activation_13[0][0]']

bn_3_identity_1_b (BatchNormal  (None, 15, 15, 128)  512       ['res_3_identity_1_b[0][0]']
ization)

activation_14 (Activation)     (None, 15, 15, 128)  0          ['bn_3_identity_1_b[0][0]']

res_3_identity_1_c (Conv2D)    (None, 15, 15, 512)  66048      ['activation_14[0][0]']

bn_3_identity_1_c (BatchNormal  (None, 15, 15, 512)  2048      ['res_3_identity_1_c[0][0]']
ization)

add_4 (Add)                    (None, 15, 15, 512)  0          ['bn_3_identity_1_c[0][0]',
                                                                'activation_12[0][0]']
```



```
bn_4_identity_1_b (BatchNormal  (None, 7, 7, 256)   1024       ['res_4_identity_1_b[0][0]']
ization)

activation_23 (Activation)     (None, 7, 7, 256)    0          ['bn_4_identity_1_b[0][0]']

res_4_identity_1_c (Conv2D)    (None, 7, 7, 1024)   263168     ['activation_23[0][0]']

bn_4_identity_1_c (BatchNormal  (None, 7, 7, 1024)  4096       ['res_4_identity_1_c[0][0]']
ization)

add_7 (Add)                    (None, 7, 7, 1024)   0          ['bn_4_identity_1_c[0][0]',
                                                                'activation_21[0][0]']

activation_24 (Activation)     (None, 7, 7, 1024)   0          ['add_7[0][0]']

res_4_identity_2_a (Conv2D)    (None, 7, 7, 256)    262400     ['activation_24[0][0]']

bn_4_identity_2_a (BatchNormal  (None, 7, 7, 256)   1024       ['res_4_identity_2_a[0][0]']
ization)

activation_25 (Activation)     (None, 7, 7, 256)    0          ['bn_4_identity_2_a[0][0]']

res_4_identity_2_b (Conv2D)    (None, 7, 7, 256)    590080     ['activation_25[0][0]']

bn_4_identity_2_b (BatchNormal  (None, 7, 7, 256)   1024       ['res_4_identity_2_b[0][0]']
ization)

activation_26 (Activation)     (None, 7, 7, 256)    0          ['bn_4_identity_2_b[0][0]']

res_4_identity_2_c (Conv2D)    (None, 7, 7, 1024)   263168     ['activation_26[0][0]']

bn_4_identity_2_c (BatchNormal  (None, 7, 7, 1024)  4096       ['res_4_identity_2_c[0][0]']
ization)

add_8 (Add)                    (None, 7, 7, 1024)   0          ['bn_4_identity_2_c[0][0]',
                                                                'activation_24[0][0]']

activation_27 (Activation)     (None, 7, 7, 1024)   0          ['add_8[0][0]']

Averagea_Pooling (AveragePooli  (None, 3, 3, 1024)  0          ['activation_27[0][0]']
ng2D)

flatten (Flatten)              (None, 9216)         0          ['Averagea_Pooling[0][0]']

Dense_final (Dense)            (None, 5)            46085      ['flatten[0][0]']

==================================================================================================
Total params: 4,987,525
Trainable params: 4,967,685
Non-trainable params: 19,840
__________________________________________________________________________________________________
```

```
activation_15 (Activation)    (None, 15, 15, 512)  0          ['add_4[0][0]']

res_3_identity_2_a (Conv2D)    (None, 15, 15, 128)  65664      ['activation_15[0][0]']

bn_3_identity_2_a (BatchNormal  (None, 15, 15, 128)  512       ['res_3_identity_2_a[0][0]']
ization)

activation_16 (Activation)    (None, 15, 15, 128)  0          ['bn_3_identity_2_a[0][0]']

res_3_identity_2_b (Conv2D)    (None, 15, 15, 128)  147584     ['activation_16[0][0]']

bn_3_identity_2_b (BatchNormal  (None, 15, 15, 128)  512       ['res_3_identity_2_b[0][0]']
ization)

activation_17 (Activation)    (None, 15, 15, 128)  0          ['bn_3_identity_2_b[0][0]']

res_3_identity_2_c (Conv2D)    (None, 15, 15, 512)  66048      ['activation_17[0][0]']

bn_3_identity_2_c (BatchNormal  (None, 15, 15, 512)  2048      ['res_3_identity_2_c[0][0]']
ization)

add_5 (Add)                    (None, 15, 15, 512)  0          ['bn_3_identity_2_c[0][0]',
                                                                'activation_15[0][0]']

activation_18 (Activation)    (None, 15, 15, 512)  0          ['add_5[0][0]']

res_4_conv_a (Conv2D)          (None, 15, 15, 256)  131328     ['activation_18[0][0]']

max_pooling2d_5 (MaxPooling2D)  (None, 7, 7, 256)   0          ['res_4_conv_a[0][0]']

bn_4_conv_a (BatchNormalizatio  (None, 7, 7, 256)   1024       ['max_pooling2d_5[0][0]']
n)

activation_19 (Activation)    (None, 7, 7, 256)    0          ['bn_4_conv_a[0][0]']

res_4_conv_b (Conv2D)          (None, 7, 7, 256)    590080     ['activation_19[0][0]']

bn_4_conv_b (BatchNormalizatio  (None, 7, 7, 256)   1024       ['res_4_conv_b[0][0]']
n)

activation_20 (Activation)    (None, 7, 7, 256)    0          ['bn_4_conv_b[0][0]']

res_4_conv_copy (Conv2D)       (None, 15, 15, 1024  525312     ['activation_18[0][0]']
                               )

res_4_conv_c (Conv2D)          (None, 7, 7, 1024)   263168     ['activation_20[0][0]']

max_pooling2d_6 (MaxPooling2D)  (None, 7, 7, 1024)  0          ['res_4_conv_copy[0][0]']

bn_4_conv_c (BatchNormalizatio  (None, 7, 7, 1024)  4096       ['res_4_conv_c[0][0]']
n)

bn_4_conv_copy (BatchNormaliza  (None, 7, 7, 1024)  4096       ['max_pooling2d_6[0][0]']
tion)

add_6 (Add)                    (None, 7, 7, 1024)   0          ['bn_4_conv_c[0][0]',
                                                                'bn_4_conv_copy[0][0]']

activation_21 (Activation)    (None, 7, 7, 1024)   0          ['add_6[0][0]']

res_4_identity_1_a (Conv2D)    (None, 7, 7, 256)    262400     ['activation_21[0][0]']

bn_4_identity_1_a (BatchNormal  (None, 7, 7, 256)   1024       ['res_4_identity_1_a[0][0]']
ization)

activation_22 (Activation)    (None, 7, 7, 256)    0          ['bn_4_identity_1_a[0][0]']

res_4_identity_1_b (Conv2D)    (None, 7, 7, 256)    590080     ['activation_22[0][0]']

bn_4_identity_1_b (BatchNormal  (None, 7, 7, 256)   1024       ['res_4_identity_1_b[0][0]']
ization)

activation_23 (Activation)    (None, 7, 7, 256)    0          ['bn_4_identity_1_b[0][0]']

res_4_identity_1_c (Conv2D)    (None, 7, 7, 1024)   263168     ['activation_23[0][0]']

bn_4_identity_1_c (BatchNormal  (None, 7, 7, 1024)  4096       ['res_4_identity_1_c[0][0]']
ization)

add_7 (Add)                    (None, 7, 7, 1024)   0          ['bn_4_identity_1_c[0][0]',
                                                                'activation_21[0][0]']

activation_24 (Activation)    (None, 7, 7, 1024)   0          ['add_7[0][0]']

res_4_identity_2_a (Conv2D)    (None, 7, 7, 256)    262400     ['activation_24[0][0]']

bn_4_identity_2_a (BatchNormal  (None, 7, 7, 256)   1024       ['res_4_identity_2_a[0][0]']
ization)

activation_25 (Activation)    (None, 7, 7, 256)    0          ['bn_4_identity_2_a[0][0]']
```

		res_4_identity_2_b (Conv2D) (None, 7, 7, 256) 590080 ['activation_25[0][0]'] bn_4_identity_2_b (BatchNormal (None, 7, 7, 256) 1024 ['res_4_identity_2_b[0][0]'] ization) activation_26 (Activation) (None, 7, 7, 256) 0 ['bn_4_identity_2_b[0][0]'] res_4_identity_2_c (Conv2D) (None, 7, 7, 1024) 263168 ['activation_26[0][0]'] bn_4_identity_2_c (BatchNormal (None, 7, 7, 1024) 4096 ['res_4_identity_2_c[0][0]'] ization) add_8 (Add) (None, 7, 7, 1024) 0 ['bn_4_identity_2_c[0][0]', 'activation_24[0][0]'] activation_27 (Activation) (None, 7, 7, 1024) 0 ['add_8[0][0]'] Averagea_Pooling (AveragePooli (None, 3, 3, 1024) 0 ['activation_27[0][0]'] ng2D) flatten (Flatten) (None, 9216) 0 ['Averagea_Pooling[0][0]'] Dense_final (Dense) (None, 5) 46085 ['flatten[0][0]'] == Total params: 4,987,525 Trainable params: 4,967,685 Non-trainable params: 19,840 ____	
2	Accuracy	Training Accuracy – 0.6733 Validation Accuracy -0.8167	
3	Confidence Score (Only Yolo Projects)	Class Detected - Confidence Score -	

Training Model

```
[15] model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics= ['accuracy'])

[16] #using early stopping to exit training if validation loss is not decreasing even after certain epochs (patience)
     #15
     earlystopping = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=15)

     #save the best model with lower validation loss
     checkpointer = ModelCheckpoint(filepath="/content/drive/MyDrive/project/diabetic retinopathy/retina_weights.hdf5", verbose=1, save_best_only=True)

     history = model.fit(train_generator, steps_per_epoch = train_generator.n // 32, epochs = 1, validation_data= validation_generator, validation_steps= validation_generator.n /
       INSERT
     77/77 [..............................] - ETA: 0s - loss: 1.0188 - accuracy: 0.6733
     Epoch 1: val_loss improved from inf to 1.88456, saving model to /content/drive/MyDrive/project/diabetic retinopathy/retina_weights.hdf5
     77/77 [==============================] - 1384s 18s/step - loss: 1.0188 - accuracy: 0.6733 - val_loss: 1.8845 - val_accuracy: 0.2019
```

performance

```
[ ] model.load_weights("/content/drive/MyDrive/project/diabetic retinopathy/retina_weights.hdf5")

[ ] # Evaluate the performance of the model
    evaluate = model.evaluate(test_generator, steps = test_generator.n // 32, verbose =1)

    print('Accuracy Test : {}'.format(evaluate[1]))

    22/22 [==============================] - 135s 6s/step - loss: 0.4880 - accuracy: 0.8168
    Accuracy Test : 0.8167613744735718
```

10.**ADVANTAGES:**

Diabetes screening tests are a good preventative method for catching the development of diabetes at an early stage.

Conducting health surveys measuring clinical biomarkers of diabetes, other markers of chronic disease and nutrition status will allow for the determination of population health trends and better understanding of the number of people living with pre-existing and previously undiagnosed diabetes.

For many people with diabetes, checking their blood glucose level each day is an important way to manage their diabetes. Monitoring your blood glucose level is most important if you take insulin. The results of blood glucose monitoring can help you make decisions about food, physical activity, and medicines.

11.**CONCLUSION:**

.Diabetic retinopathy is a serious complication of diabetes mellitus, leading to progressive damage and even blindness of the retina. Its early detection and treatment is important in order to prevent its deterioration and the retina's damage. The interest in applying deep learning in detecting diabetic retinopathy has increased during the past years and as several DL systems evolve and become integrated into the clinical practice, they will enable the clinicians to treat the patients in need more effectively and efficiently. This article presents the current state of research regarding the application of deep learning in diagnosing dia-betic retinopathy. Although deep learning has paved the way for more accurate diagnosis and treatment, further improvements are still necessary regarding performance, interpretability and trustworthiness from opthalmologists. The AI DR tool can assist the clinician with fundus image analysis, which in turn helps to quickly inform the next steps in the patient's treatment. Also, doctors can attend to more patients that need attention without mydriasis. Emerging healthcare technologies emphasize on reducing visits to eye specialists, curtailing the overall cost of treatment and optimizing the number of patients seen by each doctor. AI can help the health care professional in achieving the goal. Though it assists in health care sector but should not substitute a clinician at its current level. Novel developments in the sector of artificial intelligence are opening up new promises for running DR detection and grading algorithms.

12. FUTURE SCOPE:

Deep Leaning and AI has tremendous potential to reshape health care and it is more rapidly to do so. But the legal issues involved with the development and implementation of AI algorithms are considerable. Regulation, legal causes of action such as medical malpractice and product liability, intellectual property, and patient privacy all have real implications for the way AI is developed and deployed. When it comes to AI and machine learning, there are currently more legal questions than answers. How can AI systems ensure consent? How will questions of liability be addressed? How does AI fit into existing ethical frameworks in India? How can the security and accuracy of AI solutions be ensured—particularly in the health sector as individual lives can be at stake and highly sensitive data is being handled? These are few questions that are still to be answered. Use of AI in medical diagnostics, especially in ophthalmology heralds a new era. If proven to be sensitive and specific enough this technology can totally change the way we look at screening programs and community-based ophthalmology programs. Most of the present systems use conventional of 30–50° fundus images. Perhaps applications based on wide field imaging and OCT angiography based vascular analysis might yield even more consistent results. However, the high cost of wide field imaging and OCT angiography may be a limiting factor for this at present. A lot of work is also being done on identifying serum biomarkers for early detection and monitoring of diseases like diabetic retinopathy. Thus, a comprehensive analysis of ocular imaging, systemic parameter profile and other serum biomarkers using AI might provide better insights, perhaps even better conclusions than what human intelligence is capable of deriving.

13.APPENDIX:

SOURCE CODE:

App.py

```
From pydoc import render_doc


import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import keras
import os
import matplotlib.pyplot as plt
import PIL
import seaborn as sns
import plotly
import plotly.graph_objs as go
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
from plotly.offline import iplot, init_notebook_mode
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.applications.inception_resnet_v2 import InceptionResNetV2
from tensorflow.keras.layers import *
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.initializers import glorot_uniform
from tensorflow.keras.utils import plot_model
from IPython.display import display
from tensorflow.keras import backend as K
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping, ModelCheckpoint, LearningRateScheduler
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import keras
import os
import matplotlib.pyplot as plt
import PIL
import seaborn as sns
import plotly
import plotly.graph_objs as go
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
from plotly.offline import iplot, init_notebook_mode
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.applications.inception_resnet_v2 import InceptionResNetV2
from tensorflow.keras.layers import *
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.initializers import glorot_uniform
from tensorflow.keras.utils import plot_model
from IPython.display import display
from tensorflow.keras import backend as K
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.callbacks import ReduceLROnPlateau, EarlyStopping, ModelCheckpoint, LearningRateScheduler

from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
import cv2
from flask import Flask, render_template, request, url_for
from cloudant.client import Cloudant

client=Cloudant.iam('6b2497c0-c725-4938-b8a1-1c3e458f5e77-bluemix','aXe9BTFF_q8czjIKToN3ALOYUhXda38k37bW2maLgOAF', connect=True)
my_database=client.create_database('my_database')

```

```
app = Flask(__name__)
app.static_folder = 'static'

@app.route('/')
def home():
   return render_template('home.html')

@app.route('/login')
def login():
   return render_template('login.html')

@app.route('/afterlogin', methods=['POST'])
def afterlogin():
   user = request.form['_id']
   passw= request.form['psw']
   print(user,passw)

   query={'_id':{'$eq': user}}

   docs= my_database.get_query_result(query)
   print(docs)
   print(len(docs.all()))

   if (len(docs.all())==0):
      return render_template('login.html')
   else:
      if((user==docs[0][0]['_id'] and passw==docs[0][0]['psw'])):
         return render_template('prediction.html')
      else:
         print('invalid user')
```

```
@app.route('/register')
def register():
    return render_template('register.html')


@app.route('/afterreg', methods=['POST'])
def afterreg():
    username=request.form['_id']
    password=request.form['psw']
    data={
    '_id':username,
    # 'name':x[0],
    'psw':password
    }
    print(data)

    query={'_id': {'$eq':username}}

    docs=my_database.get_query_result(query)
    print(docs)
    print(len(docs.all()))

    if(len(docs.all())==0):
        url=my_database.create_document(data)
        return render_template('prediction.html')
    else:
        return render_template('register.html')




@app.route('/prediction')
def prediction():
    return render_template('prediction.html')

@app.route('/uploader', methods = ['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        f = request.files['file']
        f.save(f.filename)
        file_name=f.filename

        model=load_model("retina_weights.hdf5")
        prediction = []

        image = []

        labels = {0: 'Mild', 1: 'Moderate', 2: 'No_DR', 3:'Proliferate_DR', 4: 'Severe'}

        img= PIL.Image.open(file_name)

        img = img.resize((256,256))

        image.append(img)

        img = np.asarray(img, dtype= np.float32)

        img = img / 255

        img = img.reshape(-1,256,256,3)

        predict = model.predict(img)

        predict = np.argmax(predict)

        prediction.append(labels[predict])


        print(prediction)

        return render_template("output.html", prediction=prediction[0])

if __name__ == '__main__':
    app.run(debug = True)
```

Main.py

```
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import keras
import os
import matplotlib.pyplot as plt
import PIL
import seaborn as sns
import plotly
import plotly.graph_objs as go
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
from plotly.offline import iplot, init_notebook_mode
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.applications.inception_resnet_v2 import
InceptionResNetV2
from tensorflow.keras.layers import *
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.initializers import glorot_uniform
from tensorflow.keras.utils import plot_model
from IPython.display import display
from tensorflow.keras import backend as K
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.callbacks import ReduceLROnPlateau,
EarlyStopping, ModelCheckpoint, LearningRateScheduler
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import keras
import os
import matplotlib.pyplot as plt
import PIL
import seaborn as sns
import plotly
import plotly.graph_objs as go
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle
from plotly.offline import iplot, init_notebook_mode
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applicati ons.resnet50 import ResNet50
from tensorflow.keras.applications.inception_resnet_v2 import
InceptionResNetV2
from tensorflow.keras.layers import *
from tensorflow.keras.models import Model, load_model
from tensorflow.keras.initializers import glorot_uniform
from tensorflow.keras.utils import plot_model
from IPython.display import display
from tensorflow.keras import backend as K
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.callbacks import ReduceLROnPlateau,
EarlyStopping, ModelCheckpoint, LearningRateScheduler
In [ ]:
from google.colab import drive
drive.mount('/content/drive
In [ ]:
from numpy import append
train=[]
label=[]
for i in os.listdir('/content/drive/MyDrive/project/diabetic
retinopathy/train'):

train_class=os.listdir(os.path.join('/content/drive/MyDrive/project/diabetic
retinopathy/train',i))
    for j in train_class:
        img=os.path.join('train',i,j)
        train.append(img)
        label.append(i)


print(f"Number of train images {len(train)}")
sns.countplot(label)
fig,axs=plt.subplots(5,5,figsize=(20,20))
count=0
for i in os.listdir('/content/drive/MyDrive/project/diabetic
retinopathy/train'):

train_class=os.listdir(os.path.join('/content/drive/MyDrive/project/diabetic
retinopathy/train',i))
```

```
  for j in range(5):
    img=os.path.join('/content/drive/MyDrive/project/diabetic
retinopathy/train',i,train_class[j])
    img=PIL.Image.open(img)
    axs[count][j].title.set_text(i)
    axs[count][j].imshow(img)
  count=count+1
fig.tight_layout()

no_of_images_in_class=[]
class_name=[]
for i in os.listdir('/content/drive/MyDrive/project/diabetic
retinopathy/train'):

train_class=os.listdir(os.path.join('/content/drive/MyDrive/project/diabetic
retinopathy/train',i))
  no_of_images_in_class.append(len(train_class))
  class_name.append(i)
  print(f'no of image in {i} is {len(train_class)}')

retina_df=pd.DataFrame({'Image':train,'Labels':label})
retina_df
fig1,ax1=plt.subplots()
ax1.pie(no_of_images_in_class,labels=class_name,autopct='%1.1f%%')
retina_df=shuffle(retina_df)
train,test=train_test_split(retina_df,test_size=0.2)
In [ ]:
train_datagen = ImageDataGenerator(
        rescale = 1./255,
        shear_range = 0.2,
        vertical_flip=True,
        validation_split = 0.15)

test_datagen = ImageDataGenerator(rescale = 1./255)
In [ ]:
train_generator = train_datagen.flow_from_dataframe(
    train,
    directory='/content/drive/MyDrive/project/diabetic retinopathy',
    x_col="Image",
    y_col="Labels",
    target_size=(256, 256),
    color_mode="rgb",
    class_mode="categorical",
    batch_size=32,
    subset='training')

validation_generator = train_datagen.flow_from_dataframe(
    train,
    directory='/content/drive/MyDrive/project/diabetic retinopathy',
    x_col="Image",
    y_col="Labels",
    target_size=(256, 256),
    color_mode="rgb",
    class_mode="categorical",
    batch_size=32,
    subset='validation')

test_generator = test_datagen.flow_from_dataframe(
    test,
    directory='/content/drive/MyDrive/project/diabetic retinopathy',
    x_col="Image",
    y_col="Labels",
    target_size=(256, 256),
    color_mode="rgb",
    class_mode="categorical",  batch_size=32)
def res_block(X, filter, stage):

  # Convolutional_block
  X_copy = X

  f1 , f2, f3 = filter

  # Main Path
  X = Conv2D(f1, (1,1),strides = (1,1), name ='res_'+str(stage)+'_conv_a', 
kernel_initializer= glorot_uniform(seed = 0))(X)
  X = MaxPool2D((2,2))(X)
  X = BatchNormalization(axis =3, name = 'bn_'+str(stage)+'_conv_a')(X)
  X = Activation('relu')(X)

  X = Conv2D(f2, kernel_size = (3,3), strides =(1,1), padding = 'same', 
name ='res_'+str(stage)+'_conv_b', kernel_initializer= 
glorot_uniform(seed = 0))(X)
  X = BatchNormalization(axis =3, name = 'bn_'+str(stage)+'_conv_b')(X)
  X = Activation('relu')(X)
  X = Conv2D(f3, kernel_size = (1,1), strides =(1,1),name ='res_'+str(stage)+'_conv_c', kernel_initializer= glorot_uniform(seed = 0))(X)
  X = BatchNormalization(axis =3, name = 'bn_'+str(stage)+'_conv_c')(X)


  # Short path
  X_copy = Conv2D(f3, kernel_size = (1,1), strides =(1,1),name ='res_'+str(stage)+'_conv_copy', kernel_initializer= glorot_uniform(seed = 0))(X_copy)
  X_copy = MaxPool2D((2,2))(X_copy)
  X_copy = BatchNormalization(axis =3, name = 'bn_'+str(stage)+'_conv_copy')(X_copy)

  # ADD
  X = Add()([X,X_copy])
  X = Activation('relu')(X)

  # Identity Block 1
  X_copy = X


  # Main Path
  X = Conv2D(f1, (1,1),strides = (1,1), name ='res_'+str(stage)+'_identity_1_a', kernel_initializer= glorot_uniform(seed = 0))(X)
  X = BatchNormalization(axis =3, name = 'bn_'+str(stage)+'_identity_1_a')(X)
  X = Activation('relu')(X)

  X = Conv2D(f2, kernel_size = (3,3), strides =(1,1), padding = 'same', name ='res_'+str(stage)+'_identity_1_b', kernel_initializer= glorot_uniform(seed = 0))(X)
  X = BatchNormalization(axis =3, name = 'bn_'+str(stage)+'_identity_1_b')(X)
  X = Activation('relu')(X)

  X = Conv2D(f3, kernel_size = (1,1), strides =(1,1),name ='res_'+str(stage)+'_identity_1_c', kernel_initializer= glorot_uniform(seed = 0))(X)
  X = BatchNormalization(axis =3, name = 'bn_'+str(stage)+'_identity_1_c')(X)
  # ADD
  X = Add()([X,X_copy])
X = Activation('relu')(X)

  # Identity Block 2
  X_copy = X


  # Main Path
  X = Conv2D(f1, (1,1),strides = (1,1), name
='res_'+str(stage)+'_identity_2_a', kernel_initializer= glorot_uniform(seed
= 0))(X)
  X = BatchNormalization(axis =3, name =
'bn_'+str(stage)+'_identity_2_a')(X)
  X = Activation('relu')(X)

  X = Conv2D(f2, kernel_size = (3,3), strides =(1,1), padding = 'same',
name ='res_'+str(stage)+'_identity_2_b', kernel_initializer=
glorot_uniform(seed = 0))(X)
  X = BatchNormalization(axis =3, name =
'bn_'+str(stage)+'_identity_2_b')(X)
  X = Activation('relu')(X)

  X = Conv2D(f3, kernel_size = (1,1), strides =(1,1),name
='res_'+str(stage)+'_identity_2_c', kernel_initializer= glorot_uniform(seed
= 0))(X)
  X = BatchNormalization(axis =3, name =
'bn_'+str(stage)+'_identity_2_c')(X)

  # ADD
  X = Add()([X,X_copy])
  X = Activation('relu')(X)

  return X
In [ ]:
input_shape = (256,256,3)

#Input tensor shape
X_input = Input(input_shape)

#Zero-padding

X = ZeroPadding2D((3,3))(X_input)
```

```
# 1 - stage

X = Conv2D(64, (7,7), strides= (2,2), name = 'conv1', kernel_initializer=
glorot_uniform(seed = 0))(X)
X = BatchNormalization(axis =3, name = 'bn_conv1')(X)
X = Activation('relu')(X)
X = MaxPooling2D((3,3), strides= (2,2))(X)

# 2- stage
X = res_block(X, filter= [64,64,256], stage= 2)

# 3- stage

X = res_block(X, filter= [128,128,512], stage= 3)

# 4- stage

X = res_block(X, filter= [256,256,1024], stage= 4)

# # 5- stage

#X = res_block(X, filter= [512,512,2048], stage= 5)

# Average Pooling

X = AveragePooling2D((2,2), name = 'Averagea_Pooling')(X)

# Final layer

X = Flatten()(X)
X = Dense(5, activation = 'softmax', name = 'Dense_final',
kernel_initializer= glorot_uniform(seed=0))(X)


model = Model( inputs= X_input, outputs = X, name = 'Resnet18')

model.summary()

model.compile(optimizer = 'adam', loss = 'categorical_crossentropy',
metrics= ['accuracy'])
In [ ]:
```

```
#using early stopping to exit training if validation loss is not decreasing
even after certain epochs (patience)
#15
earlystopping = EarlyStopping(monitor='val_loss', mode='min',
verbose=1, patience=40)

#save the best model with lower validation loss
checkpointer =
ModelCheckpoint(filepath="/content/drive/MyDrive/project/diabetic
retinopathy/weights1.hdf5", verbose=1, save_best_only=True)
In [ ]:
history = model.fit(train_generator, steps_per_epoch = train_generator.n //
32, epochs = 50, validation_data= validation_generator, validation_steps=
validation_generator.n // 32, callbacks=[checkpointer , earlystopping])

In [ ]:
model.load_weights("/content/drive/MyDrive/project/diabetic
retinopathy/retina_weights.hdf5")
In [ ]:
# Evaluate the performance of the model
evaluate = model.evaluate(test_generator, steps = test_generator.n // 32,
verbose =1)

print('Accuracy Test : {}'.format(evaluate[1]))
22/22 [==============================] - 135s 6s/step - loss:
0.4880 - accuracy: 0.8168
Accuracy Test : 0.8167613744735718
In [ ]:
# Assigning label names to the corresponding indexes
labels = {0: 'Mild', 1: 'Moderate', 2: 'No_DR', 3:'Proliferate_DR', 4:
'Severe'}
In [ ]:
# Loading images and their predictions

from sklearn.metrics import confusion_matrix, classification_report,
accuracy_score
import cv2

prediction = []
original = []
image = []
count = 0
```

```
for item in range(1):
  # code to open the image

img= PIL.Image.open('/content/drive/MyDrive/project/diabetic
retinopathy/test/severe.png')
  # resizing the image to (256,256)
  img = img.resize((256,256))
  # appending image to the image list
  image.append(img)
  # converting image to array
  img = np.asarray(img, dtype= np.float32)
  # normalizing the image
  img = img / 255
  # reshaping the image in to a 4D array
  img = img.reshape(-1,256,256,3)
  # making prediction of the model
  predict = model.predict(img)
  # getting the index corresponding to the highest value in the prediction
  predict = np.argmax(predict)
  # appending the predicted class to the list
  prediction.append(labels[predict])
  # appending original class to the list
  original.append(test['Labels'].tolist()[item])
In [ ]:
# Getting the test accuracy
score = accuracy_score(original,prediction)
print("Test Accuracy : {}".format(score))

In [ ]:
Prediction
```

GITHUB AND PROJECT DEMO LINK:

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-10192-1659112371

PROJECT DEMO LINK:

https://vimeo.com/772772463

NALAIYATHIRAN LAB PROJECT

Team ID : PNT2022TMID00519

Team Leader : RADHA PRIYADARSHAN

Team member : MANIVANNAN

Team member : MANOJ

Team member : KEVIN