Displays 70 (2021) 102061

Contents lists available at ScienceDirect

Displays

journal homepage: www.elsevier.com/locate/displa

Early detection of diabetic retinopathy from big data in hadoop framework☆,☆☆

Amartya Hatua [a,*], Badri Narayan Subudhi [b], Veerakumar T. [c], Ashish Ghosh [d]

[a] University of Southern Mississippi, Hattiesburg, MS 39406, USA
[b] Department of Electrical Engineering, Indian Institute of Technology Jammu, Nagrota, Jammu 181221, India
[c] Department of Electronics and Communication Engineering, National Institute of Technology Goa, Ponda, Goa 403401, India
[d] Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India

ARTICLE INFO

Keywords:
Big data
Hadoop
Retinopathy
Histogram oriented gradient
Principal component analysis

ABSTRACT

In this article, we have designed a fast and reliable Diabetic Retinopathy (DR) detection technique in Hadoop framework, which can identify the early signs of diabetes from eye retinal images. In the proposed scheme the retinal images are classified into five categories: No Diabetic Retinopathy (DR), Mild DR, Moderate DR, Severe DR and Proliferative DR. The proposed scheme follows three distinct steps for classification of the diabetic retinopathy images: feature extraction, feature reduction and image classification. In the initial stage of the algorithm, the Histogram of Oriented Gradients (HOG) is used as a feature descriptor to represent each of the Diabetic Retinopathy images. Principal Component Analysis (PCA) is used for dimensional reduction of HOG features. In the final stage of the algorithm, K-Nearest Neighbors (KNN) classifier is used, in a distributed environment, to classify the retinal images to different classes. Experiments have been carried out on a substantial number of high-resolution retinal images taken under an assortment of imaging conditions. Both left and right eye images are provided for every subject. To handle such large datasets, Hadoop platform is used with MapReduce and Mahout framework for programming. The results obtained by the proposed scheme are compared with some of the close competitive state-of-the-art techniques. The proposed technique is found to provide better results than the existing approaches in terms of some standard performance evaluation measures.

1. Introduction

Designing of a computer aided medical diagnostic system needs automatic image analysis to help the medical practitioner. Such a system facilitates a great reduction of the working load of medical practitioners and also helps in early detection of specific diseases of patients. Image analysis [1] showed tremendous usefulness in medical imaging systems [2] for last few years. At the same time, the medical imaging devices are constantly generating medical image data for patients at a faster rate. In 2014 only, worldwide medical image data generation was more than 1ZB. It is expected that by 2028, due to the development of high quality medical imaging devices, each of the medical systems will generate 10–20 GB of image data every day. At the same time, it is expected that the amount of medical image data is supposed to grow at a 40% rate. This indicates that the processing and generation of data grows exponentially. Hence such a system is supposed to generate "Big Data". Big data is a collection of growing volume of heterogeneous datasets which is so large and complex that it becomes difficult to process using available database management tools [3].

Retinopathy imaging [4] is one of the popular imaging modalities used in medical diagnostic systems for analysis of retina images. For a retinopathy data analysis, it includes several challenges: capturing, storage, detection and tracking of the individual from the field of coverage by the camera. Analysis of a single retinal image needs a large storage space as it is large in size with additional information embedded related to data. Use of retinal images got its popularity as its analysis helps early detection of diabetes.

Diabetes is a group of metabolic diseases in which blood sugar level is high for a prolonged period. Diabetes are caused if either the pancreas does not produce enough insulin or the cells of the body do not respond properly to the insulin produced [5]. Diabetic retinopathy (DR) is one of the most common diseases in diabetic patients. It is calculated that

☆ This paper has been recommended for acceptance by G. Guangtao Zhai.☆☆ This work is supported by MeitY with grant no. 4(16)/2019-ITEA.
* Corresponding author.
E-mail address: amartya.hatua@usm.edu (A. Hatua).

https://doi.org/10.1016/j.displa.2021.102061
Received 26 March 2021; Received in revised form 11 July 2021; Accepted 4 August 2021
Available online 27 August 2021

worldwide 93 million people have some DR and 28 million among them are in a critical phase of DR, where they may lose their eyesight [6]. Other than diabetes, blood pressure, glycemic control [7,8], dyslipidemia [9], and nephropathy are also considered as a major cause for DR. As all diabetic patients do not develop DR, some researchers think DR is caused because of some genetic factors.

To get rid of the sight-threatening effect of DR, it is very important for detection of the disease in it's early stage, where risk of losing the eyesight is less and the patient is cured completely over a period of time. Digital color photographs [10] of the retina are analysed to find evidence associated with DR. Such evidences are microaneurysms, hemorrhages, neovascularization or other vascular abnormalities and hard exudate deposits. This process needs expert professionals because intra- and inter-observer variability can create a huge difference. Hence, to diagnose DR, a lot of expertise and equipment are required. These facilities are absent in rural areas or less developed countries. The available equipment in less developed countries are either with lower processor speed, are unable to process high resolution retinal images or have rarely such facility to have an intelligence mechanism to process the images in real-time environment conditions. Further the devices used for such task are unable to manage such large database while processing and updating new database. Several significant works have been done in the past related to automatic detection of diabetic retinopathy [11–23]. However considering the available low processor based equipment and accessing high resolution image is a quite critical job. This motivated the researchers to explore the area of automated methods for DR detection using image processing, pattern recognition, and machine learning tools with easy and low cost solutions in this regard. Considering the importance of the topic, we found in the image processing and Machine learning literature Big-data analytic is one of the important tools suitable for such data analysis.

This article describes the development of a system to identify signs of DR in eye images and classify the images into five different types of diabetic symptoms such as, no DR, mild DR, moderate DR, severe DR, proliferative DR for a large dataset or in other words "Big Data" set framework. Such attempt is the unique and first attempt in the filed. Feature descriptors in an input images are identified using Histogram of Gradients (HoG). Optimal features are extracted using Principal Component Analysis (PCA) and for classification of data KNN algorithm is used. To handle this large data, Hadoop framework is used. MapReduce and Mahout are used as programming framework. The proposed scheme is tested on four different datasets taken from benchmark retinal image database. The proposed scheme is also compared with one existing state-of-the-art competitive methods and the proposed algorithm with different values of K. The performance of the proposed scheme is evaluated using different performance evaluation measures. The proposed scheme is found to provide better results for the tested dataset compared to existing methods.

The organization of the remaining portion of this article is as follows: the description on state-of-the-art-techniques are provided in Section 2. Section 3 describes the proposed methodology. A brief description of Hadoop Framework is given in Section 4. In Section 5, experimental results and discussions along with future works are provided. Conclusions of the works are presented in Section 6.

2. State-of-the-art-techniques

Study about diabetic retinopathy, evaluating the viability of local or systemic treatment by Phillips et al. [13] is considered as one of the initial attempts on diabetic retinopathy. Main objective of this research was to quantify the diabetic maculopathy. The digital fundus [24] imaging and image processing system developed by them put forward a quantitative measurement technique of macular oedema, retinal exudates, and microaneurysms in diabetic retinopathy. To quantify each of the factors they provided certain ways. Fluorescein angiograms was used to determine the degree of macular oedema, severity of oedema, detection and counting of microaneurysms. To detect and measure retinal exudates; a color transparency is projected through a red free filter and a combination of shade correction and thresholding techniques were used to analyze the color transparency. As an early attempt this experiment was very good towards diabetic retinopathy research; but there were several other areas which needed to be improved. First of all this experiment needed human intervention. Other than that this method was not very efficient and accurate also.

The study of Spencer et al. [15] using digital image-processing technique did not involve human interaction. The authors have proposed a novel algorithm and using this microaneurysms can be detected and counted in an image. Both digital and analog images were subjected to manual count by clinicians and their image analysis system. To compare both the results with "gold standards" [25] Free-response ROC (Receiver Operating Characteristic) curves were used.

Although the previously mentioned method was automated, the major short coming was the efficiency and accuracy. To increase both these factors Spencer et al. [14] described a method to detect and quantify microaneurysm present in digitized fluorescein angiogram using some image processing strategies. In this study, initially the image data were preprocessed based on some predefined conditions, and segmentation of the input images were done by a bilinear top-hat transformation and matched filter. These processed images were converted to binary image containing candidate microaneurysm by thresholding. Thresholding technique was used on images containing candidate microaneurysm to convert the processed images into binary images. A region-growing algorithm was introduced in this study. This algorithm was useful to analyze different aspects like size, shape, and energy characteristics of each of the images in the final segmentation of microaneurysm. The correctness of the result using this technique was validated by five clinicians using ROC curves. The results of the proposed automatic system matched with the clinicians' results. To monitor the progress of diabetic retinopathy this strategy is valuable. For small, low contrast microaneurysm, it was difficult to discriminate from small, discrete patches of background fluorescence using this technique.

Earlier attempts for automated microaneurysm detection was performed on isolated images and only simplistic morphological and thresholding techniques were used. In order to get good result, values of sensitivity and specificity should be good; but in thresholding techniques only good sensitivity value was achieved. Moreover this technique did not address the issue of serial study of each patient. Cree et al. [11] addressed this problem. Here one repeatable way was used to quantify microaneurysm in digitized fluorescein angiogram, where microaneurysm count is used as the parameter to access the development and progression of diabetic retinopathy. However this technique is very time consuming, tedious and error prone. In automatic technique, there was very little or no scope of human intervention. This also enhanced efficiency for repeatable analysis. The microaneurysm detection system used for this experiment was based on the method of Spencer et al. [14]. Although Spencer et al. [14] approach was having a significant improvement on previous attempts, it was suboptimal in terms of execution speed and its ability to distinguish micro aneurysms from other pathologies. Moreover, the method required human intervention to identify the specific region of the fundus for counting, and to register images for serial studies. To overcome these problems the authors redesigned the region-growing and classification algorithms used in microaneurysm detection, and addressed automated registration and identification of the macula. The research has been conducted by Hipwell et al. in [12] for the automatic detection of microaneurysms in digital red-free photographs. Initially, the variation of the background intensity is removed from the images, and then the classification of microaneurysm is performed based on its intensity and size. In [17] the authors described a method of automatic analysis of Diabetic retinopathy. They used statistical classifiers, Bayesian, Mahalanobis, and KNN classifiers for testing. The system was tested on 134 retinal images. In [18] image enhancement (noise removal and image normalization) and

pattern recognition techniques to detect early lesions of diabetic retinopathy.

Lee et al. [18] proposed a system to determine whether a computer vision system is as good as human to detect early retinal lesions of diabetic retinopathy using color fundus photographs. Hemorrhages and microaneurysm, hard exudates, and cotton-wool spots can be considered as early lesions of diabetic retinopathy. The system was developed to achieve three purposes: image enhancement, noise removal, and image normalization. The entire system included creating digital image, image quality test, image processing and pattern recognition. The source of the color fundus photographs used in this research is American Indians in Oklahoma. Both development and testing had been done with this images. The results of test data were compared with the results of two human experts, grader at the University of Wisconsin Fundus Photograph Reading Center (Madison) and a general ophthalmologist. The experiment's result was initially divided into three categories (Y/N/Q), but later the third category questionable (Q) was excluded. The research on diabetic retinopathy detection has diverted towards the use of Artificial Intelligence research [26]. Kamble et al. [27] proposed a radial basis function (RBF) neural network classifier for detection of Non DR or DR based images. Recently, research application of different deep learning algorithms in diabetic retinopathy research is prevalent. As diabetic retinopathy mainly deals with images, the application of Convolution Neural Network (CNN) is very common [20,26,28–30,21,22]. In most of the cases the CNN is used for the multiclass classification.

Verbraak et al. [19] proposed development of devices where the hybrid deep learning enhanced architecture for high diagnostic accuracy for the detection of vision-threatening diabetic retinopathy. A deep learning based architecture is designed by Raman et al. [20] to recognize the pathological lesions from fundus images. Rakhonde et al. [28] also proposed a deep CNN architecture for diabetic retinopathy detection. Arcadu et al. [29] proposed a deep convolutional neural network architecture which can predict the DR progression, where the said deep CNN model was trained on early treatment DR and DR Severity Scale. A new multi-layer architecture called as active deep learning model (trained with an active learning framework) is proposed by Qureshi et al. [30] to detect DR. The use of deep learning architecture is reported by Gulshan et al. [21] for detecting diabetic retinopathy and macular edema in retinal fundus photographs. A 22 layers deep model called GoogLeNet as proposed by Lam et al. [22] also explored for DR detection. Further, ResNet architecture is also used for detection of DR [23]. Pratt et al. [31] proposed a deep learning-based CNN method with stochastic gradient descent by Nestrov momentum for DR detection. Shanthi and Sabeenian [32] also proposed a AlexNet architecture for DR detection.

3. Proposed methodology

A block diagram of the proposed technique is given in Fig. 1. The first part of the block diagram represents the data collection part, where a substantial number of high-determination retina images taken under an assortment of imaging conditions has been collected. After data collection, the data is divided into two parts; training set and testing set. HOG is used as feature descriptors of the images. After calculating HoG it has been observed that all the features are not equally important. Hence to extract a set of important features, Principal Component Analysis (PCA) is used. After feature extraction next step is the training phase of KNN algorithm. In the next step the data is processed by the KNN algorithm for prediction of their respective class labels (testing phase).

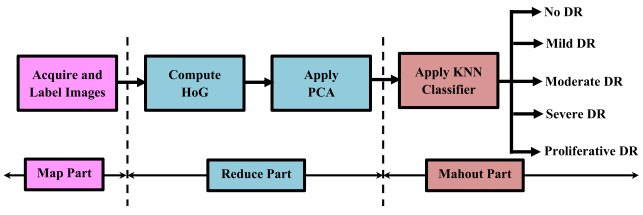


Fig. 1. Block diagram of the proposed scheme.

3.1. Acquisition of images

Every image of the Kaggle dataset [10] is having one subject id as well as either left or right tag (e.g. 1_left.jpeg is the left eye of patient id). An expert rated each image on a scale of 0 to 4 depending on the presence of diabetic retinopathy in it. The scale 0 to 4 signifies as: 0-No DR, 1-Mild, 2-Moderate, 3-Severe, 4-Proliferative DR. The visual appearance of left image vs right image may vary as the sources of images are different. Some of the images are collected as it can be seen normally, where macula is on the left eye and optic nerve is on the right eye. Rest of the images are collected as it can be seen through a microscope condensing lens, that is an inverted image. There are generally two ways to identify an inverted image:

1. If midline of optic nerve is at a slightly lower level than meculla (the small dark central area), then the image is inverted.
2. Absence of notch (square, triangle or circle) confirms that, the image is inverted. If there is a notch on the image, then that image is not inverted.

Like every real-world dataset, this dataset also has noise in images. The noise can be any artifact, images may be out of focus, underexposed, or overexposed. One of the prime objectives of this work to develop a robust algorithm which can handle these noise and variation.

3.2. Histogram of oriented gradients (HoG)

The histogram of oriented gradients (HOG) [33] is a feature descriptor of an image mainly used for the purpose of object detection [34]. Occurrence of gradient orientation in a localized portion of an image is computed using this technique. It enhances the accuracy by the use of overlapping local contrast normalization as compared to other similar techniques: edge orientation histograms [35], scale-invariant feature transform descriptors [36], and shape contexts [37]. The main idea behind HoG descriptor is the local object appearance and shape of an image. HoG suggests the distribution of intensity gradients or edge directions. Initially the image is divided into small blocks, after that a histogram of gradient directions is generated for each cell. Local histogram values can be contrast-normalized by calculating a measure of the intensity for a large region. To find the gradient values Dalal and Triggs [33] used 3×3 Sobel mask. To detect noise and complex backgrounds, normally local features are being used and to increase the DR detection accuracy global features are considered.

3.3. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [38] is an unsupervised feature extraction method for projecting high dimensional data onto a new lower dimensional space that preserves as much of the variance in the data as possible with minimum reconstruction error. Principal component analysis is a quantitatively rigorous method for achieving this simplification. The method generates a new set of variables, called principal components (PC). Each principal component is a linear combination of the original variables. All the principal components are orthogonal to each other, so that there will be no redundant information. The principal components as a whole form an orthogonal basis for the space of the data. PCs are calculated using the Eigen value decomposition of the data covariance matrix/correlation matrix or singular value decomposition of a data matrix. Usually after mean centring the data for each attribute covariance matrix is calculated when the variances of

variables are very high compared to correlation. It would be better to choose a type of correlation when the variables are of different types. Similarly the Singular Value Decomposition (SVD) [39] method is used for transforming correlated variables into a set of uncorrelated ones which better exposes the various relationships among the original data items. At the same time, SVD is a method for identifying and ordering the dimensions along which data points exhibit the most variation.

In other words PCA can be defined as an orthogonal linear transformation. It transforms the axes to have the maximum variance along the first coordinate (called the first principal component) [38], the second greatest variance on the second coordinate, and so on.

Let $X(n \times p)$ be a data matrix and can be represented as $X = [\mathbf{X}_{(0)} \ldots \mathbf{X}_{(p)}]$. Matrix X is having a special property called column-wise zero empirical mean. Each of the n rows represent a data point and each of the p columns correspond to the feature set of the data points. A set of p-dimensional vectors of weights, $[\mathbf{w}_{(k)} = (w_1, \ldots, w_p)_{(k)}]$ which normally represent a unit vector; map each of the data points or each row of the matrix X to a new vector of principal component scores $\mathbf{t}_{k(i)}$. This new vector of principal component scores can be obtained as $\mathbf{t}_{(i)} = (t_1, \ldots, t_p)_{(i)}$. This can be defined as $\mathbf{t}_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_{(k)}$.

3.4. KNN algorithm

In the next step, K Nearest Neighbors (KNN) [40], a very popular nonparametric classification algorithm is used to classify the images. In case of KNN classification, the output is a member of a predefined set of classes. Classification is done by the majority votes from its neighbors. The object is assigned to the class which gets maximum votes among its K nearest neighbors. Here K is typically very small value. If K = 1, then the object is assigned to a class having the nearest neighbor.

Let us consider a dataset comprising N_j points in class C_j with N points in total, so that $\sum_j N_j = N$. If we want to classify a new point u, we draw a sphere centered at u containing precisely k points irrespective of their classes. Suppose, this sphere has volume V and contains k_j points from class C_j, then density estimation of each class is

$$p(u|C_j) = \frac{k_j}{N_j V}. \tag{1}$$

Similarly, unconditional density is given by

$$p(u) = \frac{k}{NV}, \tag{2}$$

while the class prior probabilities are given by

$$p(C_j) = \frac{N_j}{N}. \tag{3}$$

We can now combine these three equations using Bayes' theorem to obtain the posterior probability of class probability as

$$p(C_j|u) = \frac{p(u|C_j)p(C_j)}{p(u)} = \frac{k_j}{k}. \tag{4}$$

If we wish to minimize the probability of misclassification, this is done by assigning the test point u to the class having the largest posterior probability, corresponding to the largest value of k_j/k. Thus to classify a new point, we identify the k nearest points from the training dataset and then assign the new point to the class having the largest number of representatives amongst this set [41].

KNN algorithm is one of the simplest machine learning algorithms. In KNN algorithm, the function is approximated locally and all computation is deferred until classification. KNN algorithm can be implemented by assigning weights to the contributions of the neighbors also. In this case the nearest neighbors can contribute more than the distant neighbors. For example the weight of each neighbor is considered as 1/d, where d is the distance between the neighbor and the object. Neighbors can be taken from each of the classes of training dataset also, though no explicit training step is required.

There are several advantages in using KNN against any other classification technique. This is easily implementable and found to be providing near optimal solution as $N \to \infty$. KNN is highly adaptive as it uses local information and is parallely implementable. Other classification schemes like Bayes classification will produce poor result if the input data are not having conditional independence or discission boundary is non-linear. KNN is non-parametric, and never makes any assumptions about the data distribution.

The success of the algorithm depends on the selection of the value of K. The value of K highly depends on the pattern of the data. A larger value of K helps in reducing the noise but at the same time it makes a boundary between less distinct classes in order to increase the number of classes. Different heuristic techniques are used to find the best value of K. Measuring distance between classes is an important aspect of KNN classifier. This distance can be measured using different techniques, such as Euclidean distance [42], Mahalanobis distance[43], Minkowski distance [44], Manhattan distance [45].

The steps of a KNN algorithm can be enumerated as below:

1. A set of training patterns are available.
2. A positive integer K is specified, along with a new sample.
3. Select the K entries from the database which are closest to the new sample.
4. Find the most common class of these entries based on majority.
5. This is the class label assigned to the new sample.

In the proposed scheme, we adhered to the Euclidean distance.

4. Hadoop framework for problem solving

Selection of platform to perform the experiment is one of the major steps to achieve the success of the experiment. While selecting the platform of the experiment two major factors were considered. First one is the amount of data to be processed. Second is how to get the best possible result. In Big Data platform a huge number of images can be used as training set, which increases the chance of accuracy improvement [46]. On the other hand Big Data platform is known to be efficient as compared to stand alone system. In the present context Hadoop framework [47] is used for Big Data analysis. The large dataset used in the present experiment has been handled using Haddop, MapReduce and Mahout framework [48].

Hadoop Distributed File System (HDFS) is a distributed file system. This file system can be run on commodity hardware. HDFS is highly fault-tolerant and can process large data. HDFS was originally built as an infrastructure for the Apache Nutch web search engine project [49]. HDFS is now an Apache Hadoop subproject.

HDFS is famous for its ability to support large dataset or Big Data. Typically it manages files of size gigabytes to terabytes. As it supports to process large datasets, it also needs to be scalable. It supports high bandwidth of data, hundreds of nodes in a single cluster and in a single instance tens of millions of files. To reduce data coherency and increase throughput HDFS provides write-once-read-many access for files to its applications. Web crawler and MapReduce applications fit perfectly in this model.

HDFS uses a master/slave architecture. An HDFS cluster consists of one NameNode. NameNode is typically the master in a master/slave architecture. It manages all nodes attached to it. These nodes are called DataNode. In an HDFS one NameNode and many DataNodes are present. HDFS is a namespace file system where users can store data in files. Each of the files is splitted into a number of blocks and these blocks are stored in DataNodes. All operations related to file system like opening, closing files are executed by the NameNode. The mapping of blocks to DataNode is also performed by NameNode. All file system's client related operations, block creation, deletion, and replication are done by the

DataNodes. The job tracker is the master daemon which runs on the same node that runs these multiple jobs on data nodes. The task tracker is the one that actually runs the task on the data node. The NameNodes and DataNodes are typically run in a Linux/GNU operating system. The Hadoop architecture is given in Fig. 2.

Hadoop uses MapReduce algorithm in order to achieve parallel processing across the cluster. It helps in moving the computation part to the data location. MapReduce algorithm is designed to compute large volumes of data by splitting it into various blocks and processing each block in a parallel fashion. It works on key and value pairs. MapReduce is having two major phases, Map phase and Reduce phase.

Reliability and fault tolerance ensured by replicating data across multiple hosts.

Mahout was earlier a different venture called "Taste" and has proceeded with improvement inside Mahout close by other Hadoop-based code. It might be seen as a fairly isolated, more thorough and more develop part of this code, contrasted with current improvement endeavors concentrating on Hadoop-based conveyed recommenders. The architecture for Mahout is given in Fig. 3.

4.1. MapReduce program for feature selection

In this experiment average size of each image is 60 MB and we are dealing with a large number of images, hence performing operations like HoG and PCA on all images are not possible using normal java progrmming paradigm. So Hadoop framework is used to manage a large amount of data. While performing HoG and PCA MapReduce prgramming paradigm is used.

In Mapper section of the program each image mapped with its corresponding class mentioned in separate (.csv) file, provided with dataset. As per MapReduce prgramming paradigm in mapper section, two important components are key and value. In our implementation keys are class labels of images and value is corresponding image data in bytearray format.

In reduce section all values are reduced to their respective classes. After that, HoG is performed on every image of each of the classes. To extract the most important features, PCA is applied on the output of HoG. In reduce section PCA and HoG algorithms are used as it can be used in a stand-alone system, but since it is used on HDFS it can handle a large amount of data.

4.2. KNN classification using Apache Mahout

To perform classification using the KNN classifier, four steps are followed. These steps are, 1) Create Sequence Files, 2) Convert Sequence Files to Vectors, 3) Training, and 4) Testing. Details of each of these steps are described in Appendix A.

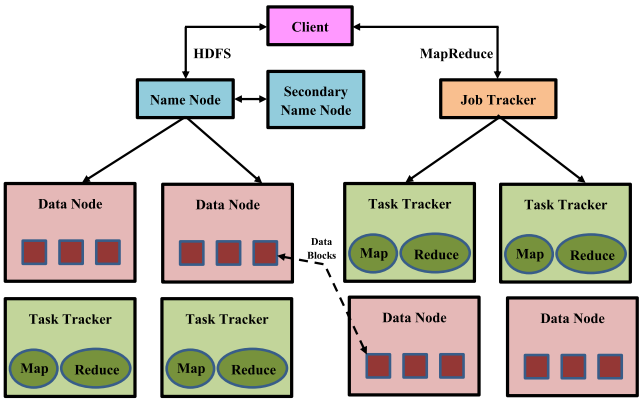


Fig. 2. Architecture of Hadoop framework.

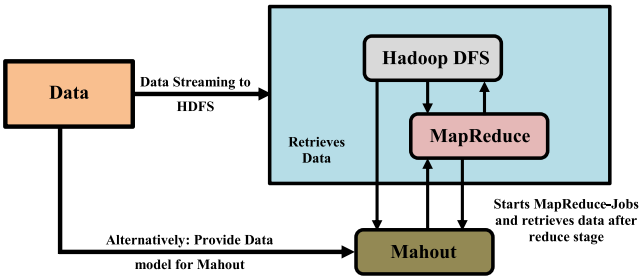


Fig. 3. Architecture of Mahout.

5. Results and discussions

The proposed scheme is implemented in Hadoop with MapReduce and Mahout framework. The code is run on Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz with 4 GB RAM and Ubuntu 14.04.4 operating system. Further, this section is devided into two parts: Experimental Results & Analysis and Discussions and Future works.

5.1. Experimental results and analysis

The proposed scheme is verified by performing experiments on three benchmark databases: Kaggle diabetic retinopathy detection data with four sets of images [10], DIARETDB0 [50] and Messidor-2 [51]. The performance of the proposed scheme is accessed on the considered databases by using three performance evaluation measures: Accuracy, Specificity, and Sensitivity. The performance of the proposed scheme is evaluated by comparing the results obtained by it with those of the seven state-of-the-art-techniques. This includes two conventional machine learning based five deep learning based techniques: Gulshan et al. [21], Lam et al. [22], Esfahani et al. [23], Pratt et al. [31], Kauppi et al. [50], Kamble et al. [27] and Shanthi et al. [32].

Accuracy of the algorithm is measured by using two indices Sensitivity and Specificity. Sensitivity of each class can be calculated from its $TP/(TP + FN)$ and specificity of each class can be calculated from its $TN/(TN + FP)$. Here TP denotes the true positive, TN denotes true negative, FP denotes false positive, and FN denotes false negative. For multi-class classification, one may use one against all approach. Suppose there are three classes: $C1$,$C2$, and $C3$. Here "TP of $C1$" is all $C1$ instances that are classified as $C1$, "TN of $C1$" is all non-$C1$ instances that are not classified as $C1$, "FP of $C1$" is all non-$C1$ instances that are classified as $C1$ and "FN of $C1$" is all $C1$ instances that are not classified as $C1$.

The Sensitivity and the Specificity are computed as:

$$Sensitivity = \frac{T_P}{T_P + F_N}; \tag{5}$$

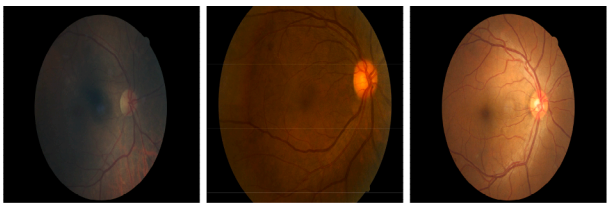
and

$$Specificity = \frac{T_N}{T_N + F_P}. \tag{6}$$

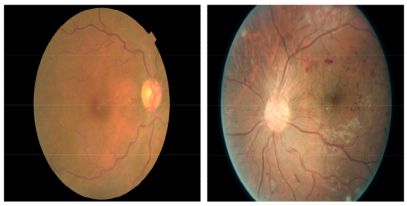
5.1.1. Experiments on Kaggle diabetic Retinopathy detection data

The size of each of the images is, on an average, 1.2 MB with a fixed dimension of (3888×2592). Two images (left eye and right eye) of each class are presented in Figs. 4, 5.

In the proposed scheme, using KNN algorithm each of the image test data is assigned to either of the classes: NDR, MDR, MoDR, SDR and PDR. If the prediction is matched with the original data then the prediction is correct otherwise the prediction is wrong. Based on the number of correctly predicted image data, accuracy of the entire model is determined. Total dataset consisting of test data and training data has 5334 images. Training dataset consists of 975 of the total training image data. Testing dataset consists of 4359 images. The division of training and testing data set are done as provided by the Kaggle dataset [10].

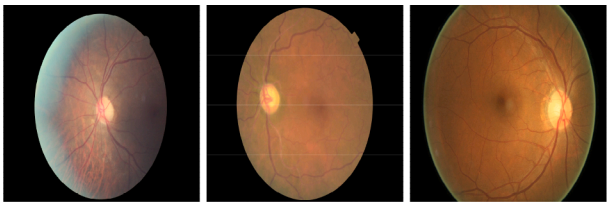


(a) No DR (b) Mild DR (c) Moderate DR

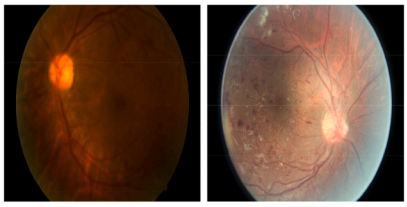


(d) Severe DR (e) Proliferative DR

Fig. 4. Images of right eyes.



(a) No DR (b) Mild DR (c) Moderate DR



(d) Severe DR (e) Proliferative DR

Fig. 5. Images of left eyes.

KNN algorithm is applied for different values of K (1,2,3). Table 1 gives the details about the different testing sets and correctly predicted data for different values of K. The details of predicted labels for each of the datasets (Set 1, Set 2, Set 3, Set 4) are given below. In Tables 2–5; the number of correctly and details of wrongly predicted data are mentioned in detail.

To find the four terms of $C2$ or $C3$ one can replace $C1$ with $C2$ or $C3$. In the following tables (Tables 6–9) TP, TN, FP, FN of each dataset (Set 1 to 4) is mentioned. The following tables (Tables 6–9) are called confusion matrix [52]. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class (or vice versa). Using the above confusion matrices Sensitivity and Specificity is measured. The following table (Table 10) depicts the Specificity and Sensitivity for different test sets (Set 1 to 4).

Table 1
Results using the proposed algorithm for different values of K in KNN algorithm.

Test set number	1	2	3	4
Number of images	275	1509	1535	1040
Cumulative Test set	275	1784	3319	4359
Accuracy (K = 1)	92.1%	84.5%	81.2%	78.16%
Accuracy (K = 2)	93.06%	84.5%	81.2%	78.16%
Accuracy (K = 3)	93.06%	85.01%	81.2%	78.16%

To test the effectiveness of the proposed scheme, we have compared the results obtained by the proposed scheme with several CNN based algorithms. In past few years CNN is one of the most popular deep learning techniques also used for the detection of diabetic retinopathy. Some of the very significant research on this subject are reported: [21,22,31,23]. In Table 11 the sensitivity and specificity of the proposed scheme results along with [21,22,31,23] are reported. It may be observed from this table, that the proposed scheme outperformed against [31,23] algorithms in terms of sensitivity and specificity. The proposed scheme provides lesser but comparable results with those of the [21,22] techniques. It is to be concluded that the proposed scheme with simple machine learning techniques can perform at par with those of even deep learning counter-parts.

5.1.2. Experiments on DIARETDB0 data

To validate the effectiveness of the proposed method, it is applied on DIARETDB0 [50] dataset. The result is compared with previously proposed methods mentioned in Tomi et al. [50] which also used DIARETDB0 dataset. It is required to mention here that use of big data analysis framework for retinopathy is very rare. We found that only Tomi et al's. [50] method may be regarded as a close competitive scheme of the proposed method. Here we have compared our result with that. In DIARETDB0 dataset there are 130 images. These images are divided into 5 image categories. From each category of image a fixed number of randomly selected images are taken and training (30%) set is prepared. Rest of the images are considered as test set. The diabetic retinopathy finding types that each image group contains are the following: i) Red small dots, haemorrhages, hard exudates, ii) Red small dots, haemorrhages, hard exudates, soft exudates, iii) Red small dots, haemorrhages, hard exudates, soft exudates, neovascularisation, iv) Red small dots, haemorrhages, soft exudates, neovascularisation, and v) Normal.

In Kauppi et al. [50] Sensitivity and Specificity are measured for four classes of images, namely Exudates, Red small dots, Haemorrhages and Exudates. In the present context the proposed algorithm is applied on the same dataset (DIARETDB0) to find Sensitivity and Specificity. The comparison of the results are reported in Table 12. It may be observed from Table 12, that the proposed scheme have outperformed the Tomi et al. [50] technique.

To test the effectiveness of the proposed scheme, we have compared the results obtained by the proposed scheme with those of the existing algorithm [50,27] techniques. All the results are reported in Table 13. It may be observed from this table, that the proposed scheme outperformed against both [50,27] algorithms in terms of sensitivity and specificity.

5.1.3. Experiments on Messidor-2 Data

The third database we have considered for experimental use is Messidor-2 [51]. The Messidor-2 is a collection of diabetic retinopathy images with two macula-centered eye fundus images (one per eye). It was captured with a Topcon TRC NW6 non-mydriatic fundus camera with a 45 degree field of view. The Messidor-2 databse contains 1748 retina images.

The performance of the proposed scheme is also evaluated on Messidor-2 database assuming 15% training and remaining as testing. The results obtained by the proposed scheme is compared aginst two state-of-the-art-techniques: Gulshan et al. [21] and Shanthi et al. [32]. The results obtained by all these techniques are reported in Table 14. It may be observed from this table that the proposed scheme provides a higher values of specificity and sensitivity against both Gulshan et al. [21] and Shanthi et al. [32] techniques.

5.2. Discussions and future works

Present experiment consists of a few steps: experimental environment set-up, preparation of training set and testing set, HoG calculation,

Table 2
Details of Prediction for dataset 1.

Data		Prediction		Wrong prediction for different classes				
	Test set	Correct	Wrong	NDR	MiDR	MoDR	SDR	PDR
NDR	1815	1562	253	0	132	49	52	20
MiDR	1276	884	392	292	0	75	16	9
MoDR	319	210	109	61	21	0	15	12
SDR	663	510	153	79	31	20	0	13
PDR	286	240	46	19	10	7	10	0

Table 3
Details of Prediction for dataset 2.

Data		Prediction		Wrong prediction for different classes				
	Test set	Correct	Wrong	NDR	MiDR	MoDR	SDR	PDR
NDR	1352	1121	231	0	114	56	41	20
MiDR	876	745	131	66	0	39	17	9
MoDR	441	335	106	56	26	0	13	11
SDR	409	301	108	49	28	22	0	9
PDR	221	193	28	11	5	7	5	0

Table 4
Details of Prediction for dataset 3.

Data		Prediction		Wrong prediction for different classes				
	Test set	Correct	Wrong	NDR	MiDR	MoDR	SDR	PDR
NDR	110	101	9	0	4	2	2	1
MiDR	56	52	4	1	0	2	18	4
MoDR	48	44	4	17	9	0	4	3
SDR	51	49	2	23	4	6	0	4
PDR	10	10	0	5	2	2	5	0

Table 5
Details of Prediction for dataset 4.

Data		Prediction		Wrong prediction for different classes				
	Test set	Correct	Wrong	NDR	MiDR	MoDR	SDR	PDR
NDR	793	688	105	0	47	24	26	8
MiDR	449	378	71	32	0	21	14	4
MoDR	252	211	41	18	12	0	9	2
SDR	193	162	31	15	11	2	0	3
PDR	97	78	19	8	7	2	2	0

Table 6
Confusion matrix for Set 1.

Test set type	True Positive	True Negative	False Negative	False Positive
NO DR	1562	239	451	253
Mild DR	884	749	194	392
Moderate DR	210	792	151	109
Severe DR	510	850	93	153
Proliferative DR	240	889	54	46

Table 7
Confusion matrix for Set 2.

Test set type	True Positive	True Negative	False Negative	False Positive
NO DR	1121	191	182	231
Mild DR	745	431	173	131
Moderate DR	335	480	124	335
Severe DR	301	528	76	108
Proliferative DR	193	555	49	28

Table 8
Confusion matrix for Set 3.

Test set type	True Positive	True Negative	False Negative	False Positive
NO DR	101	63	46	9
Mild DR	52	99	19	4
Moderate DR	44	106	12	44
Severe DR	49	89	29	2
Proliferative DR	10	106	12	0

Table 9
Confusion matrix for Set 4.

Test set type	True Positive	True Negative	False Negative	False Positive
NO DR	688	89	73	105
Mild DR	378	190	77	71
Moderate DR	211	218	49	211
Severe DR	162	216	51	31
Proliferative DR	78	250	17	19

Table 10
Sensitivity and Specificity for different test sets.

Test Set number	Data type	Sensitivity	Specificity
Set 1	NO DR	77.59%	48.57%
	Mild DR	82.03%	65.64%
	Moderate DR	58.17%	87.90%
	Severe DR	84.57%	84.74%
	Proliferative DR	81.63%	95.08%
Set 2	NO DR	86.03%	45.26%
	Mild DR	81.15%	76.69%
	Moderate DR	72.98%	58.89%
	Severe DR	79.84%	83.01%
	Proliferative DR	79.75%	95.19%
Set 3	NO DR	68.70%	87.5%
	Mild DR	73.23%	96.11%
	Moderate DR	78.57%	70.66%
	Severe DR	62.82%	97.80%
	Proliferative DR	45.45%	100%
Set 4	NO DR	90.40%	45.87%
	Mild DR	83.07%	72.79%
	Moderate DR	81.15%	50.81%
	Severe DR	76.05%	87.44%
	Proliferative DR	82.1%	92.93%

Table 11
Comparison of results on Kaggle database.

	Sensitivity	**Specificity**
Gulshan et al. [21]	97.5%	93.4%
Lam et al. [22]	95%	96%
Pratt et al. [31]	30%	95%
Esfahani et al. [23]	85%	86%
Proposed method	91.1%	86.07%

Table 12
Analysis of results on DIARETDB0 dataset.

	Existing method [50] **Sensitivity**	**Specificity**
Exudates	79%	58%
Red small dots	73%	70%
Haemorrhages	92%	75%
Exudates	77%	50%
	Proposed method	
Exudates	89.6%	71.2%
Red small dots	82.9%	83.1%
Haemorrhages	93.3%	89.2%
Exudates	82.6%	79.6%

Table 13
Comparison of results on DIARETDB0 database.

	Sensitivity	**Specificity**
Kauppi et al. [50]	80.25%	63.25%
Kamble et al. [27]	83.0%	43.0%
Proposed method	87.1%	80.77%

Table 14
Comparison of results on Messidor-2 data.

	Specificity	Sensitivity
Gulshan et al. [21]	93.90	96.10
Shanthi et al. [32]	97.45	96.35
Proposed method	97.62	96.42

feature selection, training and testing of the prediction model. Different segments of the used Hadoop architecture for the proposed technique is shown in Fig. 1. The present experiment is performed in a three node cluster. As per the Hadoop architecture one node is called name node and other two are called data nodes. Hadoop-2.2.0 is used in each of the nodes. As Hadoop is a parallel distributed platform, data processing can be done in a cluster of distributed systems. Replication factor is two. Block size is considered to be 128 MB. Hadoop can be implemented using commodity hardware. Commodity hardware suggests personal computers and laptops. In the present experiment, we have used three systems with configuration Intel(R) Core(TM) i7-3770 CPU 3.40 GHz with 4 GB RAM and Ubuntu 14.04.4 operating system.

Classification of the dataset is a major part of this experiment. To support machine learning algorithms in Hadoop framework Apache Mahout is used. For a successful prediction model, dataset plays a major role. Dataset is divided into two parts: training and testing set. Training dataset is used to train the prediction model. Hence training dataset is always a representative of all the classes of the data. On the other hand accuracy of prediction by the model is determined applying testing data on the model.

Histogram of oriented gradients (HoG) is implemented to find the feature descriptor. These feature descriptor find the features from each image data. Here HoG is used to find the global features of the images. In this process gradient of each pixel is calculated first. The range of direction of the gradient is $(-\frac{\pi}{2}, \frac{\pi}{2})$. This range is divided into sixteen bins with an interval of 11.25°. Every pixel is assigned to a particular bin depending on the gradient direction. For every bin, the number of pixels belonging to that bin is counted. This number of pixels to a particular bin is a component of the feature descriptor. HoG of rest of the training set and entire testing set is also calculated using the same method. In PCA computation step of the experiment, Eigen vectors are generated in Eigen space. This vector is used to transform the rest of the training and test data into Eigen space.

We also made a subjective quality assessment of the proposed scheme by considering the experts in the said area. Subjective image quality assessment is standardised by ITU recommendations [53]. It consists of two measures: (i) 'Mean Opinion Score' (MOS) and 'Differential Mean Opinion Score' (DMOS). The MOS is defined as the average score of the observer's opinion score between the reference image and the processed image. In the subjective quality assessment, we have taken into consideration 15 observers, 25 images are validated by each observer and the score is rated in the range from 0 to 100%. We have taken the help of different researchers working in the area of Diabetic Retinopathy and Ophthalmologist of the subjective assessment of the proposed scheme. All the 25 images taken from each databases are randomly chosen and experts are asked to score the results in 0 to 100%. The MOS of three data sets is given in Table 15. From the table, it is possible to observe that the proposed approach claims good subjective quality measures.

There are a few important things which have been addressed in this paper including Handling of Big Data, implementation of Hadoop, MapReduce and Mahout framework. Some scope of improvement is always there. KNN, as a classifier, is very simple to use but KNN first computes the distance and sort all the training data at each prediction, this process is very time consuming specially when it is being used for Big Dataset. The Big Data platform helped to reduce this time. Further research comparison with results using different feature selection and

Table 15
Subjective quality assessment.

S. No.	Data set	Number of observer	Number of images	Image type	Mean Opinion Score (MOS) in (%)
1.	Kaggle	15	25	Colour/ gray	86
2.	DIARETDB0	15	25	Colour/ gray	89
3.	Messidor-2	15	25	Colour/ gray	96

classification techniques could always be made.

6. Conclusions

In this article, we propose a Diabetic Retinopathy (DR) detection technique using the Hadoop framework. Hadoop framework is used to support large amount of data. In this experiment large set of high-resolution retina images taken under a variety of imaging conditions is used as input data. The dataset is divided into two parts, training set and testing set. Histograms of oriented gradients (HoG) are used to find the feature descriptors of the images. Here HoG is calculated as the global feature descriptor. All feature descriptors do not always carry significant information about the image, rather most of the time a subset of feature descriptors carry the most significant information. To extract the best subset of feature descriptors, Principal Component Analysis (PCA) is used. This subset of features is used for classification. KNN classifier is uesd to classify the dataset into five classes such as, No DR, Mild DR, Moderate DR, Severe DR and Proliferative DR.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. KNN classification using Apache Mahout

Step 1:Create Sequence Files

The input file format of Mahout is SequenceFile format. After completion of feature extraction process, the input file needs to be converted into SequenceFile format. SequenceFile is a hadoop class which allows representing data in key and value pair. In the present context, the key is set to features and value in its class for the training data. The following command is used:

```
mahout seqdirectory -i < input> -o < output>
```

Step 2: Convert Sequence Files to Vectors

In the next step the SequenceFile needs to be converted to vectors. Using seq2parse utility SequenceFile is converted into vectors.

The following command is used:

```
$MAHOUT_HOME/bin/mahout seq2sparse
-analyzerName (-a) analyzerName
The class name of the analyzer
-chunkSize (-chunk) chunkSize
The chunkSize in MegaBytes.
-output (-o) output
The directory pathname for o/p
-input (-i) input
Path to job input directory.
```

Step 3: Training

Using the trainnb utility, training process is performed in the next step. mahout trainnb.

The following command is used:

```
-i ${PATH_TO_TFIDF_VECTORS}
-el
-o ${PATH_TO_MODEL}/model
-li ${PATH_TO_MODEL}/labelindex
-ow
-c
```

Step 4: Testing

Using the testnb utility, testing process is performed in the next step. mahout testnb.

The following command is used:

```
-i ${PATH_TO_TFIDF_TEST_VECTORS}
-m ${PATH_TO_MODEL}/model
-l ${PATH_TO_MODEL}/labelindex
-ow
-o ${PATH_TO_OUTPUT}
-c
-seq
```

References

[1] S. Jayaraman, S. Esakkirajan, T. Veerakumar, Digital Image Processing, Tata McGraw Hill, 2013.
[2] A.B. Wolbarst, P. Capasso, A.R. Wyant, Medical Imaging: Essentials for Physicians, Wiley-Blackwell, 2013.
[3] A. Ghosh, Big data and its utility, Consulting Ahead 10 (2016) 52–69.
[4] D.A. Salz, A.J. Witkin, Imaging in diabetic retinopathy, Middle East Afr. J. Ophthalmol. 22 (2015) 145–150.
[5] World Health Organization diabetes, https://www.who.int/health-topics/diabetes, 2021. Accessed: 2021-25-03.
[6] J.W. Yau, S.L. Rogers, R. Kawasaki, E.L. Lamoureux, J.W. Kowalski, T. Bek, S. J. Chen, J.M. Dekker, A. Fletcher, J. Grauslund, et al., Global prevalence and major risk factors of diabetic retinopathy, Diabetes Care 35 (2012) 556–564.
[7] UK Prospective Diabetes Study (UKPDS) Group and others, Intensive blood-glucose control with sulphonylureas or insulin compared with conventional treatment and risk of complications in patients with type 2 diabetes (ukpds 33), 1998.
[8] D. Control, C.T.R. Group, et al., Progression of retinopathy with intensive versus conventional treatment in the diabetes control and complications trial, Ophthalmology 102 (1995) 647–661.
[9] H.A. Van Leiden, J.M. Dekker, A.C. Moll, G. Nijpels, R.J. Heine, L.M. Bouter, C. D. Stehouwer, B.C. Polak, Blood pressure, lipids, and obesity are associated with retinopathy: the hoorn study, Diabetes Care 25 (2002) 1320–1325.
[10] Kaggle, https://www.kaggle.com/c/diabetic-retinopathy-detection/data, 2015. Accessed: 2021-25-03.
[11] M.J. Cree, J.A. Olson, K.C. McHardy, P.F. Sharp, J.V. Forrester, A fully automated comparative microaneurysm digital detection system, Eye 11 (1997) 622.
[12] J. Hipwell, F. Strachan, J. Olson, K. McHardy, P. Sharp, J. Forrester, Automated detection of microaneurysms in digital red-free photographs: a diabetic retinopathy screening tool, Diabetic Med. 17 (2000) 588–594.
[13] R. Phillips, T. Spencer, P. Ross, P. Sharp, J. Forrester, Quantification of diabetic maculopathy by digital imaging of the fundus, Eye 5 (1991) 130.
[14] T. Spencer, J.A. Olson, K.C. McHardy, P.F. Sharp, J.V. Forrester, An image-processing strategy for the segmentation and quantification of microaneurysms in fluorescein angiograms of the ocular fundus, Comput. Biomed. Res. 29 (1996) 284–302.
[15] T. Spencer, R.P. Phillips, P.F. Sharp, J.V. Forrester, Automated detection and quantification of microaneurysms in fluorescein angiograms, Graefe's Archive Clin. Exp. Ophthalmol. 230 (1992) 36–41.
[16] G. Gardner, D. Keating, T.H. Williamson, A.T. Elliott, Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool, British J. Ophthalmol. 80 (1996) 940–944.
[17] B.M. Ege, O.K. Hejlesen, O.V. Larsen, K. Møller, B. Jennings, D. Kerr, D.A. Cavan, Screening for diabetic retinopathy using computer based image analysis and statistical classification, Comput. Methods Programs Biomed. 62 (2000) 165–175.
[18] S.C. Lee, E.T. Lee, R.M. Kingsley, Y. Wang, D. Russell, R. Klein, A. Warn, Comparison of diagnosis of early retinal lesions of diabetic retinopathy between a computer system and human experts, Arch. Ophthalmol. 119 (2001) 509–515.
[19] F.D. Verbraak, D.A. Michael, C.B. Gonny, K. Caroline, N. Giel, O.S. Reinier, A.v.d. H. Amber, Diagnostic accuracy of a device for the automated detection of diabetic retinopathy in a primary care setting, Diabetes Care 42 (2019) 651–656.
[20] R. Raman, S. Sangeetha, V. Sunny, S. Sobha, R. Chetan, R. Ramachandran, Fundus photograph-based deep learning algorithms in detecting diabetic retinopathy, Eye 33 (2019) 97–109.
[21] V. Gulshan, L. Peng, M. Coram, M.C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, et al., Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs, JAMA 316 (2016) 2402–2410.
[22] C. Lam, D. Yi, M. Guo, T. Lindsey, Automated detection of diabetic retinopathy using deep learning, AMIA Summits Translat. Sci. Proc. 2018 (2018) 147.
[23] M.T. Esfahani, M. Ghaderi, R. Kafiyeh, Classification of diabetic and normal fundus images using new deep learning method, Leonardo Electron. J. Pract. Technol 17 (2018) 233–248.
[24] J.L. Olson, N. Mandava, Fluorescein angiography, in: Retinal Imaging, Elsevier, 2006, pp. 3–21.
[25] R.T. St. Laurent, Evaluating agreement with a gold standard in method comparison studies, Biometrics (1998) 537–545.
[26] V.A. Josep, R.F. Dídac, A.Z. Miguel, X.M.G. Francesc, S.F. Oscar, Artificial intelligence for the detection of diabetic retinopathy in primary care: Protocol for algorithm development, JMIR Res. Protocols 8 (2019) e12539.

[27] V.V. Kamble, R.D. Kokate, Automated diabetic retinopathy detection using radial basis function, Procedia Comput. Sci. 167 (2020) 799–808.
[28] A.N. Rakhonde, P.R. Kshirsagar, S.M. Marve, Diabetes retinopathy disease detection using convolution neural network, Test Eng. Manage. (2020) 4431–4434.
[29] F. Arcadu, F. Benmansour, A. Maunz, J. Willis, Z. Haskova, M. Prunotto, Deep learning algorithm predicts diabetic retinopathy progression in individual patients, NPJ Digital Med. 2 (2019) 1–9.
[30] I. Qureshi, J. Ma, Q. Abbas, Diabetic retinopathy detection and stage classification in eye fundus images using active deep learning, Multimedia Tools Appl. (2021) 1–31.
[31] H. Pratt, F. Coenen, D.M. Broadbent, S.P. Harding, Y. Zheng, Convolutional neural networks for diabetic retinopathy, Procedia Comput. Sci. 90 (2016) 200–205.
[32] T. Shanthi, R. Sabeenian, Modified alexnet architecture for classification of diabetic retinopathy images, Comput. Electrical Eng. 76 (2019) 56–64.
[33] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, 2005.
[34] C. Vondrick, A. Khosla, T. Malisiewicz, A. Torralba, Hoggles: Visualizing object detection features, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1–8.
[35] D. Gerónimo, A. López, D. Ponsa, A.D. Sappa, Haar wavelets and edge orientation histograms for on–board pedestrian detection, in: Iberian Conference on Pattern Recognition and Image Analysis, Springer, 2007, pp. 418–425.
[36] D.G. Lowe, et al., Object recognition from local scale-invariant features., in: ICCV, volume 99, 1999, pp. 1150–1157.
[37] S. Belongie, J. Malik, J. Puzicha, Shape context: A new descriptor for shape matching and object recognition, in: Advances in Neural Information Processing Systems, 2001, pp. 831–837.
[38] Department of computer science at princeton university, pca, https://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pd, 2014. Accessed: 2021-25-03.
[39] S. Banerjee, A. Roy, Linear algebra and matrix analysis for statistics, Chapman and Hall/CRC, 2014.
[40] N.S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Am. Statist. 46 (1992) 175–185.
[41] C.M. Bishop, Pattern recognition and machine learning, Springer Science+ Business Media, 2006.
[42] M.M. Deza, E. Deza, Encyclopedia of distances, in: Encyclopedia of distances, Springer, 2009, pp. 1–583.
[43] Department of computer science at princeton university, mahalanobis metric, https://www.cs.princeton.edu/courses/archive/fall08/cos436/Duda/PR_Mahal/M_metric.htm, 2015. Accessed: 2021-25-03.
[44] Derrick lyndon pallas, minkowski metric, https://gist.github.com/pallas/5565528, 2018. Accessed: 2021-25-03.
[45] Wolfram research, inc., manhattan, http://mathworld.wolfram.com/TaxicabMetric.html, 1999. Accessed: 2021-25-03.
[46] F. Tekiner, J.A. Keane, Big data framework, in: 2013 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, 2013, pp. 1494–1499.
[47] P. Zikopoulos, C. Eaton, et al., Understanding big data: Analytics for enterprise class hadoop and streaming data, McGraw-Hill Osborne Media, 2011.
[48] The apache software foundation, mahout, http://mahout.apache.org, 2014. Accessed: 2021-25-03.
[49] A.L. 2.0, nutch, http://nutch.apache.org, 2004. Accessed: 2021-25-03.
[50] T. Kauppi, V. Kalesnykiene, J. Kamarainen, L. Lensu, I. Sorri, J. Pietila, H. Kalviainen, H. Uusitalo, Diaretdb0 - standard diabetic retinopathy database (2007). Accessed: 2021-25-03.
[51] E. Decenciére, X. Zhang, G. Cazuguel, B. Lay, B. Cochener, C. Trone, P. Gain, R. Ordonez, P. Massin, A. Erginay, B. Charton, J.C. Klein, Feedback on a publicly distributed database: the messidor database, Image Anal. Stereol. 33 (2014) 231–234.
[52] H. Hamilton, Confusion, http://www2.cs.uregina.ca/dbd/cs831, 1998. Accessed: 2021-25-03.
[53] ITU-R Rec. BT. 500: methods for the subjective assessment of the quality of television pictures, 2012.