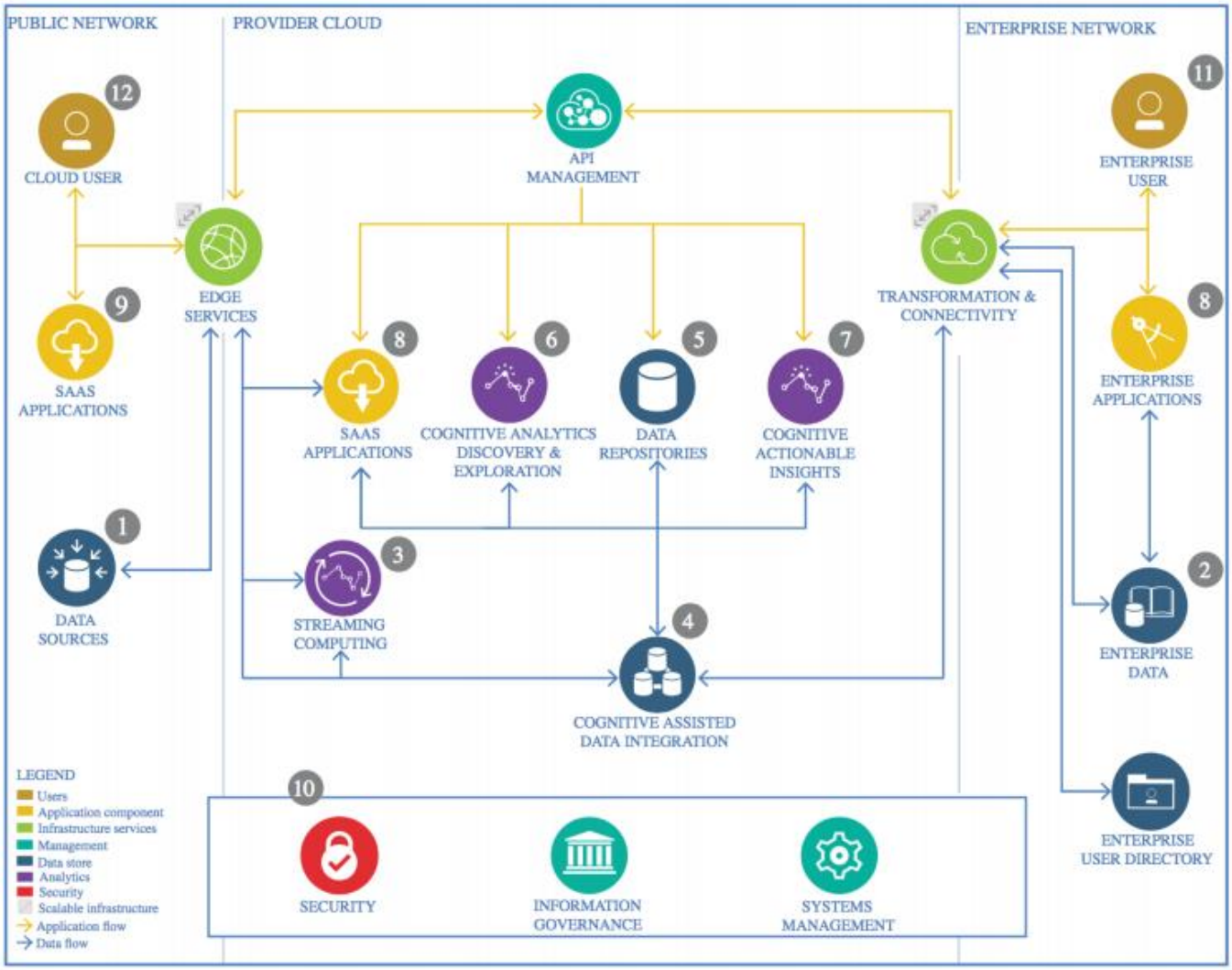
The Lightweight IBM Cloud Garage Method for Data Science

PREDICTING OCCURRENCE OF GLAUCOMA FROM RETINA FUNDUS IMAGES

Author: Laima Marcinkeviciute

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

The data was downloaded from Kaggle (https://www.kaggle.com/sshikamaru/glaucoma-detection)

1.1.1 Technology Choice

For my data science Capstone Project, I chose to analyse glaucoma fundus images to predict if they are glaucoma positive (in which case the patient needs to be referred to ophthalmologist for further diagnostics and treatment) or glaucoma negative (healthy eye vision).

The research is based on ORIGA-light : An Online Retinal Fundus Image Database for Glaucoma Analysis and Research. This Database consists of 650 images (including 168 glaucomatous images and 482 randomly selected nonglaucoma images) from SiMES study. Each image is segmented and annotated by trained professionals from Singapore Eye Research Institute.

According to World Health Organization and the World Glaucoma Association:

There are an estimated **4.5 million blind people globally due to glaucoma**

Percent of all global blindness cause by glaucoma: slightly more than 12%

2nd most common cause of blindness worldwide

There are not enough ophthalmologists to assess and diagnose all people.
We need automated tools for screening, prediction and tracking the progression of eye diseases for whole population.

1.1.2 Justification

The reason I chose to use free online data source for this project is because I could not justify paying to procure a data set for this Capstone project. I could not find many datasets available for glaucoma research due to patient's data protection and due to lack of uniformity in centralized medical records with ophthalmologists' notes.
Finally, I found a public dataset on Kaggle with 650 fundus images available for download and a CSV with file names, CDR (cut to disk ratio) and classification of Glaucoma negative(a) or positive(b)

If CDR ≥ 0.65, the image is Glaucoma positive.
If CDR <0.65, the image is Glaucoma negative.

CSVs are a standard input data form to data science projects allowing for easy input and structuring as a Pandas data frame, for instance.

1.2 Data Quality Assessment

1.2.1 Technology Choice

I checked how many images there are available for download and is which folders they are stored and now they are labelled (650 images (jpeg) saved into labelled input folder Fundus_Train_Val_Data)
I checked how many images are glaucoma positive and how many glaucoma negative and found that there is an uneven distribution of classes (26% glaucomatus (168 images) and 74% nonglaucomatus (482 image)).
- Image size from 192KB – 437KB
- Image dimensions 3072 x 2048 pixels
- Resolution 96 dpi
- Bit depth 24
- RGB color mode

1.2.2 Justification

I have checked various research projects and information available online for Glaucoma analysis and found that Origa-light project is one of the best and it was done by 8 scientists: Zhuo Zhang, Feng Shou Yin, Jiang Liu, Wing Kee Wong, Ngan Meng Tan, Beng Hai Lee, Jun Cheng, Tien Yin Wong. There are 650 fundus images in the dataset and each image is segmented and annotated by trained professionals from Singapore Eye Research Institute. I was very glad to find these images available for download for free on Kaggle.
Knowing that my data source is trustworthy, I am confident to continue using this data for the rest of the Capstone.

1.3 Data Discovery and Exploration

1.3.1 Technology Choice

I was not able to find an easy way to download all the images to IBM data storage cloud without having to upload each single image individually, I decided to you Jupyter notebook on Anaconda and I uploaded images to Anaconda cloud. I have also tried to use Google colab, but for some reason it did not work on my computer so I was glad to find that I am still able to continue with my capstone project using Python 3 and Jupyter notebook on Anaconda.

For image loading / transformation I used:
OS module for interacting with operating system
TKinter for standard Graphical User Interface (GUI) package (filedialog)
Scikit-image for importing data
OpenCV (cv2.imread) for reading jpeg images and (cv2.resize) for resizing jpeg images

I used Pandas for reading csv file and to train the labels.

I used Numpy to import asarray
from PIL import Image
# load the image
image = Image.open('input/Fundus_Train_Val_Data/Fundus_Scanes_Sorted/Validation/Glaucoma_Positive/613.jpg')
# summarize some details about the image
print(image.format)
print(image.mode)
print(image.size)
# show the image
image.show()
pixels = asarray(image)

# global centering
# calculate global mean
mean = pixels.mean()
print('Mean: %.3f' % mean)
print('Min: %.3f, Max: %.3f' % (pixels.min(), pixels.max()))
# global centering of pixels
pixels = pixels - mean
# confirm it had the desired effect
mean = pixels.mean()
print('Mean: %.3f' % mean)
print('Min: %.3f, Max: %.3f' % (pixels.min(), pixels.max()))
print(pixels)

# example of pixel normalization
# confirm pixel range is 0-255
print('Data Type: %s' % pixels.dtype)
print('Min: %.3f, Max: %.3f' % (pixels.min(), pixels.max()))
# convert from integers to floats
pixels = pixels.astype('float32')
# normalize to the range 0-1
pixels /= 255.0
mean = pixels.mean()
print('pixel mean = ', mean)

# I used matplotlib for visualization
import matplotlib.pyplot as plt
fig, (ax0, ax1) = plt.subplots(1, 2)
ax0.imshow(image)
ax0.axis('off')
ax0.set_title('image')
ax1.imshow(pixels)
ax1.axis('off')
ax1.set_title('result')
plt.show()

1.3.2 Justification

After doing my research and meeting difficulties in downloading large amount of images to my dataset, I decided to use Anaconda Navigator, Jupyter notebook 5.7.8, Python 3 and load data to Anaconda Cloud.

1.4 Feature Engineering Method

1.4.1 Technology Choice

I used Scikit-image.feature 'greycomatrix' and 'greycoprops' for extracting features from images
I used SciPy signal for Gaussian window filter
I used (cv2.split) for splitting RGB into 3 channels (Red, Green, Blue) and for Morphological segmentation.
I plotted preprocessed Green channel and smoothed histogram Green channel
I plotted preprocessed Red channel and smoothed histogram Red channel
I used Numpy.convolve to return linear convolution of two one-dimensional sequences.
I used Numpy.ravel to change a 2-dimensional array into contiguous flattened array.
I converted images to grey scale and used Morphological segmentation for defining optic disc from Green channel and optic cup from Red channel

1.4.2 Justification

I have decided to use a Gaussian smoothing filter because I have found out that it works as 'unsharp mask in photoshop' if we subtract the smoothed version from the original image (in a weighted way so the values of a constant area remain constant).
I have found that skimage.feature.greycoprops and greycoprops works very well for feature extraction and image segmentation.

1.5 Machine and Deep Learning Algorithm Choice

1.5.1 Technology Choice

Deep Learning Model:
Convolutional Neural Network (CNN) in Keras, TensorFlow

Machine Learning Model:
sklearn SVM (Support Vector Machine) and
sklearn Random Forest Regressor

1.5.2 Justification

Deep Learning Model:

- I chose to use Convolutional Neural Network (CNN) in Keras, TensorFlow because the CNN-based deep neural system is widely used in the medical classification task. CNN is an excellent feature extractor, therefore utilizing it to classify medical images can avoid complicated and expensive feature engineering.
CNNs are used for image classification and recognition because of its high accuracy. ... The CNN follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully-connected layer where all the neurons are connected to each other and the output is processed.

Machine Learning Model:

- I chose to use SVM (Support Vector Machine) because it has high classifying accuracy and good capabilities of fault-tolerance and generalization and is widely used for classifying medical images.

- I chose to use a Random Forest Regressor because there are multiple advantages to the random forest algorithm including that it is not biased because there are multiple trees trained on different subsets of the data and it is stable even when new data is introduced to the model. Random forest is also robust to missing values and values that have not been properly scaled. Therefore, I conclude it is a good stable, robust model choice for my classification prediction problem.

1.6 Model Evaluation Metric / Performance Indicator

1.6.1 Technology Choice

Classification Performance accuracy to check which model performed best and had a highest accuracy on test data.

1.6.2 Justification

Best results with deep learning model - Convolutional Neural Network (CNN) in Keras, TensorFlow. Accuracy on test data: 93.83%
SVM (support vector machine), accuracy on test data: 71.53%
Random Forest, accuracy on test data: 93.50%

1.7 Data Science Framework / Application Data Products
1.7.1 Technology Choice
Jupyter Notebooks in Python 3
- Python
- Matplotlib & Seaborn for visualization
- Keras for deep learning
- Sklearn for machine learning

1.7.2 Justification
For the implementation of my Capstone project, I chose to use Jupyper Notebooks and coded my solution in Python 3. My justification for this choice was that these were the most well supported technologies introduced in the IBM Advanced Data Science course. I wanted to produce a result that would be easily submitted and exported for review. I chose to use Python as the coding language within the notebooks since Python is a well-known proven data science language. Python's pandas package has good support for efficient data cleaning and processing in data frames. Matplotlib and seaborn are excellent data visualization packages. Keras and sklearn are solid data science packages including functionality for splitting data into training and testing sets, defining deep learning and machine learning models, and model evaluation using various metrics. Python is an open source language which means it has excellent online support on sites such as Stack Overflow and has very well documented packages for ease of implementation and understanding.

1.9 Security, Information Governance and Systems Management
1.9.1 Technology Choice
Worked on publicly available data set and did not used any credential based security restrictions on the out product.
1.9.2 Justification
NA