



Available online at <http://scik.org>

Commun. Math. Biol. Neurosci. 2021, 2021:59

<https://doi.org/10.28919/cmbn/5832>

ISSN: 2052-2541

DIABETIC RETINOPATHY DETECTION AND CAPTIONING BASED ON LESION FEATURES USING DEEP LEARNING APPROACH

RIZKA AMALIA¹, ALHADI BUSTAMAM^{1,*}, ANGGUN RAMA YUDANTHA², ANDI ARUS VICTOR²

¹Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Indonesia, Depok 16424,
Indonesia

²Department of Ophthalmology, Faculty of Medicine, Universitas Indonesia Cipto Mangunkusumo National General
Hospital, Jakarta Pusat 10430, Indonesia

Copyright © 2021 the author(s). This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract: Diabetic Retinopathy (DR) can lead to vision loss if the patient does not get effective treatment based on the patient's condition. Early detection is needed to know what an effective treatment for those patients is. For helping ophthalmologists, DR detection methods using computer-based were developed. Ophthalmologists can use the result of the method as a consideration in diagnosing the class of DR. One of the powerful methods is deep learning. The proposed method uses two deep learning architectures, namely Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), for DR detection. CNN is used to detect DR lesion features, and RNN is used for captioning based on those lesion features. We used three pre-trained CNN models, including AlexNet, VGGNet and GoogleNet, and used Long Short-Term Memory (LSTM) as RNN models. In the image preprocessing, we applied contrast enhancement using Contrast Limited Adaptive Histogram Equalization (CLAHE) and compared the results with those without CLAHE. We have done the training and testing process with a different proportion of data. The experimental

*Corresponding author

E-mail address: alhadi@sci.ui.ac.id

Received April 9, 2021

results show that our proposed method can detect the lesion features and generate caption with the highest average accuracy of 96.12% for GoogleNet and LSTM with CLAHE and the proportion 70% training data 30% testing data.

Keywords: diabetic retinopathy; deep learning; convolutional neural network (CNN); long short-term memory (LSTM).

2010 AMS Subject Classification: 92B20.

1. INTRODUCTION

Diabetic Retinopathy (DR) is a complication of diabetes caused by uncontrolled high blood sugar levels in the long-term [1]. DR could lead to vision loss if the patient were untreated properly based on the patient's condition [2]. Early detection is required to prevent permanent vision loss in patients [3]. According to [4], early detection and treatment can reduce vision loss's impact up to 95%. One of the methods to detect DR is through a retinal fundus image's patient. Ophthalmologists identify the retinal fundus image whether there are DR lesion features, such as microaneurysms, exudates, hemorrhages, and neovascularization [5]. Since those processes take a time, researchers developed computer-based methods to speed up the DR detection process [6].

There are various computer-based methods for DR detection, and one of the methods is deep learning. Deep learning has a higher accuracy performance than other DR detection methods [7]. Deep learning consists of hierarchically structured layers that can automatically extract features from complex and big data, such as images or text [8]. Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) are two examples of deep learning architecture that are designed for spatial and sequential data, respectively. CNN is mostly used for DR detection because it has a convolutional layer that can extract informative features of retinal fundus images well [8, 9]. There are numerous CNN models for DR detection, both self-constructed models that trained from scratch and pre-trained models, such as LeNet, AlexNet, VGGNet, GoogleNet, ResNet and DenseNet. Those models are differentiated based on the number of layers and the size of parameters. The pre-trained CNN model is a CNN model suitable for training new small dataset because it can overcome the overfitting problem in the new model [9].

As reviewed by Alyoubi et al. [8] and Asiri et al. [9], many researchers use deep learning, especially CNN, for DR detection. For instance, Athira et al. [10] constructed the CNN with two convolutional layers for DR detection. Using self-constructed CNN, Athira et al. were able to detect exudates and red lesions in the retinal fundus image also classified those fundus images into normal (without DR) and abnormal (with DR). Athira et al. get an accuracy of 93.8%. Wang et al. [11] constructed a region-based CNN that can detect and classify DR along with locating the lesion features regions. Wang et al. were able to detect tiny DR lesion features in the retinal fundus image and classified those fundus images into healthy, mild, moderate, severe, and proliferative, with an accuracy of 92.95%.

Some researchers use the pre-trained CNN model to detect DR lesion features. Esfahani et al. [12] use a pre-trained CNN model for DR lesion features detection and classified them based on those lesion features into normal and DR. Esfahani et al. used ResNet and gets an accuracy of 86%. Raj et al. [13] use VGGNet to detect blood vessels and microaneurysms, including classifying them into normal, mild, moderate, severe, and proliferative. Raj et al. get an accuracy of 95.41%.

To increase the model's performance accuracy, some researchers applied a contrast enhancement method on preprocessing the retinal fundus images. Contrast Limited Adaptive Histogram Equalization (CLAHE) is one of the contrast enhancement methods that often used for retinal fundus images. CLAHE is a histogram-based method that can produce a satisfactory contrast of retinal fundus image, and because of that, the DR lesion features detected clearly [14, 15]. Kaur and Mann [16] applied CLAHE to their retinal fundus images before processing them to the model for detecting DR based on blood vessels. As a result, blood vessels are more apparent, and they get a model accuracy of 96.17%. Soomro et al. [14] compared the results of blood vessels detection through image preprocessing with CLAHE and image preprocessing without CLAHE. As a result, the model through image preprocessing with CLAHE is higher than without CLAHE.

The output of the DR detection that has been widely carried out using CNN is a DR class. In case the output is only class of DR, the ophthalmologist does not know the retinal fundus image patient's condition. The ophthalmologist needs the output in the form of a caption that explains the condition present in the retinal fundus image. Those captions help the ophthalmologist as a

consideration in diagnosing the class of DR.

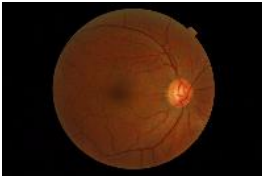

CNN, along with RNN, is usually used for image captioning. CNN is used to detect objects in the image, while RNN is used to generate captions based on those objects [17]. RNN has a hidden layer that can save the previous layer's output and use it for the next layer so that RNN can predict well next word in captioning [18]. Although RNN can predict well the next word, RNN can not overcome long-term dependence [17]. Long Short-Term Memory (LSTM) is one of the RNN models with a memory cell that can overcome those issues. The memory cell allows LSTM to store information and context in case of a long caption [19].

Related to DR detection, CNN can detect the DR lesion features with good results. Applying CLAHE as a contrast enhancement method on retinal fundus images before inputting those fundus images to the model can improve the model's performance accuracy. The aim of this study is to determine the results and performance accuracy of CNN and LSTM in detecting DR lesion features and generating captions based on these lesion features. Besides, it is also to determine the effect of applying CLAHE to our retinal images on the model's performance accuracy.

2. MATERIALS AND METHODS

2.1 Dataset

Table 1. Example of retinal fundus images with captions.

Retinal Fundus Image	Captions
	<ol style="list-style-type: none"> 1. Normal fundus image. 2. A healthy fundus image. 3. A fundus image without Diabetic Retinopathy disease characteristics.
	<ol style="list-style-type: none"> 1. This fundus image has microaneurysms, hemorrhages, and neovascularization. 2. A fundus image with microaneurysms, hemorrhages, and neovascularization. 3. There are microaneurysms, hemorrhages, and neovascularization in this fundus image.

Source: <http://www.adcis.net/en/third-party/messidor/> [20]

A total of 54 normal and 54 DR fundus images from the MESSIDOR dataset were used in this study. We limited this study using only three lesion features based on the MESSIDOR dataset's information, i.e. microaneurysms, hemorrhages and neovascularization. Each retinal fundus image has captions describing what lesion features are present on those retinal fundus images. One retinal fundus image has three captions with a similar meaning, as shown in Table 1. Each retinal fundus image will be flipped horizontally and vertically, thus becoming 324 fundus images.

2.2 CNN

CNN has three main layers that are the convolutional, pooling and fully connected layers. The convolutional layer will extract the input image features by convolving small areas in the input image with a matrix-shaped filter, as shown in Figure 1 [21]. There are adjustable parameters, such as filter and stride. For example, given a filter to convolute small areas in the q -th layer with the size $F_q \times F_q$, the weights of the filter are represented by

$$W^{(q)} = [w_{ij}^{(q)}] \quad (1)$$

where i and j are the positions along with the height and width, respectively. The result of the q -th convolutional layer is obtained using the following formula [21]:

$$h_{ij}^{(q+1)} = \sum_{r=1}^{F_q} \sum_{s=1}^{F_q} w_{rs}^{(q)} h_{i+r-1, j+s-1}^{(q)} \quad (2)$$

with $i = 1, \dots, L_q - F_q + 1$ and $j = 1, \dots, B_q - F_q + 1$, where L_q and B_q are size of q -th layer. Meanwhile, the stride is a parameter that determines the amount of convolution shift. As shown in Figure 1, given a stride equal to 1, it means that a convolution will be carried out every one shift.

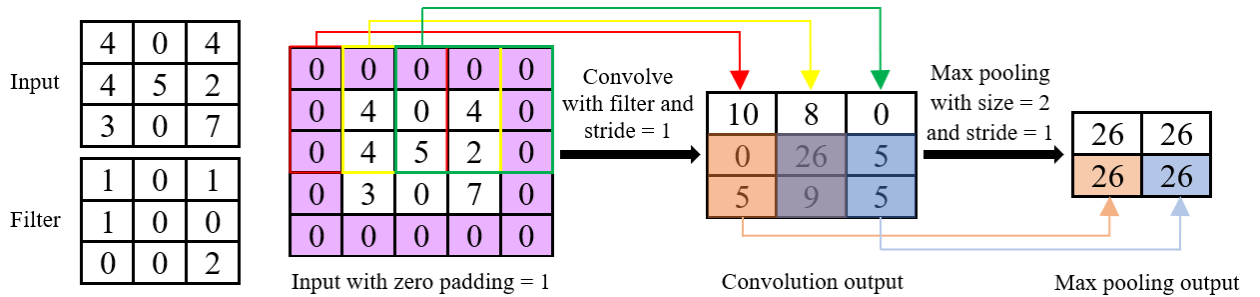


Figure 1. Example computation in the convolutional and pooling layer. Convolve input with filter size 3×3 , stride 1, and zero padding 1. Then pooled the convolution's output with max pooling size 2×2 and stride 1.

Besides filter and stride, there is another parameter that can be adjusted, namely zero padding. Zero padding is a parameter that determines the number of pixels containing the number 0 to be padded to the input border, which makes the output size of the convoluted layer not much different from the input size [22]. Zero padding is equal to 1, which means that one pixel containing the number 0 in the input border will be added, as shown in Figure 1. The convolutional layer's output will be transformed with a non-linear operation, namely Rectified Linear Unit (ReLU), using the following formula [23]:

$$ReLU(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{else} \end{cases} \quad (3)$$

The pooling layer reduces the size of the convolutional layer's output by a math operation while retaining important information from the image [24]. Average and max pooling are types of operation in the pooling layer. Both are based on the average and the largest value of the small area in the image, respectively. Similar to the convolutional layer, the pooling layer also has a stride. Example operation for max pooling with the size is equal to 2 and stride is equal to 1 shown in Figure 1.

After adding the appropriate number of convolutional and pooling layers, the next layer is fully connected. In the fully connected layer, the dimensions of the last layer of the convolutional or pooling layer will be converted into one dimension, which will be further processed using Feed Forward Neural Network [25]. The output size from CNN is based on the task to be solved. If the classification task, then the output size is the number of classes.

Many CNN models have been widely applied for DR detection. Since this study uses small data set, so we use pre-trained CNN model. This study uses three pre-trained CNN models through transfer learning, i.e. AlexNet, VGGNet and GoogleNet. Transfer learning is the process of using the weight of a large data set to initialize the new model's weight, followed by finetuning the new model's weight on the new data set [25]. The large data set we used for transfer learning is the ImageNet data set.

2.2.1 AlexNet

AlexNet with input image size 227×227 is CNN which does not have layers that are too deep.

According to [26], AlexNet has a better computational capability of handling complexity than other models. AlexNet architecture shown in Figure 2 (a). Filter sizes used in the convolutional layer are 11×11 with stride 4 and zero padding 2, 5×5 with stride 1 and zero padding 2 and 3×3 with stride 1 and zero padding 1. AlexNet uses max pooling size equal to 3 with stride 2. Between three fully connected layers, there are two dropout layers. The dropout layer is setting the neuron's output of a fully connected layer to zero [27]. The dropout layer does not contribute to the feed-forward process [28]. For AlexNet, each dropout layer's size is equal to 0.5, which means 50% of the neuron's output of the fully connected layer will be set to zero. The number of neurons in the first fully connected layer is 9,216 and the second and the last is 4,096.

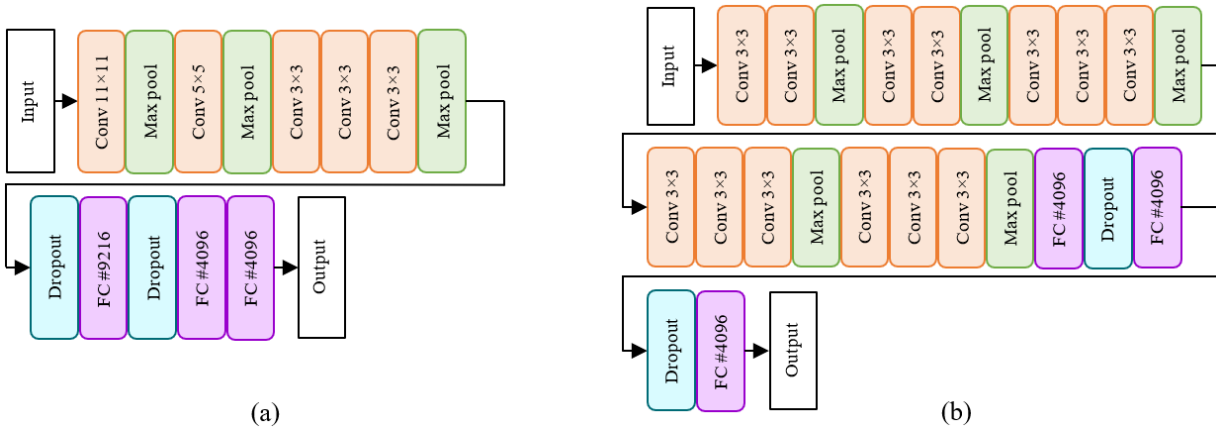


Figure 2. (a) AlexNet architecture, (b) VGGNet architecture.

2.2.2 VGGNet

VGGNet with an input image size 224×224 is a CNN that has a deeper layer than AlexNet. VGGNet performs very well with the image that has dense features, such as the retinal fundus image [13]. VGGNet architecture shown in Figure 2 (b). VGGNet uses a 3×3 filter with stride 1 and zero padding 1 for all convolutional layers. For all pooling layers, VGGNet uses max pooling with size 2 and stride 2. Similar to AlexNet, between three fully connected layers, there are two dropout layers. The size of each dropout layer is 0.5. The number of neurons for all fully connected layers is 4,096.

2.2.3 GoogleNet

GoogleNet has a modify layer called inception module. The inception module is concatenating

of several convolutional channels and a max pooling channel, as shown in Figure 3 (a). The inception module allows GoogleNet to extract different features of the same image in parallel computation; hence, GoogleNet performs well even with limited memory and computing budgets [28, 29]. For the whole GoogleNet architecture can be seen in Figure 3 (b). In the convolutional layer, GoogleNet uses filter sizes 7×7 with stride 2 and zero padding 3, 3×3 with stride 1 and zero padding 1, and 1×1 with stride 1. GoogleNet uses max pooling with size 2 and stride 2 and average pooling with size 7. GoogleNet uses nine inception modules. The size of the dropout and fully connected layers are 0.2 and 1,024, respectively.

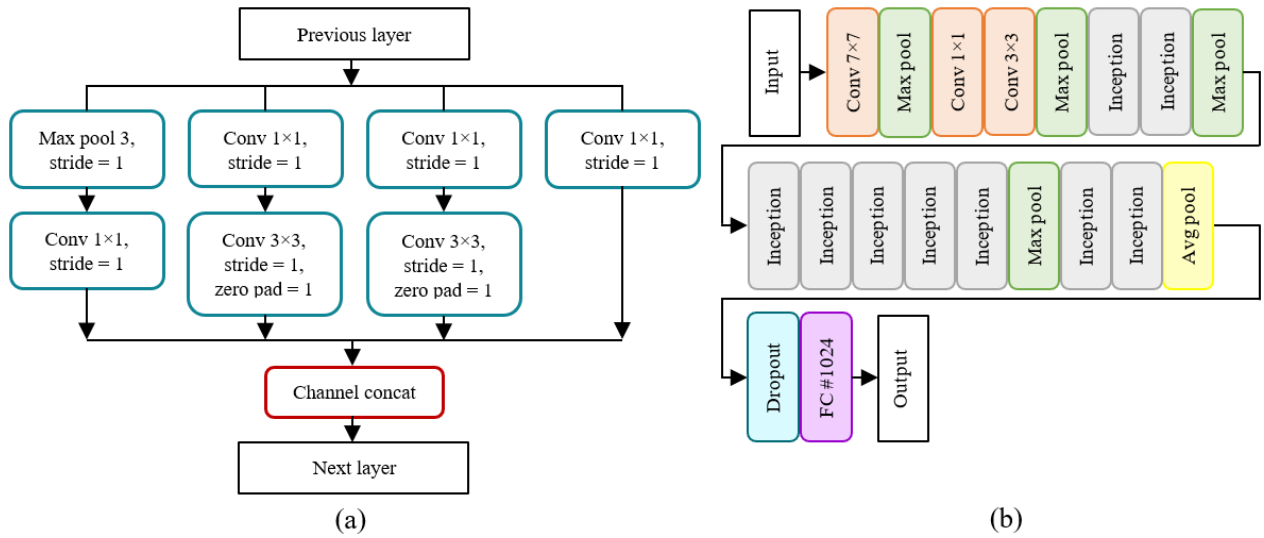


Figure 3. (a) Inception module of GoogleNet, (b) GoogleNet architecture.

2.3 RNN

Let $x = x_1, x_2, \dots, x_T$ denote as sequential data for the inputs of RNN, where x_t is input at time step t . As shown in Figure 4 (a), the input x_1 will be processed to produce hidden h_1 and output y_1 . The hidden h_1 will be used in next computations to produce hidden h_2 , and then hidden h_2 will be used to produce output y_2 . This process was done until we get output y_T .

There are various RNN models, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). LSTM has a memory cell and three gates in the hidden layer, namely forget, input and output gates; meanwhile, GRU has no separate memory cell and only has two gates, namely reset and update [30]. Although GRU has a simpler structure, which leads to faster computation than LSTM, GRU does not have enough memory for a longer time [31]. In this study,

we used LSTM as an RNN model.

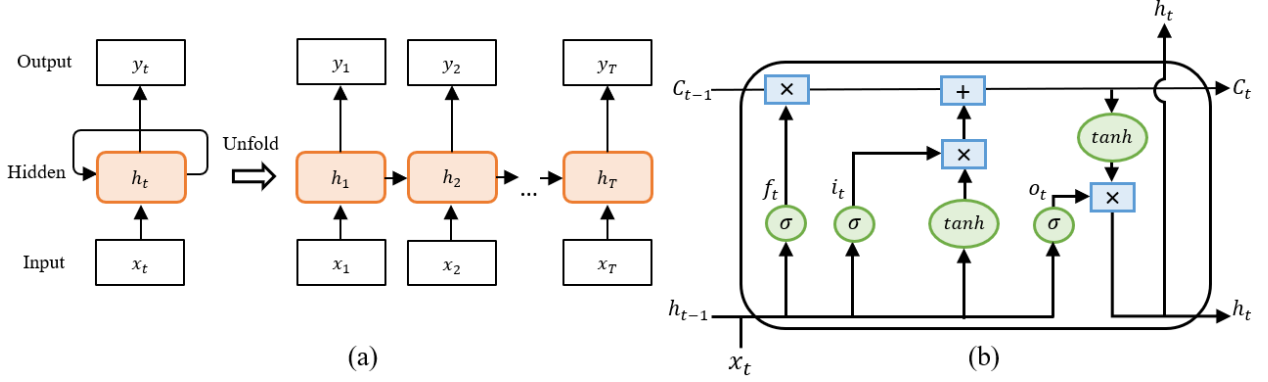


Figure 4. (a) Architecture RNN, (b) hidden layer of LSTM.

As shown in Figure 4 (b), the hidden layer of LSTM has memory cell C_t that will store information at each time step t , and the information stored in the memory cell C_t is controlled by three gates [32]. The forget gate f_t to determine what information in the previously hidden layer will be stored in the memory cell C_t . The input gate i_t determine what information in new input x_t which will be stored in the memory cell C_t . The output gate o_t to determine what information will be used as the output of the hidden layer h_t . The formula in hidden layers of LSTM as follows [19]:

$$f_t = \sigma(W_x^f x_t + W_h^f h_{t-1}) \quad (4)$$

$$i_t = \sigma(W_x^i x_t + W_h^i h_{t-1}) \quad (5)$$

$$C_t = C_{t-1} \times f_t + \tanh(W_x^c x_t + W_h^c h_{t-1}) \times i_t \quad (6)$$

$$o_t = \sigma(W_x^o x_t + W_h^o h_{t-1}) \quad (7)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (8)$$

where W^f , W^i , W^c and W^o are weights of forget gate, input gate, memory cell, and output gate, respectively.

2.4 Model Simulation

In this study, there are two processes, training dan testing. The training and testing process frameworks are shown in Figure 5 (a) and Figure 5 (b), respectively. The training process will produce the model that can map the retinal fundus image into a caption. Then, those models will

be evaluated in the testing process.

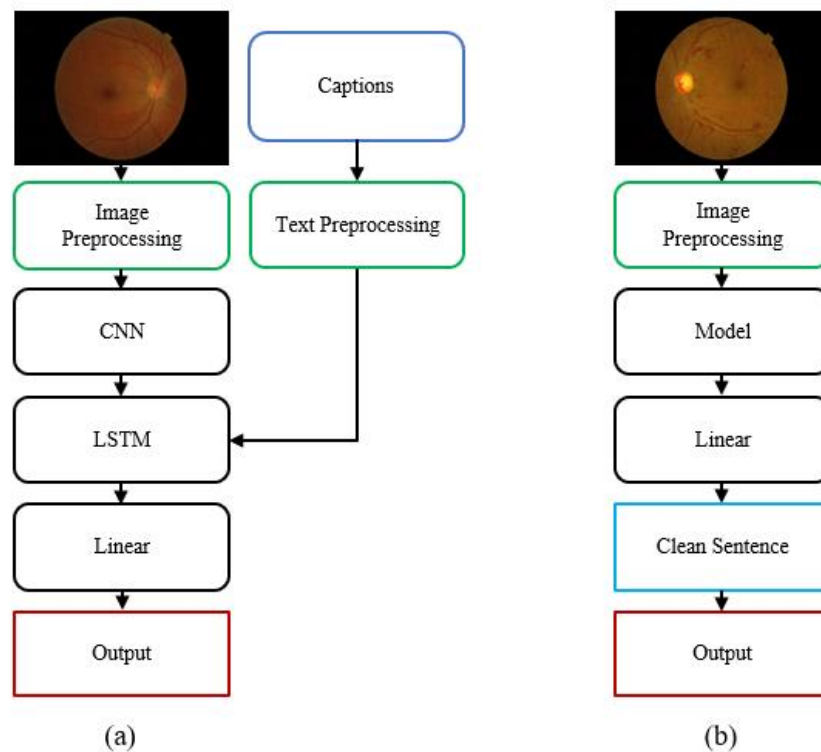


Figure 5. Framework of the (a) training process and (b) testing process. Retinal fundus image source: <http://www.adcis.net/en/third-party/messidor/> [20]

2.4.1 Image preprocessing

In the image preprocessing, we do two different phases that produce two different models. Those models are the model with and without contrast enhancement. The first step in image preprocessing is cropping each retinal fundus image. The original size of the retinal fundus image in MESSIDOR is 2240×1488 for width and height, respectively. After we cropped the fundus image, the size becomes 1450×1420 . This step is done for all models, with and without contrast enhancement.

The second step is converting the color of the retinal fundus images into greyscale and applying CLAHE. CLAHE is a contrast enhancement method by applying a threshold on the transformation function that transforms the histogram of the pixel level in the image [33]. Although CLAHE was used on the greyscale image for the first time, CLAHE can also work for a color image. The third step is to resize the retinal fundus image according to the CNN model used. AlexNet is 227×227 ,

VGGNet and GoogleNet are 224×224 . For the model without contrast enhancement, we do not through the second step and go directly to the third step.

2.4.2 Text preprocessing

Since the LSTM works on vectors and not integers, then the captions for the training process will be preprocessed before we input it to LSTM. The captions that consist of characters, such as words and punctuations, will be processed to the tokenizer. The tokenizer is used to convert a sequence of characters into a sequence of integers known as integer tokens. For example, in the caption ‘A healthy fundus image.’, then the caption’s integer tokens are [7, 22, 9, 13, 24]. The character ‘A’ is represented by integer 7, the character ‘healthy’ is represented by integer 22, and so on until the character ‘.’ is represented by integer 24. Determining the character’s integer based on the order in which the characters appear in the caption list.

The integer tokens will be processed to word embedding. Word embedding is used to convert an integer of one character into the vector of floating-point numbers. The vector of floating-point numbers is called an embedding vector. An example of word embedding using integer tokens [7, 22, 9, 13, 24] is shown in Table 2.

Table 2. Example of word embedding with an embedding vector size of 4.

Character	Integer of Character	Embedding Vector
A	7	[-0.9619, -2.4508, 1.1078, -0.3850]
healthy	22	[0.0290, 0.7733, -1.1539, -0.8179]
fundus	9	[1.0061, -1.0170, -0.1592, 0.4713]
Image	13	[0.5004, 0.2346, 1.3357, -0.2154]
.	24	[-0.5194, 1.4388, -0.2997, 0.4496]

Determining the value of the embedding vector of each character is done by treating the embedding vector as weight that will be trained in the training process. The embedding vector’s size is based on the features vector; if AlexNet and VGGNet, then the embedding size is 4,096. If GoogleNet, the embedding size is 1,024.

2.4.3 Training process

For the training process shown in Figure 5 (a), we need retinal fundus images and captions, which describe the lesion features in those retinal fundus images. The retinal fundus images that

have been preprocessed will be inputted to the CNN. Since this study uses CNN to extract images to obtain the image's features, the output layer is not used. The output of CNN is a feature vector, which summarizes the contents of the retinal fundus image. The feature vector's size is based on the last fully connected layer in CNN; 4,096 for AlexNet and VGGNet, 1,024 for GoogleNet. This feature vector is given as an initial memory cell to LSTM.

Furthermore, the embedding vector from text preprocessing's output will be inputted to LSTM. The hidden size of LSTM follows the embedding vector's size. If AlexNet and VGGNet are used, the hidden size of LSTM is 4,096. If GoogleNet is used, the hidden size of LSTM is 1,024. The LSTM's output is further given to the Linear layer to be converted into integer tokens. In the training process, the output is integer tokens. The integer tokens from the training process will be compared with the integer tokens of ground-truth to compute a model's loss. The training process will be done until we get the smallest loss.

2.4.4 Testing process

In the testing process, the models obtained from the training process are AlexNet and LSTM without CLAHE; AlexNet and LSTM with CLAHE; VGGNet and LSTM without CLAHE; VGGNet and LSTM with CLAHE; GoogleNet and LSTM without CLAHE; GoogleNet and LSTM with CLAHE. Those models will be evaluated by inputting each retinal fundus image into the model, then inputting the output's model to the Linear layer. The output of the Linear layer is integer tokens. Those integer tokens will be converted to the sequence of characters in the Clean Sentence phase, so that at the end, the output of the testing process in the form of a sentence. The testing process is shown in Figure 5 (b).

The indicators for evaluating the captions generation mostly use Bleu and Meteor, which evaluate whether the model can make captions that correspond to grammar based on the corpus [34]. Since the model's purpose in this study is to determine whether the generated captions contain lesion features relate to the retinal fundus image, those evaluating indicators are not applicable. We used the keyword of the name of the lesion features to evaluate the models. We will know which caption matches the condition lesion features in the retinal fundus image from the keyword.

The accuracy of the model is done by computing the percentage of the caption matching the condition of the lesion features in the retinal fundus image.

3. RESULTS AND DISCUSSIONS

We used Google Collaboratory with PyTorch library for model simulation. The data set was divided into data for training and data for testing processes with three proportions, i.e. 80% training data and 20% testing data; 70% training data and 30% testing data; 60% training data and 40% testing data. Each model’s training and testing process was simulated five times with different data split for each simulation. The results of the testing process can be seen in Table 3.

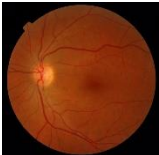

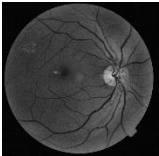
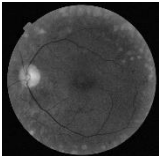
Table 3. The average accuracy of the testing process.

Proportion of Data	Model	Experiment					Average Accuracy
		1	2	3	4	5	
Training data: 80%, testing data: 20%.	AlexNet and LSTM without CLAHE	53.85	69.23	70.77	61.54	72.31	65.54
	AlexNet and LSTM with CLAHE	70.77	73.85	81.54	92.31	67.69	77.23
	VGGNet and LSTM without CLAHE	73.85	86.15	78.46	81.54	81.54	80.31
	VGGNet and LSTM with CLAHE	92.31	86.15	90.77	76.92	87.69	86.77
	GoogleNet and LSTM without CLAHE	93.85	93.85	87.69	95.38	92.31	92.62
	GoogleNet and LSTM with CLAHE	93.85	95.38	98.46	92.31	93.85	94.77
Training data: 70%, testing data: 30%.	AlexNet and LSTM without CLAHE	60.20	72.45	58.16	58.16	63.27	62.45
	AlexNet and LSTM with CLAHE	80.61	76.53	73.47	74.49	69.39	74.90
	VGGNet and LSTM without CLAHE	79.59	82.65	88.78	78.57	79.59	81.84
	VGGNet and LSTM with CLAHE	90.82	91.84	87.76	87.76	80.61	87.76
	GoogleNet and LSTM without CLAHE	93.88	94.90	95.92	97.96	95.92	95.71
	GoogleNet and LSTM with CLAHE	94.90	96.94	96.94	94.90	96.94	96.12
Training data: 60%, testing data: 40%.	AlexNet and LSTM without CLAHE	57.69	68.46	54.62	54.62	78.46	62.77
	AlexNet and LSTM with CLAHE	68.46	78.46	56.15	60.77	74.62	67.69
	VGGNet and LSTM without CLAHE	86.92	88.46	70.00	86.92	86.92	83.85
	VGGNet and LSTM with CLAHE	86.92	92.31	89.23	84.62	86.92	88.00
	GoogleNet and LSTM without CLAHE	93.85	92.31	92.31	95.38	93.85	93.54
	GoogleNet and LSTM with CLAHE	95.38	94.62	93.08	94.62	92.31	94.00

As shown in Table 3, GoogleNet and LSTM always get an accuracy of more than 87.69% for all experiments. We get the highest average accuracy of 96.12% by using GoogleNet and LSTM with CLAHE and a proportion of data 70% training data and 30% testing data.

The models using GoogleNet as the CNN model always get higher average accuracy than other models. This result indicates that our retinal fundus images are well extracted using GoogleNet. Furthermore, the models with CLAHE in most experiments, eighty-one of ninety experiments, have higher average accuracy than without CLAHE; these show that contrast enhancement with CLAHE affects the model accuracy. Besides the CNN model and model with CLAHE or not, the proportion of data used also affects our model accuracy. For our results, the highest average accuracy got when we use proportion of data 70% training data and 30% testing data.

Table 4. Example of output of this study.

No.	Input	Output	Ground-truth
1		a fundus image contains microaneurysms , hemorrhages , neovascularization .	Normal fundus image
2		normal fundus image .	Normal fundus image
3		this is a normal fundus image .	DR fundus image
4		there are microaneurysms , hemorrhages , and neovascularization in this fundus image .	DR fundus image

Compared to the previous work as mentioned in Section Introduction, our method able to produce the output in the form of captions. The example of the output can be seen in Table 4. Numbers 1 and 2 are examples of input for the model without CLAHE; meanwhile, numbers 3 and

4 are examples of input for the model with CLAHE. The output of numbers 1 and 3 is an example of incorrect output because it does not match the ground-truth. The correct output example is the output of numbers 2 and 4 because it matches the ground-truth.

4. CONCLUSION

Detecting DR lesion features and generate brief captions based on those lesion features using deep learning have been done. The experiment results show that from all models obtained, GoogleNet and LSTM with CLAHE using 70% training data and 30% testing data has the highest average accuracy. For future works, we suggest other researchers modify LSTM and use other pre-trained CNN models or construct a CNN model from scratch to improve the performance of accuracy.

ACKNOWLEDGMENT

This research was supported by PDUPT 2021 research grand with contract number NKB-155/UN2.RST/HKP.05.00/2021 from Kementrian Riset dan Teknologi/Badan Riset dan Inovasi Nasional.

CONFLICT OF INTERESTS

The author(s) declare that there is no conflict of interests.

REFERENCES

- [1] R. Yuneta, T.D. Gondhowiardjo, R. Dharma, et al. Levels of Hypoxia Inducible Factor-1 α (HIF-1 α) and Intercellular Adhesion Molecule-1 (ICAM-1) after Intravitreal Bevacizumab in Proliferative Diabetic Retinopathy. *Int. J. Retina*. 2 (2019), 21-27.
- [2] L. Wu, P. Fernandez-Loaiza, J. Sauma, et al. Classification of diabetic retinopathy and diabetic macular edema. *World J. Diabetes*, 4 (2013), 290-294.
- [3] S. Vujosevic, S.J. Aldington, P. Silva, et al. Screening for diabetic retinopathy: new perspectives and challenges. *Lancet Diabetes Endocrinol*, 8 (2020), 337-347.
- [4] A. Lands, A.J. Kottarathil, A. Biju, E.M. Jacob, S. Thomas, Implementation of deep learning based algorithms for diabetic retinopathy classification from fundus images, in: 2020 4th International Conference on Trends in

- Electronics and Informatics (ICOEI)(48184), IEEE, Tirunelveli, India, 2020: pp. 1028–1032.
- [5] Q.H. Nguyen, R. Muthuraman, L. Singh, G. Sen, A.C. Tran, B.P. Nguyen, M. Chua, Diabetic Retinopathy Detection using Deep Learning, in: Proceedings of the 4th International Conference on Machine Learning and Soft Computing, ACM, Haiphong City Viet Nam, 2020: pp. 103–107.
- [6] S. Pansawira, A. Bustamam, D. Sarwinda, Classification of Diabetic Retinopathy using shallow learning approach, in: Depok, Indonesia, 2020: p. 030009.
- [7] I. Qureshi, J. Ma, Q. Abbas, Recent development on detection methods for the diagnosis of diabetic retinopathy, *Symmetry*. 11 (2019), 749.
- [8] W.L. Alyoubi, W.M. Shalash, M.F. Abulhair, Diabetic retinopathy detection through deep learning techniques: A review, *Inform. Med. Unlocked*. 20 (2020), 100377.
- [9] N. Asiri, M. Hussain, F. Al Adel, N. Alzaidi, Deep learning based computer-aided diagnosis systems for diabetic retinopathy: A survey, *Artif. Intell. Med*. 99 (2019), 101701.
- [10] T.R. Athira, A. Sivadas, A. George, A. Paul, N.R. Gopan, Automatic detection of diabetic retinopathy using R-CNN. *Int. Res. J. Eng. Technol*. 6 (2019), 5595-5600.
- [11] J. Wang, J. Luo, B. Liu, R. Feng, L. Lu, H. Zou, Automated diabetic retinopathy grading and lesion detection based on the modified R - FCN object - detection algorithm, *IET Computer Vision*. 14 (2020), 1-8.
- [12] M.T. Esfahani, M. Ghaderi, R. Kafiyeh, Classification of diabetic and normal fundus images using new deep learning method. *Leonardo Electron. J. Practices Technol*. 17 (2018), 233-248.
- [13] Md.A. Habib Raj, Md.A. Mamun, Md.F. Faruk, CNN Based Diabetic Retinopathy Status Prediction Using Fundus Images, in: 2020 IEEE Region 10 Symposium (TENSYP), IEEE, Dhaka, Bangladesh, 2020: pp. 190–193.
- [14] T.A. Soomro, A.J. Afifi, A. Ali Shah, S. Soomro, G.A. Baloch, L. Zheng, M. Yin, J. Gao, Impact of image enhancement technique on CNN model for retinal blood vessels segmentation, *IEEE Access*. 7 (2019) 158183–158197.
- [15] H.A. Purwanithami, C. Atika Sari, E.H. Rachmawanto, D. Rosal Ignatius Moses Setiadi, Hemorrhage Diabetic Retinopathy Detection based on Fundus Image using Neural Network and FCM Segmentation, in: 2020 International Seminar on Application for Technology of Information and Communication (ISemantic), IEEE, Semarang, Indonesia, 2020: pp. 45–49.
- [16] S. Kaur, K.S. Mann, Optimized Technique for Detection of Diabetic Retinopathy using Segmented Retinal Blood Vessels, in: 2019 International Conference on Automation, Computational and Technology Management (ICACTM), IEEE, London, United Kingdom, 2019: pp. 79–83.
- [17] X. Liu, Q. Xu, N. Wang, A survey on deep neural network-based image captioning, *Vis Comput*. 35 (2019), 445–470.
- [18] X. Wei, H. Huang, L. Ma, Z. Yang, L. Xu, Recurrent Graph Neural Networks for Text Classification, in: 2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS), IEEE, Beijing, China, 2020: pp. 91–97.

DIABETIC RETINOPATHY DETECTION AND CAPTIONING

- [19] D. Verma, S.N. Muralikrishna, Semantic similarity between short paragraphs using Deep Learning, in: 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), IEEE, Bangalore, India, 2020: pp. 1–5.
- [20] <http://www.adcis.net/en/third-party/messidor/>
- [21] Md.S. Chowdhury, F.R. Taimy, N. Sikder, A.-A. Nahid, Diabetic Retinopathy Classification with a Light Convolutional Neural Network, in: 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), IEEE, Rajshahi, Bangladesh, 2019: pp. 1–4.
- [22] M. Bejiga, A. Zeggada, A. Nouffidj, F. Melgani, A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery, *Remote Sensing*. 9 (2017), 100.
- [23] D.A. Jasm, M.M. Hamad, A.T. Hussein Alrawi, Deep image mining for convolution neural network, *Indonesian J. Electric. Eng. Computer Sci.* 20 (2020), 347-352.
- [24] M.M. Fouad, E.M. Mustafa, M.A. Elshafey, Detection and localization enhancement for satellite images with small forgeries using modified GAN-based CNN structure. *Int. J. Adv. Intell. Inform.* 6 (2020), 278-289.
- [25] S. Sengupta, A. Singh, H.A. Leopold, T. Gulati, V. Lakshminarayanan, Ophthalmic diagnosis using deep learning with fundus images – A critical review, *Artif. Intell. Med.* 102 (2020), 101758.
- [26] T. Shanthi, R. S. Sabeenian, Modified Alexnet architecture for classification of diabetic retinopathy images. *Computers Electric. Eng.* 76 (2019), 56-64.
- [27] G. Kovacs, L. Totha, D.V. Compernelle, S. Ganapathy, Increasing the robustness of CNN acoustic models using autoregressive moving average spectrogram features and channel dropout. *Pattern Recogn. Lett.* 100 (2017), 44-50.
- [28] X. Wang, Y. Lu, Y. Wang, W.-B. Chen, Diabetic Retinopathy Stage Classification Using Convolutional Neural Networks, in: 2018 IEEE International Conference on Information Reuse and Integration (IRI), IEEE, Salt Lake City, UT, 2018: pp. 465–471.
- [29] I. Kandel, M. Castelli, Transfer Learning with Convolutional Neural Networks for Diabetic Retinopathy Image Classification. A Review, *Appl. Sci.* 10 (2020), 2021.
- [30] B. Wang, F. Miao, X. Wang, L. Jin, Text Classification Using a Bidirectional Recurrent Neural Network with an Attention Mechanism, in: 2020 International Conference on Culture-Oriented Science & Technology (ICCST), IEEE, Beijing, China, 2020: pp. 265–268.
- [31] M.U. Salur, I. Aydin, A novel hybrid deep learning model for sentiment classification. *IEEE Access*, 8 (2020), 58080–58093.
- [32] C. Amritkar, V. Jabade, Image Caption Generation Using Deep Learning Technique, in: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), IEEE, Pune, India, 2018: pp. 1–4.
- [33] I. Qureshi, J. Ma, K. Shaheed, A Hybrid Proposed Fundus Image Enhancement Framework for Diabetic Retinopathy, *Algorithms*. 12 (2019), 14.
- [34] Hartatik, H. Al Fatta, U. Fajar, Captioning Image Using Convolutional Neural Network (CNN) and Long-Short

Term Memory (LSTM), in: 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), IEEE, Yogyakarta, Indonesia, 2019: pp. 263–268.