COMP-3503 - Data Analytics
Meta Data Report
Liam Tingley
100161568

February 23rd, 2024

Introduction and Approach

Diabetic retinopathy (DR) is an eye condition that may cause partial or total vision loss if left untreated. Those with diabetes, of any type, are at particular risk and are advised to receive frequent eye examinations to ensure that, should diabetic retinopathy begin to develop, it can be treated swiftly to avoid any significant vision loss. Diabetic retinopathy occurs when blood vessels in the retina, located at the back of the eye, begin to leak into the vitreous, the gel-like substance which fills the centre of the eye. As the bleeding begins to become more severe, an optometrist will be able to observe spots or streaks in photos taken of the affected eyes.

The convolutional neural network (CNN) developed over the following weeks and months seeks to provide a model with which optometrists are better able to identify the presence of diabetic retinopathy sooner. The dataset provided for handling this problem is one of thousands of images, which will undergo a collection of preprocessing and transformation measures. Specifically, these measures are conversion to grayscale, enhancement, and normalization. The images will be converted to grayscale as the shading spectrum will reduce computational complexity and requirements, while also making edge detection simpler, which will be important for identifying blood vessels against the vitreous in the images. Enhancing the images will increase contrast and brightness, accentuating features such as spots and stretches. Finally, normalizing the data will ensure consistent contributions from the sample data, while also establishing a standardized learning rate across the dataset.

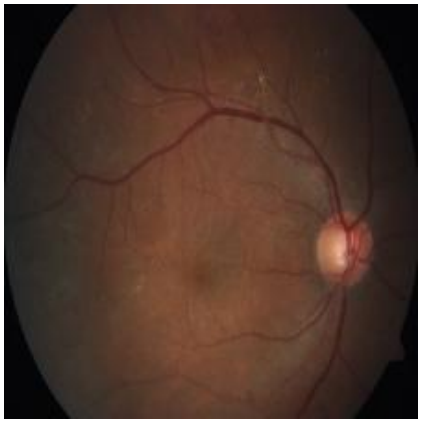
MDR I

The Diabetic Retinopathy dataset consists of several thousand images of eyes from observed patients. As this is an image dataset, the images are the only attribute present. All images are in JPG file format. There are three sets of already-partitioned images, found in the train, test, and valid folders of the PreparedData folder respectively. The training set consists of 2,076 images, while the validation set consists of 531 images, and the testing set consists of 231 images, providing us with a grand total of 2,838 total images. All of the images possess dimensions of 224 pixels in height, and 224 pixels in width. Within each of the training, testing, and validation sets, the images have been separated into examples where diabetic retinopathy is present, and where it isn’t present. Due to the nature of example partitioning, we are presented with a binary classification problem.

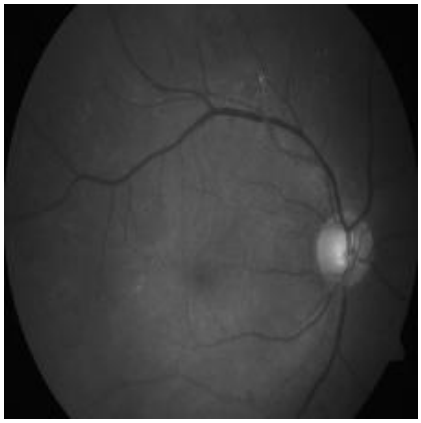
MDR II

Prior to any preprocessing and transformations being applied, the images were quite dark, and a reddish pink. Because of how dark the images are, and the lack of contrasting features they possess, the model would likely struggle to identify the features we are looking for consistently. This presents us with the opportunity to preprocess and transform the image data into more of a

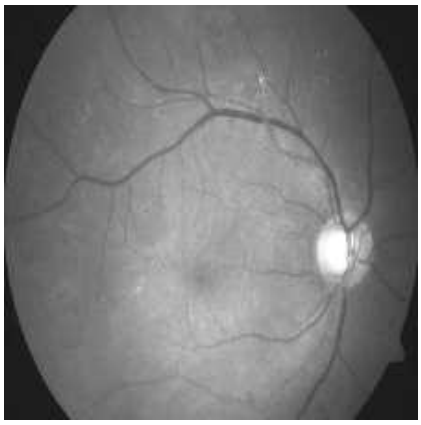
modeling-friendly format. To the right is an example image from the training set that does not contain diabetic retinopathy.



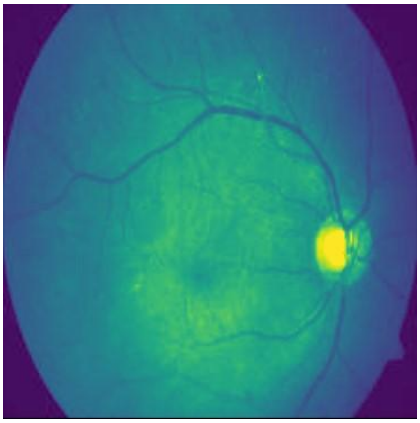
The first transformation conducted was conversion to grayscale. This was done using the CV2 library for Python, and performed in a Google Colaboratory environment. The second image at right is the initial example converted to grayscale from RGB. This conversion eliminates a dimension of complexity from future modeling as it removes colour differentiation.



Following the conversion to grayscale, we might notice that the example is still rather dark, hence we have elected to enhance the image by adjusting its brightness and contrast levels. This enhancement, like the grayscale conversion, is also performed using functions from the CV2 library. When conducting enhancing measures using CV2, we adjust the alpha and beta levels of an image. In the case of our dataset, the alpha parameter, responsible for adjusting the level of contrast, has been scaled upward by a factor of 1.75. While the beta parameter, responsible for adjusting overall brightness level, has been adjusted upward by 20 units. As can be seen in the third image at right, these adjusted parameters reveal additional clarity and detail, particularly regarding the blood vessels.



The final transformation conducted was normalization. As with the prior two preprocessing measures, CV2 was used to perform the normalization. An important disclaimer, the image data contained in the PreparedData folder will appear entirely black, this is due to the pixel data of each image being normalized to a range between 0 and 1. Because of this, the alpha level of the images is extremely low, producing extremely dark imagery. However, these scaled values will still be processable and interpretable by the CNN model, given the nature of convolutional neural networks. The green-tinted image to the right is a variant of the normalized image data with a raised alpha level, which is useful in this case for demonstrating the normalization measures employed. The visual was constructed using the matplotlib Python library.



Conclusion

Thus, the dataset has undergone its first set of preprocessing measures, and is ready for initial modeling. Further adjustments to various portions of the preprocessing steps may yet occur during the modeling process, as these two components of the Data Analytics process are closely intertwined. The prepared dataset can be found in attachment with this report, meanwhile the Jupyter Notebook, developed using Google Colaboratory, can be found at the following link: https://colab.research.google.com/drive/1iIbxvUWNkT5frlFWgVlE3UPU8mgir7B6?usp=sharing

Sources:

1. U.S. Department of Health and Human Services. (n.d.). *Diabetic retinopathy*. National Eye Institute. https://www.nei.nih.gov/learn-about-eye-health/eye-conditions-and-diseases/diabetic-retinopathy
2. Yacine, R. (n.d.). *How to read, write, and save images with opencv and python*. How to Read, Write, and Save Images with OpenCV and Python | Don't Repeat Yourself. https://dontrepeatyourself.org/post/how-to-read-write-and-save-images-with-opencv/
3. GfG. (2023, January 4). *Python: Grayscaling of images using opencv*. GeeksforGeeks. https://www.geeksforgeeks.org/python-grayscaling-of-images-using-opencv/
4. GfG. (2023b, February 15). *Image enhancement techniques using opencv - python*. GeeksforGeeks. https://www.geeksforgeeks.org/image-enhancement-techniques-using-opencv-python/
5. *How to normalize an image in opencv python?*. Tutorialspoint. (n.d.). https://www.tutorialspoint.com/how-to-normalize-an-image-in-opencv-python
6. *How to save files from Google Colab to Google Drive A stepbystep guide*. Saturn Cloud Blog. (2023, November 2). https://saturncloud.io/blog/how-to-save-files-from-google-colab-to-google-drive-a-stepbystep-guide/