Detection and description generation of diabetic retinopathy using convolutional neural network and long short-term memory

R Amalia, A Bustamam*, and D Sarwinda
Department of Mathematics, FMIPA Universitas Indonesia, Kampus UI Depok, Depok 16424, Indonesia

*alhadi@sci.ui.ac.id

Abstract. Diabetic Retinopathy (DR) is one of the eye diseases suffered by diabetes patients that will cause blindness if it does not get effectively treated for a certain period of time. Early detection is needed to help patients get effective treatment based on their severity. Researchers have done copious amounts of research regarding the methods for DR detection using shallow learning and deep learning approaches. The proposed method in this paper is a combination of two deep learning architectures, namely Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). CNN is used to detect lesions on retinal fundus images, and LSTM is used for generating description sentences based on those lesions. In the training and testing process, the CNN output will be used for the input of LSTM. The training process’s target is to produce a model that can map retinal fundus images into a sentence. The results of this experiment using the MESSIDOR data set has an accuracy of around 90%.

1. Introduction
Diabetes mellitus (DM) is a disease characterized by metabolic disorders caused by deficiency or resistance insulin [1]. DM can cause complicated diseases, such as heart diseases, renal problems, and Diabetic Retinopathy (DR) [2]. DR is one of the eye diseases suffered by DM patients that will cause blindness if it does not get effectively treated for a certain time [3]. Early detection is needed to help patients get effective treatment based on their severity.

Research regarding methods for DR detection has been done numerous times with good results. There are a couple of approach methods for DR detection, such as shallow learning and deep learning. The approaches mentioned previously are machine learning methods that are differentiated by with and without extraction feature [4,5]. Sarwinda et al. [6] used shallow learning for DR detection with Complete Local Binary Pattern as feature extraction and K-Nearest Neighbor as a classifier. Qureshi et al. [7] compared the performance accuracy of shallow learning and deep learning in DR detection using the same data set. The results show that the deep learning approach obtained a higher accuracy performance. Deep learning has various architectures, including Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). Athira et al. [8] used CNN for detection and classification of DR into two classes, non-DR and DR.

The output of the previous work is the DR class. We propose a method that will produce the output in a simple sentence describing what is contained in the retinal fundus image. If the output is only DR class, the radiologist does not know the condition of the lesion in the patient’s eye. The output with a description sentence explaining the condition in the retinal fundus image is needed, thus helping the radiologist as a consideration in diagnosis the class of DR. Recently, researchers combined two deep learning architectures to generate a description sentence based on the condition in the images. Vinyals et al. [9] used CNN for object detection in images and used Long Short-Term Memory (LSTM) as a
model of RNN for generated a description sentence based on the condition of those objects. In this paper, we will use CNN to detect lesions in retinal fundus images and use LSTM to generate a description sentence based on the condition of those lesions. The output is not a DR class but also a description sentence of the condition of the lesions in the retinal fundus images.

2. Method

2.1 Preprocessing Data

2.1.1 Preprocessing Retinal Fundus Images

The first step in preprocessing retinal fundus image is cropping each retinal fundus image so that the black background in the fundus image is not too dominant. After cropping the retinal fundus images, the next step is image enhancement using Histogram Equalization (HE). HE is used to producing necessary contrast to the fundus image, thus making the lesions easier to detect [10]. The last step is to rotate each fundus image at a 90°, 180°, and 270° counterclockwise.

2.1.2 Creating Captions for The Training Process

Each retinal fundus image used for the training process is manually captioned based on the existing label on the data set. One retinal fundus image will be captioned with three description sentences that have a similar meaning, as in Table 1.

| Fundus image | Manual caption |
|--------------|---------------|
| ![Fundus Image](image1.png) | 1. This is a normal fundus image.  
2. A healthy fundus image.  
3. This is healthy fundus image. |
| ![Fundus Image](image2.png) | 1. A fundus image with microaneurysms, hemorrhages, and neovascularization.  
2. There are microaneurysms, hemorrhages, and neovascularization.  
3. A fundus image has microaneurysms, hemorrhages, and neovascularization. |

Source: http://www.adcis.net/en/third-party/messidor/

2.2 CNN

CNN is a deep learning architecture that is commonly used for spatial data, such as an image [11]. CNN consists of three main layers, namely the convolutional layer, the pooling layer, and the fully connected layer, as shown in Figure 1. The input image will be extracted in the convolutional layer by convoluting a small area in the input image with a matrix filter [12]. After that, the convolution layer’s result will be transformed by a non-linear operation, namely the Rectified Linear Unit (ReLU). With ReLU, all negative values in the output of the convolutional layer is mapped by zero.

In the pooling layer, the input image size will be reduced by an operation without removing any important information [12]. There are two types of operations at the pooling layer: average pooling and max pooling, respectively based on the average and the largest value of the small area in the input image [13]. The last layer of the convolutional or pooling layer will be changed to become one dimension, and then it will be classified in the fully connected layer [12]. In the classification task, the output size of CNN is equal to the number of classes.
The CNN architecture has various models such as LeNet, AlexNet, ZFNet, VGGNet, GoogleNet. These models are differentiated by the number of convolutional, pooling, and fully connected layers [14]. Furthermore, the parameter sizes, included kernel size, stride, and zero padding, that have been used at each model are different [14]. For example, LeNet consists of three convolutional layers, two pooling layers, and two fully connected layers. Meanwhile, GoogleNet consists of three convolutional layers, nine inception modules, five pooling layers, and one fully connected layer. The inception module is several convolutional layers that have been concat.

2.3 LSTM

Deep learning has an architecture that is used for sequential data, namely RNN [15]. One of the various models of RNN is LSTM, which has a cell memory and three gates in the hidden layer [16]. Let suppose $x_1, x_2, ..., x_T$ is the input of LSTM, where $x_t$ is an input at time step $t$. The important information at each time step $t$ will be stored in cell memory $C_t$. The three gates in the hidden layer of the LSTM are the forget gate $\alpha(t)$, input gate $\beta(t)$, and output gate $\gamma(t)$, as shown in Figure 2.

The first step in the hidden layer of LSTM is will determine how much information in the previously hidden layer will be store in cell memory $C_t$. This decision was decided at the forget gate with a formula as follows [17]:

$$\alpha(t) = \sigma(W_x x_t + W_h h_{t-1})$$  \hspace{1cm} (1)

where $W_x$ and $W_h$ are the weight for the forget gate. How much the new input $x_t$ will be stored in the cell memory $C_t$ will be calculated at the input gate with the following formula [17]:

$$\beta(t) = \sigma(R_x x_t + R_h h_{t-1})$$  \hspace{1cm} (2)

where $R_x$ and $R_h$ are the weight for the input gate. After we get new information, we update cell memory $C_t$ with a formula as follows [17]:

$$C_t = C_{t-1} \cdot \alpha(t) + \beta(t) \cdot \tanh (p_x x_t + p_h h_{t-1})$$  \hspace{1cm} (3)
where \( P_x \) and \( P_h \) are the weight for the cell memory. The last step is to determine how much information in cell memory \( C_t \) will be used to compute the output of hidden layer \( t \) with the following formula [17]:

\[
h_t = \sigma (Q_x x_t + Q_h h_{t-1}) \cdot \tanh (C_t)
\]

(4)

where \( Q_x \) and \( Q_h \) are the weight for the output gate.

2.4 Model

As shown in Figure 3, the first step in the training process is inputting the retinal fundus image that has been through preprocessing into CNN. The CNN model that is used in this paper is GoogLeNet. The output of CNN is image features, which is a summary of what is contained in the retinal fundus image. The image features in the form of the vector, such as in a fully connected layer, as shown in Figure 1. These image features will be inputted into LSTM along with a word in the description sentence.

LSTM maps the image features to a word, and the word will be mapped again to the next word until the words form a sentence. As shown in the green area in Figure 3, the image features are mapped to the word “this”, and the word “this” maps to the word “is” and so on until getting the sentence “this is healthy fundus image”. During the training process, LSTM is trained to produce a model that can map image features into a sentence. The model obtained from the training process is evaluated in the testing process by inputting each fundus image into the model.

![Figure 3](image.png)

Figure 3. A diagram plot of the model. The blue area is the image feature extraction process, and the green area is the description generation process based on image features.

3. Result and Discussion

3.1 Data

This paper uses 72 retinal fundus images from the MESSIDOR data set, which includes 36 normal fundus images and 36 DR fundus images. Each retinal fundus image is in a .tif format and is at the size of 1440x960, width and height respectively. The data set is divided into 80% training data and 20% testing data. The training data is used in the training model, and then the model is evaluated using testing data. Model evaluation is done by measuring accuracy, i.e., the percentage of data that is predicted to be correct by the model.

3.2 Experiment and Result

The training process is done 50 epochs, which means the model will learn using the training data 50 times. This paper is used Adam as an optimizer with a learning rate of 0.001 and a batch size of 30.
The training and testing process was done three times with different data, so that each model accuracy is obtained as in Table 2.

| Model | Accuracy (%) |
|-------|--------------|
| 1     | 89.65        |
| 2     | 89.65        |
| 3     | 91.37        |
| Mean  | 90.22        |

As shown in Table 2, the first and second models achieved the same accuracy, i.e., 89.65%. The highest accuracy was achieved from the third model, with an accuracy of 91.37%. The example output using testing data can be seen in Table 3. The output in number 1 and number 2 is an example of the correct output because it does match the ground-truth. Meanwhile, the output in number 3 and number 4 is an example of incorrect output because it does not match the ground-truth.

### Table 3. Example of the output.

| No. | Input | Output | The ground-truth |
|-----|-------|--------|------------------|
| 1   | ![Input Image](image1.png) | A fundus image with microaneurysms, hemorrhages, and neovascularization. | DR fundus image |
| 2   | ![Input Image](image2.png) | Normal fundus image. | Normal fundus image |
| 3   | ![Input Image](image3.png) | A fundus image contains microaneurysms, hemorrhages, and neovascularization. | Normal fundus image |
| 4   | ![Input Image](image4.png) | There are microaneurysms, hemorrhages, and neovascularization. | Normal fundus image |

### 4. Conclusion
Detection and description generation of DR using two deep learning architectures, i.e., CNN and LSTM, have been done in this paper with an accuracy of around 90%. The description sentence obtained from the model can help radiologists as a consideration in the diagnosis of the class of DR. In further research, training data with more classes of DR can be used, such as mild non-proliferative, moderate non-proliferative, severe non-proliferative, and proliferative. So, the results are not only normal and DR, but also their severity of the disease.

### Acknowledgments
This research supported by Publikasi Terindeks Internasional Sains Teknologi dan Kesehatan (PUTI SAINTEKES Q4) 2020 from Universitas Indonesia (NKB-2395/UN2.RST/HKP.05.00/2020).
References

[1] Mookiah M R K, Acharya U R, Chua C K, Lim C M, Ng E.Y.K, and Laude A 2013 Computer Aided Diagnosis of Diabetic Retinopathy: A Review *Computers in Biology and Medicine* **43** 2136-2155

[2] Faust O, Acharya R, Ng E.Y.K, Ng Kwan, and Suri J 2012 Algorithms for the Automated Detection of Diabetic Retinopathy Using Digital Fundus Images: A Review *Journal of Medical Systems* **36** 145–157

[3] Romero-Aroca P, Navarro-Gil R, Valls-Mateu A, Sagarrà-Alamo R, Moreno-Ribas A, and Soler N 2017 Differences in Incidence of Diabetic Retinopathy Between Type 1 and 2 Diabetes Mellitus: A Nine-year Follow-up Study *Ophthalmol* **101** 1346-1351

[4] Beltramelli T and Risi S 2015 Deep-Spying: Spying Using Smartwatch and Deep Learning *Master Thesis* (Denmark: IT University of Copenhagen)

[5] Priya H A G, Anitha J, and Rani J N 2018 Computer Aided Diagnosis Methods for Classification of Diabetic Retinopathy Using Fundus Images *IEEE International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET 2018)* 1-4

[6] Sarwinda D, Bustamam A, and Wibisono A 2017 Complete Modelling of Local Binary Pattern for Detection of Diabetic Retinopathy *IEEE 1st International Conference on Informatics and Computational Sciences (ICICoS 2017)* 7-10

[7] Qureshi I, Ma J, and Abbas Q 2019 Recent Development on Detection Methods for the Diagnosis of Diabetic Retinopathy *Symmetry* **11** 1-34

[8] T R Athira, Sivadas A, George A, Paul A, and Gopan N R 2019 Automatic Detection of Diabetic Retinopathy Using R-CNN *International Research Journal of Engineering and Technology* **6** 5595-5600

[9] Vinyals O, Toshev A, Bengio S, and Erhan D 2017 Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge *IEEE Transactions on Pattern Analysis and Machine Learning Intelligence* **39** 652-663

[10] Rehman M, Khan S H, Abbas Z, and Rizvi S M D 2019 Classification of Diabetic Retinopathy Images Based on Customised CNN Architecture *Amity International Conference on Artificial Intelligence (AICAI 2019)* 244-248

[11] Maggiori E, Tarabalka Y, Charpiat G, and Alliez P 2016 Convolutional Neural Networks for Large-Scale–Remote-Sensing Image Classification *IEEE Transactions on Geoscience and Remote Sensing* **55** 645-657

[12] Wang X, Lu Y, Wang Y, and Chen W B 2018 Diabetic Retinopathy Stage Classification using Convolutional Neural Networks *IEEE 19th International Conference on Information Reuse and Integration for Data Science (IRI 2018)* 465-471

[13] Bejiga M B, Zeggada A, Nouffidj A, and Melgani F 2017 A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery *Remote Sensing* **9** 1-22

[14] Sultana F, Sufian A, and Dutta P 2018 Advancements in Image Classification using Convolutional Neural Network *IEEE 4th International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN 2018)* 122-129

[15] Wang Y, Zhang D, Liu Y, Dai B, and Lee L H 2019 Enhancing Transportation Systems via Deep Learning: A Survey *Transportation Research Part C* **99** 144-163

[16] Graves A 2012 *Supervised Sequence Labelling with Recurrent Neural Networks* (Canada: Springer)

[17] Huang C J, and Kuo P H 2018 A Deep CNN-LSTM Model for Particulate Matter (PM2.5) Forecasting in Smart City *Sensor* **18** 1-22