Journal of
Imaging

MDPI

Article

Hybrid of Deep Learning and Word Embedding in Generating Captions: Image-Captioning Solution for Geological Rock Images

Agus Nursikuwagus [1,*], Rinaldi Munir [2,*] and Masayu Leylia Khodra [2]

[1] Doctoral Program of Informatics, School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganesha No.10, Bandung 40132, Indonesia
[2] Department of Informatics, School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganesha No.10, Bandung 40132, Indonesia
* Correspondence: agusnursikuwagus@email.unikom.ac.id (A.N.); rinaldi@informatika.org (R.M.)

Abstract: Captioning is the process of assembling a description for an image. Previous research on captioning has usually focused on foreground objects. In captioning concepts, there are two main objects for discussion: background object and foreground object. In contrast to the previous image-captioning research, generating captions from the geological images of rocks is more focused on the background of the images. This study proposed image captioning using a convolutional neural network, long short-term memory, and word2vec to generate words from the image. The proposed model was constructed by a convolutional neural network (CNN), long short-term memory (LSTM), and word2vec and gave a dense output of 256 units. To make it properly grammatical, a sequence of predicted words was reconstructed into a sentence by the beam search algorithm with K = 3. An evaluation of the pre-trained baseline model VGG16 and our proposed CNN-A, CNN-B, CNN-C, and CNN-D models used BLEU score methods for the N-gram. The BLEU scores achieved for BLEU-1 using these models were 0.5515, 0.6463, 0.7012, 0.7620, and 0.5620, respectively. BLEU-2 showed scores of 0.6048, 0.6507, 0.7083, 0.8756, and 0.6578, respectively. BLEU-3 performed with scores of 0.6414, 0.6892, 0.7312, 0.8861, and 0.7307, respectively. Finally, BLEU-4 had scores of 0.6526, 0.6504, 0.7345, 0.8250, and 0.7537, respectively. Our CNN-C model outperformed the other models, especially the baseline model. Furthermore, there are several future challenges in studying captions, such as geological sentence structure, geological sentence phrase, and constructing words by a geological tagger.

Keywords: deep learning; vector embedding; convolutional neural network; recurrent neural network

check for updates

Citation: Nursikuwagus, A.; Munir, R.; Khodra, M.L. Hybrid of Deep Learning and Word Embedding in Generating Captions: Image-Captioning Solution for Geological Rock Images. *J. Imaging* **2022**, *8*, 294. https://doi.org/10.3390/jimaging8110294

Academic Editors: Antonio Fernández-Caballero and Byung-Gyu Kim

Received: 25 July 2022
Accepted: 22 September 2022
Published: 22 October 2022

1. Introduction

Geological observation involves field research by a geologist. One of the tasks is to write about the rock's content and take a photo of the rock. Each picture is paired with its description. This task requires an expert to write carefully and accurately about each image. Each description should contain the rock's characteristics, including the rock's color, shape, and constituents. This process is an essential task because the information helps in decisions on other activities, such as mining, land fertilization, field surveillance, and drilling. Sometimes, a geologist finds the same characteristics in the rocks that correspond to previous descriptions and repeatedly writes the same descriptions. It is interesting to adopt these tasks for researching how to make descriptions for the content of a photo image. The geologist's experience can be assumed as staging for the creation of a description for each rock image. In computer vision terms, those activities can be defined as captioning. One challenging caption process is how to make the images and their descriptions pairwise. With captioning in place, we can predict and describe the content of other photos. Adopting a geologist's knowledge as an intelligent system for captioning is one aspect that can be explored. How to identify rocks and create descriptions of the content of rock images has been proposed as a research topic.

Most works have proposed some image-captioning concepts. Krizhevsky proposed a CNN for ImageNet and contributed to image extractions [1]. Other research proposed by Karpathy introduced some concepts to make a caption from an image [2]. Supported by ImageNet classification, Karpathy successfully contributed to creating a captioning model for MS COCO and Flickr images [2]. The foreground object is one focus for identifying the objects in an image; the words are then joined to make a sentence that the reader can understand. Ignoring the background object should be avoided when generating a caption. The same works have contributed different ideas, such as captioning based on an identified word-by-word phrase between an object and other objects [3]. Captioning based on text analysis has also been proposed [4–9]. Other concepts, such as identifying objects by the object element [10–16] and studying the generation of captions as part of speech guidance, have been proposed by [8,9,17–23].

The development of the concept of image captioning cannot be separated from the development of deep learning models. The convolutional neural network (CNN) and recurrent neural network (RNN) can predict the word that pairs with the image region and the predicted word is close to a human description. The CNN, developed by Lecun, has supported production feature images by approximating the classification of the image [24,25]. On the other hand, the RNN also helps generate words that can be paired with the image region, followed by the conditional probability formulation of $L(x) = p(x\,|x_1 \ldots x_{i-1}; I)$ [26–36]. Model image captioning has been researched by many using CNN block and language models, such as DenseNet and LSTM [9,17], CNN and LSTM [19,26,33,37–39], inceptionV3 and RNN [14], and CNN and BERT [40,41]. One of the important parts of captioning is word embedding, which provides a vector feature value for each word. Word2vec [2,6,18,42–44] and one-hot-vector [20,23,34,44–47] are embedding models that many scholars have used for image captioning.

Many research studies have introduced a captioning model that strengthens a foreground object [5,11,12,14–16,18,20,23,36–38,44,46,48–56]. Images contain two parts: foreground and background objects. Foreground objects are identified by making a class of objects, such as cars, men, women, roads, stairs, grass, birds, and animals, whereas a *background object* is an area lying behind the foreground objects at the back of the image, including objects, such as walls, yards, and the sky. In a geological image, the rock is a dominant background object presented at the back of the image [57–60].

The problem with this study was how to identify the object that appears behind the more common objects. Previous research has not focused on the background of common objects. Thus, it is a challenge to research and explore this domain, especially with geological rock images. Many researchers have proposed effective captioning models. Nevertheless, the resulting BLEU (B@N) scores have not achieved a desirable score. Moreover, the geological rock captions have not proven similar to the geologist's descriptions. A preliminary experiment performed using CNN + LSTM + one hot vector only presented scores of 0.3990, 0.3830, 0.4050, and 0.3470 for B@1, B@2, B@3, and B@4, respectively [2]. The model identified common objects, such as humans, cars, and animals. All background objects, including the rock, were unnecessary to infer a caption. These results prove that there are still challenges in captioning.

The interpretation of objects from the geologist's perception encompasses all objects that appear in the background of the image visualization. The geologist focuses on interpreting the rock in the image and ignores the common objects. This is contrary to the previous studies about captioning and a different way of presenting captions. Captioning experiments from this perspective is directed to describing and identifying the rock in the images. Other aspects in the images, such as people, cars, and animals, are not important for identifying the rock. Furthermore, our model can predict the words for the rock in the image and acquire the image's caption. Figure 1 highlights the problem domain of the study. The objective was to generate a caption for rock images. In line with the research objective, the caption outcomes are in Indonesian and match with the image of the rock.

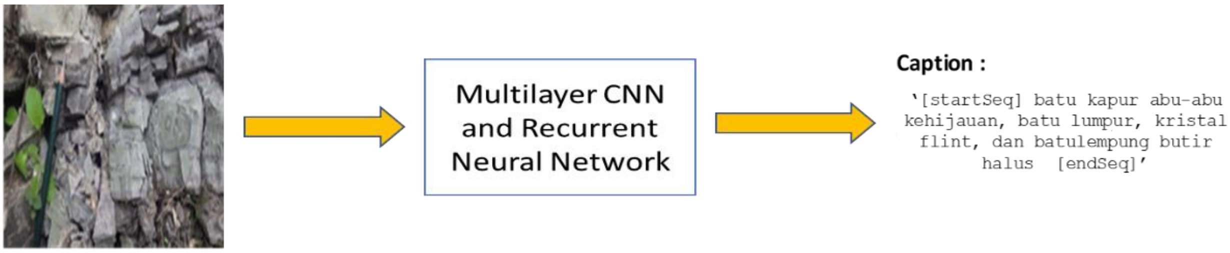

Multilayer CNN and Recurrent Neural Network
Caption :
'[startSeq] batu kapur abu-abu kehijauan, batu lumpur, kristal flint, dan batulempung butir halus [endSeq]'

Figure 1. Research Problem on captioning geological terms. The translation of the caption is greenish-gray limestone, mudstone, flint crystal, and fine-grain claystone (CNN: convolutional neural network).

With regard to the problem and the proposed model, we propose contributions to solve some problems relating to captioning. Our contribution can be presented as follows:

- Using geological field exploration to support captioning and build a model that produces a caption from an image. We collected that geological knowledge and used it to construct an algorithm and the architecture of the captioning model.
- Building the corpus for captioning that contains the pairwise images of rocks and their captions.
- Building a captioning model that can interpret images of rocks and achieve outcomes with an accuracy that is similar to a geologist's annotation. Our models can outperform the baseline model relating to the BLEU score and acquire captions that are similar to a geologist's annotation.

We arranged the sections in this paper as follows. The paper leads with an introduction, which encompasses the urgency for this type of research and the research problems. The methods section details content theory and related research, with associated research, such as on convolutional neural networks, long short-term memory, BLEU score measurement, and the estimation function. The subsequent section presents the proposed model to solve the current problems. This paper also conveys the outcome of experiments using the proposed model. Then, the discussion section explains the results. The last section presents the conclusions.

2. Methods

2.1. Long Short-Term Memory (LSTM)

The LSTM language model can be used as part of the process of generating captions automatically. Another method of sentence representation is the RNN, a simpler method than LSTM. Karpathy used the same number of dimensions as the length of words in the image descriptions from the experts [2]. The RNN method is a simple approach for producing the sentence representation of the detected object. This approach does not consider the order or context of information in a sentence. In the sequence, the results obtained are not grammatically arranged. Karpathy used bi-gram techniques or two dependency relations to generate a sentence with support from either the beam search algorithm or the greedy search algorithm to overcome this [2].

Long short-term memory (LSTM) is a language model, the successor of the RNN, that enables long-term learning. The LSTM unit has an additional hidden state as a nonlinear mechanism that allows a state to perform a backpropagation process without modification, change, or reset. Learning in LSTM uses simple function gates to learn speech recognition and language translation [19]. Figure 2 is a simple form of the recurrent neural network (RNN) and LSTM. LSTM is processed by repeating each process performed. In LSTM, the value $\sigma(x) = (1 + e^{-x})^{-1}$ as a sigmoid function with an accurate boundary value between 0 and 1, whereas for nonlinear hyperbolic functions $tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2\sigma(2x) - 1$ with input range values between -1 and 1.

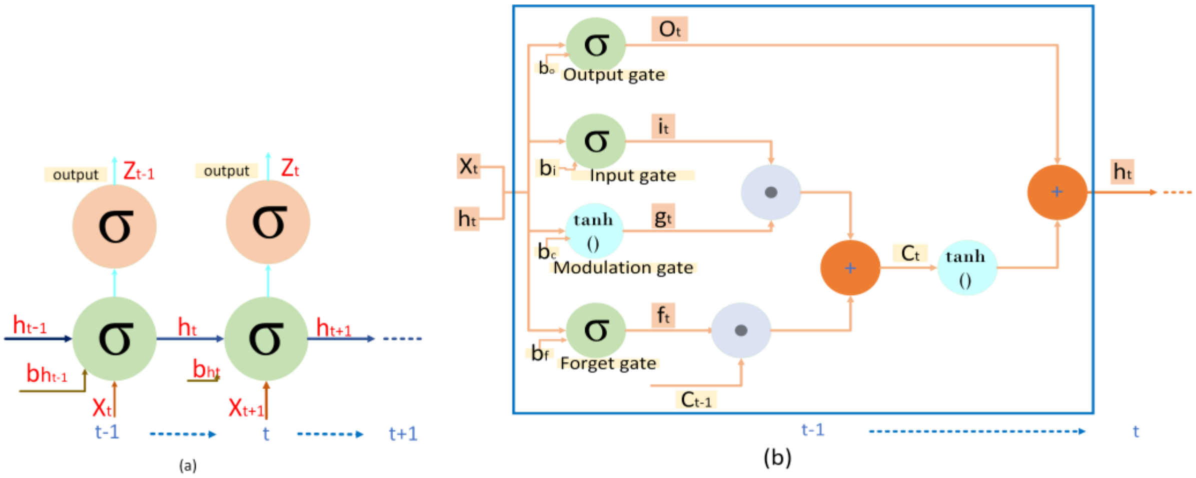


Figure 2. (**a**) RNN and (**b**) LSTM decoder architecture (RNN: recurrent neural network, LSTM: long short-term memory).

LSTM updates based on time t for input x_t, h_{t-1}, and c_{t-1} following the functions below:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \tag{2}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{3}$$

$$g_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{4}$$

$$c_t = f_t \, ^\circ \, c_{t-1} + i_t{}^\circ g_t \tag{5}$$

$$h_t = o_t{}^\circ tanh(c_t) \tag{6}$$

where i_t stands for input at timestamp t, f_t is the symbol for the forget gate at timestamp t, o_t stands for the output gate at timestamp t, and g_t is a hyperbolic function, c_t stands for the concatenation elementwise operation, and h_t is an output state.

The RNN updates the values on time t for input x_t and h_{t-1} following the formulas below:

$$h_t = g(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{7}$$

$$z_t = g(W_{hz}h_t + b_z) \tag{8}$$

where g is an elementwise nonlinearity, such as a sigmoid function or hyperbolic tangent, x_t is an input, $h_t \in \mathbb{R}^N$ is a hidden state with N hidden units, and z_t is the output at time t. For the length of the T input sequence $x_1, \; x_2, \ldots, x_T$, the update carried out is to calculate sequentially by ignoring $h_0, h_1, z_1, h_2, z_2, \ldots, h_T, z_T$.

2.2. Part One Architectures

Our pipeline model was constructed and strengthened by: (1) image preprocessing, including resizing, reducing, and cropping; (2) ensuring that the image could be recognized without losing the pixel information when using the reduce function in the CNN; (3) finding a suitable CNN for our domain, considering CNN layers, dropout, pooling, and dense units. Captioning architecture is essential to generate a word approximate to the human description. Identifying the rocks in the image differs with regard to the colors in the image.

Figure 3 depicts a geology caption model and divides the model into parts, such as image extraction, word embedding, generating words, and assembling the captions. The image extraction part is in part one and word embedding is in part two. The outcome units are 256 units for each learning stage. We concatenated both outcomes between image extraction and the LSTM unit. Furthermore, after compiling using an ADAM optimizer with learning = 0.0001, we acquired 12,746,112, 2,397,504, 20,482,432, and 104,867,300 training parameters for CNN-A, CNN-B, CNN-C, and CNN-D, respectively. These parameters were achieved from the reengineering of VGG16 and word embedding [2].

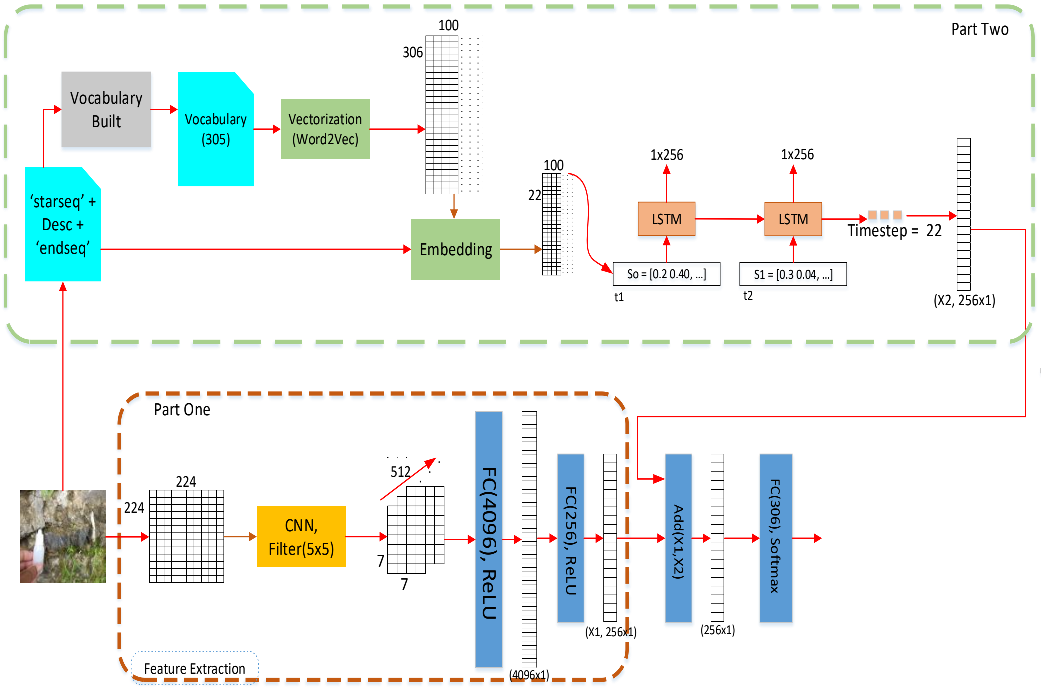


Figure 3. Proposed deep learning architecture (CNN: convolutional neural network, LSTM: long short-term memory).

Following the significant step, the captioning model always starts with a feature extraction model. An input image had a size of 224 × 224 with RGB color and three channels. We worked on the recognized image to convert it into an RGB value with three channels. After that, we proposed the image extracts using a CNN, shown in Figure 3 [52]. *Architecture classification is the classification model that identifi*es every single rock [53]. The output of the convolution process was dense with 4096 units. The convolution utilized the dropout function to avoid overfitting, with a rate of 0.5. Every value in a feature map was scaled up using the formula x/(1–0.5), where x is the single feature map.

After the dropout function, the process continued to leverage linear activation or ReLU and gave outcomes of 256 units [17]. The ReLu function calculated a maximized output between the vector feature value and 0 or Max[x,0]. The ReLu function will present 0 if the feature value is smaller than 0 or the original, x. The outcome of the ReLu function was 256 units. The outcome of part one was a sequence unit with a length of 256 neuron units.

The pipeline process of part one can be written as the following pseudocode:

```
1. Input Image ($i_1$, $i_2$, $i_3$, . . . , $i_n$)—$n$ stands for collecting the image, $i$;
2. Reduce image; function for reduction of image to a size of 224 × 224;
3. For $I \in (i_1, i_2, i_3, \ldots, i_n)$—$I$ is the collection of the image:
    a. Image_feature = CNN(I, C, F)—I is the image with a size of 224 × 224; C represents the channels of convolutions; F is the filter matrix size that can be 3 × 3 or 5 × 5;
    b. Feature_Pooling = MaxPooling(Image_feature)—operation of the MaxPooling function;
    c. Feature_ReLu = ReLu Activation (Feature Pooling)—run of the ReLu activation;
    d. Feature_Dense = Dense(Feature_ReLu)—provides outcome units with 256 dense units.
4. Return (Feature_Dense).
```

2.3. Part Two Architectures

In Figure 3, part two is a concept to create a vector feature value from a geologist's description. Word2vec is an important embedding model that provides a vector sequence value for each vocabulary unit. Word2vec can produce a vector with 306 × 100 dimensions for word embedding [54]. After the embedding process, vector word embedding was transformed into a vector feature for 22 words for captions that had 22 × 100 dimensions. LSTM used those vectors to generate a word that matched the image feature. Furthermore, the process continued to the dense layer to obtain the Max value via the ReLu function. The last process in part two was to carry out an output of 256 units. This study used the SoftMax, Equation (14), function to acquire a probability value and select the higher probability as a proper word. Equation (14) is a SoftMax function $\sigma : \mathbb{R}^K \rightarrow [0,1]^K$ which is defined as follows:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{9}$$

where $i = 1 \ldots K$ and $z = (z_i \ldots z_K) \in \mathbb{R}^K$; $\sigma(z)_i$ is a probability value for every unit at index-i; and e^{z_i} is an epsilon of the vector at each unit, z, at index-i.

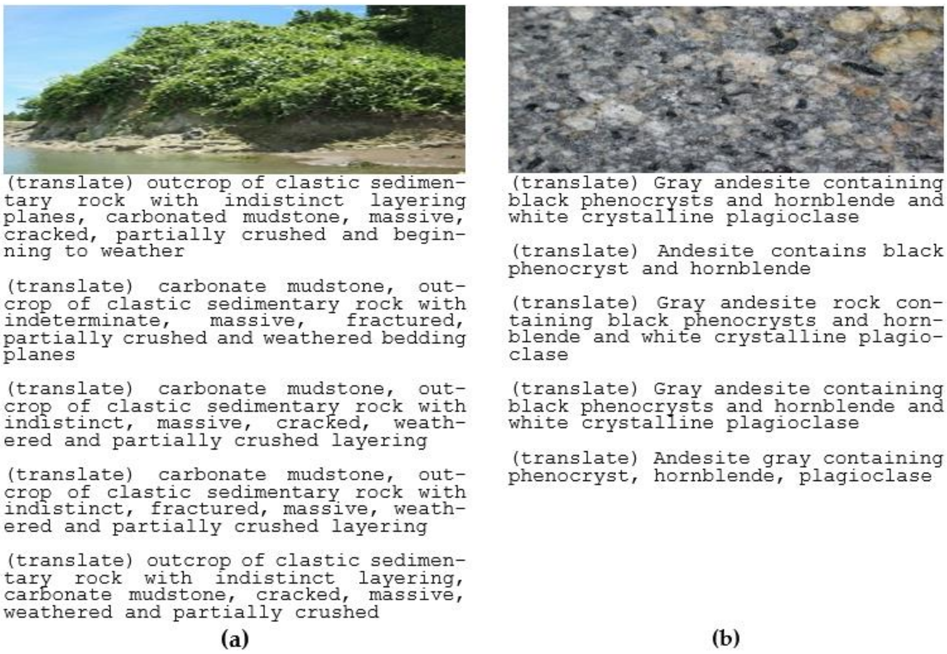
We proposed the process flow of part two from our architectures. This pseudocode writes the following schema of process:

1. C = Input (caption)—input the corpus from the geologist's annotation;
2. X = 'start_seq'—initialization of the word;
3. U = unique_word(C)—building a unique word into a vocabulary that attaches from the corpus, C;
4. C_Index = Making_Index(C)—providing the index for each word of the vocabulary;
5. C_vector = Word2Vec(C_Index)—achieving the feature vector from the pre-trained model. The outcomes were 306 × 100 dimensions;
6. W_embedding = Embedding (Vector word 'Start_seq' + C_vector);
7. Word_predict = LSTM (w_embedding);
8. Return (feature Length = 256 units).

After completing the processing of both parts, the process continues to add both units to be one pairwise vector feature. In Figure 4, there is an operation ADD (X1, X2). This means that both units were flattened from the image and text extraction to create one flattened vector feature. After that, the last process was to operate the SoftMax function to acquire the word classification. The word classification process aligns the predicted word with the image feature.

2.4. Word Embedding

To provide vector value embedding, we used Word2Vec as a word embedding model to provide a map of values of the words. Word2vec was introduced by Thomas Mikolov and consists of two processes: continuous bag of words (CBOW) and continuous n-skip gram. Each process has a unique task in handling a word. CBOW acts as a neural network process that gives a probability value and selects the higher probability as a candidate value. On the other hand, the continuous n-skip gram process takes the current word as the input and tries to accurately predict the words before and after this current word [54]. This study used 100 dimensions for each word and there was a 306 × 100 vector space for all words.



(translate) outcrop of clastic sedimentary rock with indistinct layering planes, carbonated mudstone, massive, cracked, partially crushed and beginning to weather

(translate) carbonate mudstone, outcrop of clastic sedimentary rock with indeterminate, massive, fractured, partially crushed and weathered bedding planes

(translate) carbonate mudstone, outcrop of clastic sedimentary rock with indistinct, massive, cracked, weathered and partially crushed layering

(translate) carbonate mudstone, outcrop of clastic sedimentary rock with indistinct, fractured, massive, weathered and partially crushed layering

(translate) outcrop of clastic sedimentary rock with indistinct layering, carbonate mudstone, cracked, massive, weathered and partially crushed

(a)

(translate) Gray andesite containing black phenocrysts and hornblende and white crystalline plagioclase

(translate) Andesite contains black phenocryst and hornblende

(translate) Gray andesite rock containing black phenocrysts and hornblende and white crystalline plagioclase

(translate) Gray andesite containing black phenocrysts and hornblende and white crystalline plagioclase

(translate) Andesite gray containing phenocryst, hornblende, plagioclase

(b)

Figure 4. Geological rock images and their captions translated into English: (**a**) captions for the geological image of sedimentary rocks and translated into English; (**b**) captions for geological image on andesite contents and translated into English.

3. Results

This section reports on the experiments' achievement and is separated into two subsections: dataset and experiments. We used Google Collaboratory Pro, Python version 3.6, TensorFlow, a GPU, and 25 GB of RAM for the experiments. The GPU used by default Google Collaboratory is NVIDIA T4 or P100.

3.1. Dataset

We started by collecting the images and carrying out the data preprocessing. Following the proposed pipeline model in Figure 3, the first step was to reduce the image into 224 × 224 pixels with RGB color. As the input in the CNN, all images were set to 32 channels in the first convolution process.

We collected 1297 images of geological rocks and divided them into two datasets. The training dataset was 1001 images and the validation dataset was 296 images. In addition, we added five captions for each image and acquired a caption from a geologist for each image as a geological corpus. The caption was completed by the geologist following the guidelines of writing lithology descriptions and arranged into the names of the rock, color, and dominant rocks [55].

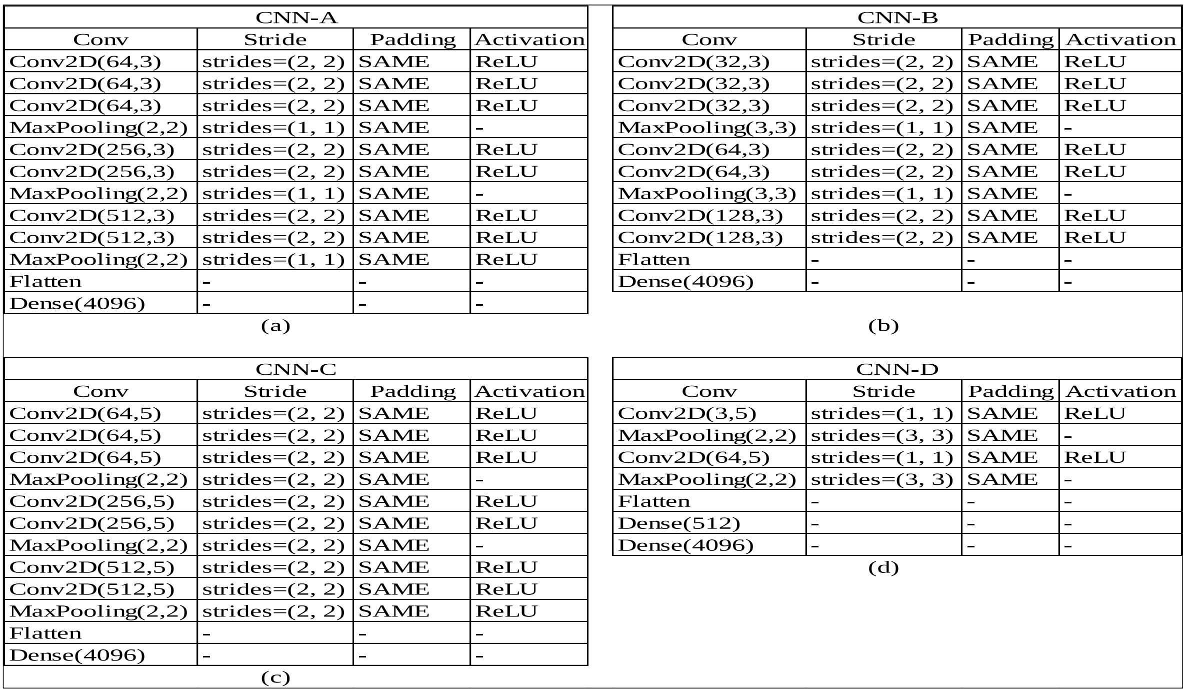
Figure 4 shows examples of images from the dataset and their captions. The captions were written in the Indonesian language and we translated the text to make the captions clear. This study was focused on developing a model in the Indonesian language, because of its usefulness for further study. The translation is included for the purpose of making the paper easy to read.

3.2. Experiments

The proposed models were assembled from reengineering the VGG16 and our model called CNN-n. The choice of the name CNN-n is based on the experiments, which consisted of several layers or shallow learning. Our model was divided into CNN-A, CNN-B, CNN-C, and CNN-D sub-models. The CNN-n model was introduced from the image classification results regarding rock types [56]. CNN-D using conv (32, 5) and conv (64, 5) produced a larger number of output parameters than the CNN-A, CNN-B, and CNN-C models.

Simonyan and Zisserman stated that the filter plays an important role in extracting an image [57]. Filters are square matrices with odd numbers, such as 1 × 1, 3 × 3, 5 × 5, and 7 × 7 [57]. In several studies on image extraction, scholars used many 3 × 3 filters. This particular filter proved effective in extracting the image and providing a feature map for recognizing the image [53,58].

Figure 5 shows several architectures from our re-engineered CNN models. We obtained outputs for VGG16, CNN-A, CNN-B, CNN-C, and CNN-D of 134,260,544, 12,746,112, 2,397,504, 20,482,432, and 104,867,300, respectively. These values were created from the flattening unit multiplied by the defined output unit. For instance, CNN-D had 104,867,300 parameters; for the CNN-D architectures in Figure 5d, the last fully connected (FC) layer was 50 flattened units, 512 dense units, and 4096 FC units. The parameter size is created by multiplication among the flattened units, dense units, and FC units. Thus, the parameter output was 104,867,300. The 50 flattened units are array units that provide output from the convolutional process after the MaxPooling layer and ReLu functions.



CNN-A			
Conv	Stride	Padding	Activation
Conv2D(64,3)	strides=(2, 2)	SAME	ReLU
Conv2D(64,3)	strides=(2, 2)	SAME	ReLU
Conv2D(64,3)	strides=(2, 2)	SAME	ReLU
MaxPooling(2,2)	strides=(1, 1)	SAME	-
Conv2D(256,3)	strides=(2, 2)	SAME	ReLU
Conv2D(256,3)	strides=(2, 2)	SAME	ReLU
MaxPooling(2,2)	strides=(1, 1)	SAME	-
Conv2D(512,3)	strides=(2, 2)	SAME	ReLU
Conv2D(512,3)	strides=(2, 2)	SAME	ReLU
MaxPooling(2,2)	strides=(1, 1)	SAME	ReLU
Flatten	-	-	-
Dense(4096)	-	-	-

(a)

CNN-B			
Conv	Stride	Padding	Activation
Conv2D(32,3)	strides=(2, 2)	SAME	ReLU
Conv2D(32,3)	strides=(2, 2)	SAME	ReLU
Conv2D(32,3)	strides=(2, 2)	SAME	ReLU
MaxPooling(3,3)	strides=(1, 1)	SAME	-
Conv2D(64,3)	strides=(2, 2)	SAME	ReLU
Conv2D(64,3)	strides=(2, 2)	SAME	ReLU
MaxPooling(3,3)	strides=(1, 1)	SAME	-
Conv2D(128,3)	strides=(2, 2)	SAME	ReLU
Conv2D(128,3)	strides=(2, 2)	SAME	ReLU
Flatten	-	-	-
Dense(4096)	-	-	-

(b)

CNN-C			
Conv	Stride	Padding	Activation
Conv2D(64,5)	strides=(2, 2)	SAME	ReLU
Conv2D(64,5)	strides=(2, 2)	SAME	ReLU
Conv2D(64,5)	strides=(2, 2)	SAME	ReLU
MaxPooling(2,2)	strides=(2, 2)	SAME	-
Conv2D(256,5)	strides=(2, 2)	SAME	ReLU
Conv2D(256,5)	strides=(2, 2)	SAME	ReLU
MaxPooling(2,2)	strides=(2, 2)	SAME	-
Conv2D(512,5)	strides=(2, 2)	SAME	ReLU
Conv2D(512,5)	strides=(2, 2)	SAME	ReLU
MaxPooling(2,2)	strides=(2, 2)	SAME	ReLU
Flatten	-	-	-
Dense(4096)	-	-	-

(c)

CNN-D			
Conv	Stride	Padding	Activation
Conv2D(3,5)	strides=(1, 1)	SAME	ReLU
MaxPooling(2,2)	strides=(3, 3)	SAME	-
Conv2D(64,5)	strides=(1, 1)	SAME	ReLU
MaxPooling(2,2)	strides=(3, 3)	SAME	-
Flatten	-	-	-
Dense(512)	-	-	-
Dense(4096)	-	-	-

(d)

Figure 5. Architectures of proposed CNN models. (**a**) CNN with seven layers convolutional and uses three layers of MaxPooling, (**b**) CNN with seven layers convolutional and two layers of MaxPooling, (**c**) same as (**a**) but different size of MaxPooling, (**d**) CNN with two layers convolutional, two MaxPooling layers, and two fully connected (FC) layers.

We delivered the duplicate operation pruning for each model at the last layer of unit classification. These actions were taken because we needed the weight of the units for the subsequent process. The process continued to concatenate operations between FC units from the CNN and LSTM units to gain a predicted word. This model follows the likelihood function $L(\theta) = \prod_{i}^{n} p(w_1, w_2, \ldots w_n,\ I|\theta)$, where w stands for previous words, I is an image region, and θ is a predicted word.

Figure 6 shows the accuracy curve for each CNN model and depicts the comparison between loss and accuracy. We observed that the accuracy increased at 80 epochs. Our CNN-A, CNN-B, CNN-C, CNN-D, and VGG16 architecture had accuracies of 0.9137, 0.9148, 0.9178, 0.9206, and 0.9228, respectively. Figure 6 presents how much the experiments were influenced by the number of CNN layers and parameters in the domain under study. The curves depend on the receptive field and channel settings.

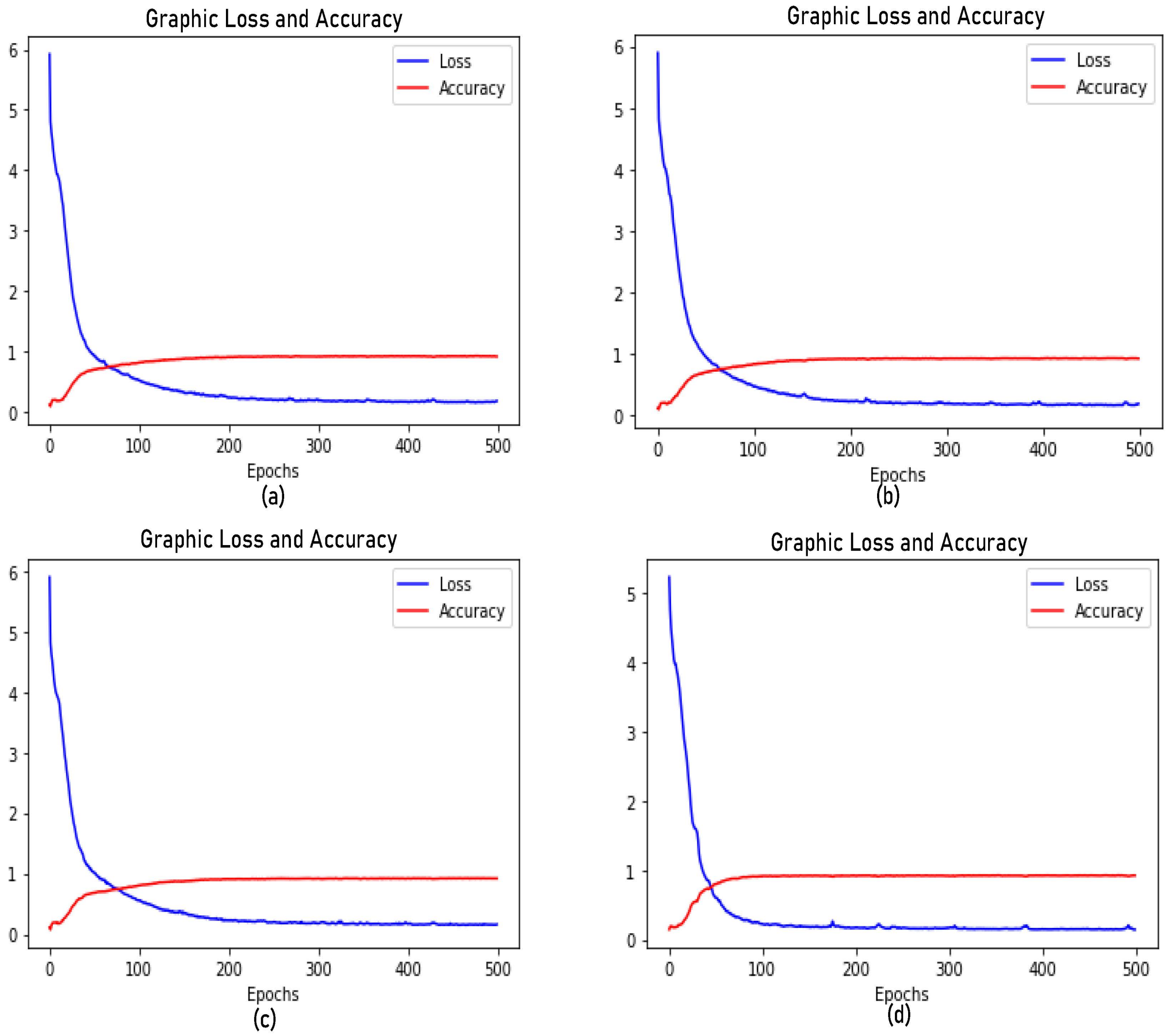


Figure 6. Accuracy and loss graphs of the proposed models: (**a**) CNN-A, (**b**) CNN-B, (**c**) CNN-C, and (**d**) CNN-D. The experiments used 500 epochs as a training measure. The graphs show that each model needed just below 100 epochs to reach its best accuracy.

Table 1 shows a comparison of the training parameters for each of the CNN architectures. The VGG16 model had 134,260,544 training parameters for image extraction and 4096 dense units. However, CNN-B and CNN-D had the same processing time. In order to achieve a sentence that could be adequately read, we generated sentences using a beam search algorithm [59].

Table 1. Training parameter comparison. ADAM optimizer with lr = 0.0001, MaxPooling, ReLu activation, and 4096 flattened units employed for tuning parameters.

Model	Layer	Filter	Parameters	Size	Time(s)
VGG16	16	1 × 1, 3 × 3	134,260,544	224 × 224	1845
Ours_CNN-A	7	3 × 3	12,746,112	224 × 224	1860
Ours_CNN-B	7	3 × 3	2,397,504	224 × 224	1230
Ours_CNN-C	7	5 × 5	20,482,432	224 × 224	1885
Ours_CNN-D	2	5 × 5	104,867,300	224 × 224	1230

Table 2 presents several proposed models constructed with the CNN, LSTM, and bi-LSTM. For the baseline model, we prefer VGG16 + bi-LSTM + one hot vector [61] and several models, such as ResNet [14] and InceptionV3 [62]. Thus, we re-engineered several models and embedded using Word2Vec.

Table 2. Comparison of BLEU-N scores. Results gained from experiments using the testing dataset and executed using the baseline and proposed models.

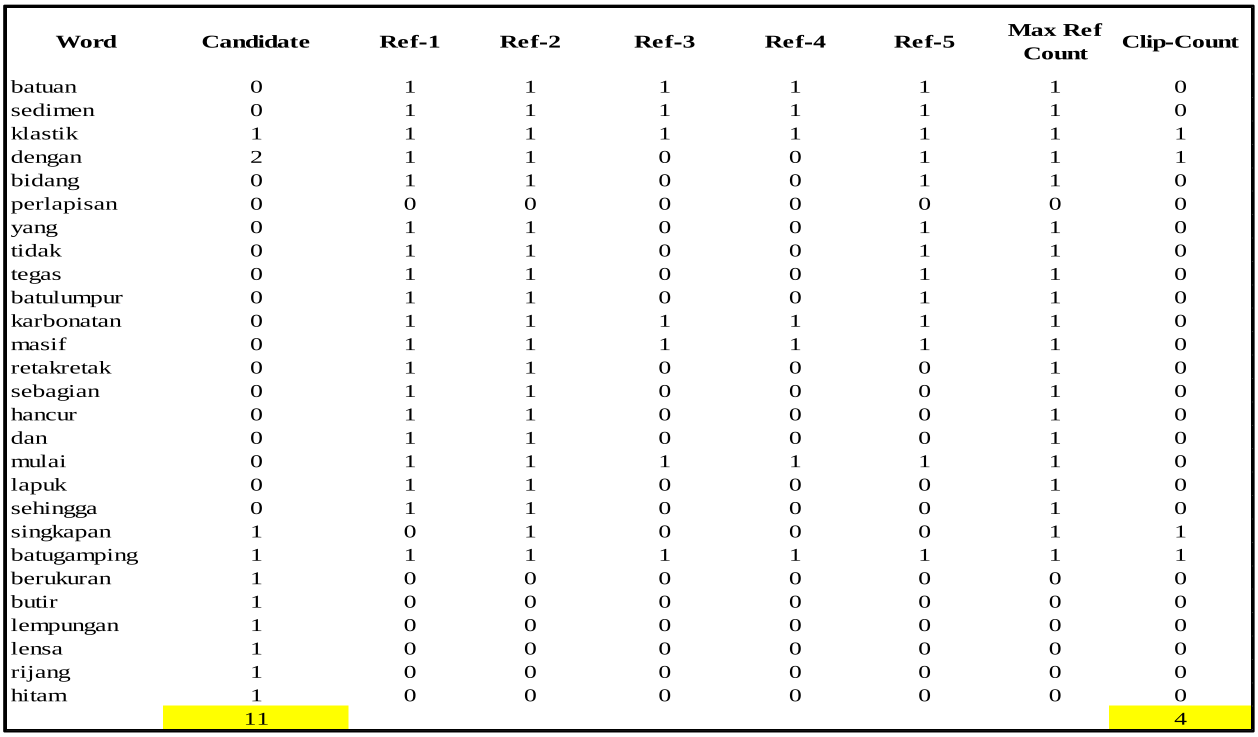
Model Caption	**BLEU-1 (Unigram)**	**BLEU-2 (bi-gram)**	**BLEU-3 (3-gram)**	**BLEU-4 (4-gram)**
VGG16 + LSTM + word2vec	0.5516	0.6048	0.6414	0.6526
ResNet50 +LSTM + Word2Vec	0.3990	0.3830	0.4030	0.3440
InceptionV3 +LSTM + Word2Vec	0.3320	0.3120	0.3300	0.2730
Ours_CNN-A + LSTM + word2vec	0.6464	0.6508	0.6892	0.6504
Ours_CNN-B + LSTM + word2vec	0.7012	0.7083	0.7312	0.7345
Ours_CNN-C + LSTM + word2vec	0.7620	0.8757	0.8861	0.8250
Ours_CNN-D+ LSTM + word2vec	0.5620	0.6578	0.7307	0.7537
CNN + LSTM + One-Hot, adapted from [2], Copyright 2015, Karpathy et al.	0.3990	0.3830	0.4050	0.3470
InceptionV3 + LSTM + One-Hot [14]	0.3760	0.3410	0.3540	0.2960
ResNet50 + LSTM + One-Hot [14]	0.4030	0.3920	0.4150	0.3590

We employed the BLEU score to measure the precision of candidate captions and reference captions. The measured BLEU scores for 500 epochs are shown in Table 2 using the validation dataset. Supposing the translation results are identical to the reference sentence and have the same length of words, then the BLEU score is 1.00. This study used a pre-trained model, Word2Vec, to obtain a feature vector for every single vocabulary word from the text processing. Using the proposed models, we achieved 306 × 100 feature dimensions and then used a flattening operation to gain 256 units. We used the Indonesian corpus ourselves to make a feature vector that related to geological captions. Table 2 shows that the B@1 score was 0.6463, 0.7012, 0.7620, and 0.5620 for CNN-A, CNN-B, CNN-C, and CNN-D, respectively. These scores were acquired from the following BLEU formula: $\text{BLEU} = BP * e^{(\sum_{i=1}^{N} W_n \log P_n)}$ with $BP = \begin{cases} 1 & \text{, if } c > r \\ e^{(1-r/c)} & \text{, if } c \leq r \end{cases}$.

This study leveraged the precise parameter metric, which was introduced by [60]. The BLEU score emphasizes precise captions using the N-gram. Proper and grammatical order is a significant requirement for clear meaning. The BLEU score measures how much a caption is similar to a geologist's descriptions.

Here, we have written an illustration for measuring a caption. For instance, we acquired a caption, such as "singkapan batugamping klastik dengan berukuran butir lempungan dengan lensa rijang" in Indonesian. We calculated the BLEU score by the following algorithm [60]:

1. Tokenize each caption $(w_1, w_2, \ldots, w_n)$;
2. Calculate the variable "count" and "clip_count" from the reference token and candidate token, see Figure 7;
3. Compute precision modification with formula $p_n = \frac{clip\_count}{word\_length} = \frac{4}{20} = 0.2$,
4. If length of candidate <= reference, calculate brevity penalty (BP) with $BP = e^{(1-\frac{r}{c})}$ $= e^{(1-\frac{20}{2})} = 0.442$, else $BP = 1$;
5. Calculate BLEU-1 and the result can be seen in Table 2.



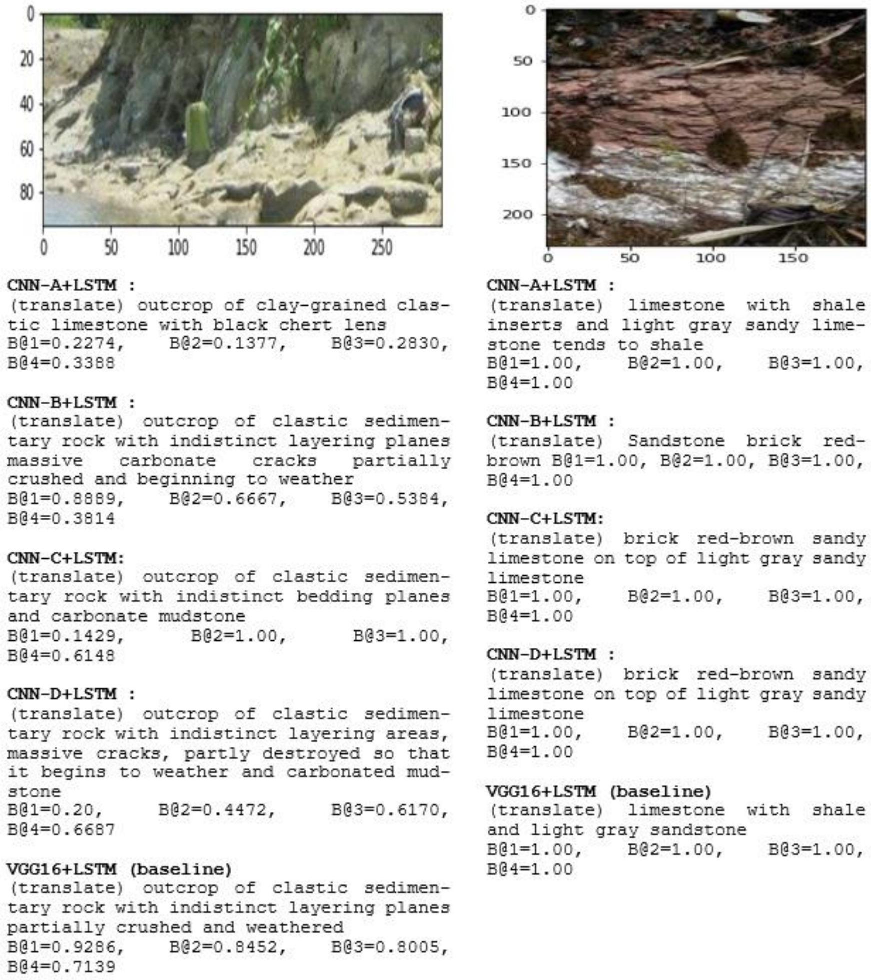
Word	Candidate	Ref-1	Ref-2	Ref-3	Ref-4	Ref-5	Max Ref Count	Clip-Count
batuan	0	1	1	1	1	1	1	0
sedimen	0	1	1	1	1	1	1	0
klastik	1	1	1	1	1	1	1	1
dengan	2	1	1	0	0	1	1	1
bidang	0	1	1	0	0	1	1	0
perlapisan	0	0	0	0	0	0	0	0
yang	0	1	1	0	0	1	1	0
tidak	0	1	1	0	0	1	1	0
tegas	0	1	1	0	0	1	1	0
batulumpur	0	1	1	0	0	1	1	0
karbonatan	0	1	1	1	1	1	1	0
masif	0	1	1	1	1	1	1	0
retakretak	0	1	1	0	0	0	1	0
sebagian	0	1	1	0	0	0	1	0
hancur	0	1	1	0	0	0	1	0
dan	0	1	1	0	0	0	1	0
mulai	0	1	1	1	1	1	1	0
lapuk	0	1	1	0	0	0	1	0
sehingga	0	1	1	0	0	0	1	0
singkapan	1	0	1	0	0	0	1	1
batugamping	1	1	1	1	1	1	1	1
berukuran	1	0	0	0	0	0	0	0
butir	1	0	0	0	0	0	0	0
lempungan	1	0	0	0	0	0	0	0
lensa	1	0	0	0	0	0	0	0
rijang	1	0	0	0	0	0	0	0
hitam	1	0	0	0	0	0	0	0
	11							4

Figure 7. Collecting tokens taken from reference and candidate captions in Indonesian. Candidate is the number of words from the caption generation, ref-n is the number of words that have a similar word as the candidate, Max ref count is a value from selecting Max (candidate, ref-n), and clip_count is a value of Min (candidate, Max ref count). The yellow color is sum of candidate and sum of cli-count.

Figure 7 express the collection of vocabulary for each reference. These tabular shows the simulation counting of every single word in the caption. Candidate column show how many words that have similar with the reference. Beside that, ref-1 trough ref-4 is the calculating number of words to compare with the candidate.

Figure 8 shows an instance of the captioning from our training models. We validated the training model using dataset validation, as shown in Figure 8. Different layers, filters, and training parameters caused different results. Varying the parameters supports the creation of an accurately predicted word that aligns with the image area [2]. It is a point of discussion because, in CNN operation, every mapping process results in a new block value that is smaller than before. Reducing the H × W × D of the CNN block avoided overfitting [5].

We compared the generated captions with ground-truth captions. Figure 8a compares five geologist references as follows: Ref-1: singkapan batuan sedimen klastik dengan bidang perlapisan yang tidak tegas, batulumpur karbonatan masif retakretak sebagian hancur dan mulai lapuk—(translation) outcrop of clastic sedimentary rock with indistinct layering, massive fractured carbonate mudstone partially crushed and beginning to weather; Ref-2: singkapan batuan sedimen klastik dengan bidang perlapisan yang tidak tegas masif retakretak sebagian hancur sehingga mulai lapuk dan batulumpur karbonatan—(translation) outcrop of clastic sedimentary rock with indistinct layering areas, massive cracks, partially destroyed so that it begins to weather and carbonate mudstone; Ref-3: singkapan batuan sedimen klastik dan batulumpur karbonatan—(translation) outcrop of clastic sedimentary rock and carbonate mudstone; Ref-4: batulumpur karbonatan dan singkapan batuan sedimen klastik—(translation) carbonate mudstone and outcrop of clastic sedimentary rock; Ref-5: singkapan batuan sedimen klastik dengan bidang perlapisan yang tidak tegas dan batulumpur karbonatan—(translation) outcrop of clastic sedimentary rock with indistinct bedding planes and carbonate mudstone.



```
CNN-A+LSTM :
(translate) outcrop of clay-grained clas-
tic limestone with black chert lens
B@1=0.2274,    B@2=0.1377,    B@3=0.2830,
B@4=0.3388

CNN-B+LSTM :
(translate) outcrop of clastic sedimen-
tary rock with indistinct layering planes
massive   carbonate   cracks   partially
crushed and beginning to weather
B@1=0.8889,    B@2=0.6667,    B@3=0.5384,
B@4=0.3814

CNN-C+LSTM:
(translate) outcrop of clastic sedimen-
tary rock with indistinct bedding planes
and carbonate mudstone
B@1=0.1429,      B@2=1.00,      B@3=1.00,
B@4=0.6148

CNN-D+LSTM :
(translate) outcrop of clastic sedimen-
tary rock with indistinct layering areas,
massive cracks, partly destroyed so that
it begins to weather and carbonated mud-
stone
B@1=0.20,     B@2=0.4472,     B@3=0.6170,
B@4=0.6687

VGG16+LSTM (baseline)
(translate) outcrop of clastic sedimen-
tary rock with indistinct layering planes
partially crushed and weathered
B@1=0.9286,    B@2=0.8452,    B@3=0.8005,
B@4=0.7139
```

(a)

```
CNN-A+LSTM :
(translate)  limestone  with  shale
inserts and light gray sandy lime-
stone tends to shale
B@1=1.00,    B@2=1.00,    B@3=1.00,
B@4=1.00

CNN-B+LSTM :
(translate)  Sandstone  brick  red-
brown B@1=1.00, B@2=1.00, B@3=1.00,
B@4=1.00

CNN-C+LSTM:
(translate)  brick  red-brown  sandy
limestone on top of light gray sandy
limestone
B@1=1.00,    B@2=1.00,    B@3=1.00,
B@4=1.00

CNN-D+LSTM :
(translate)  brick  red-brown  sandy
limestone on top of light gray sandy
limestone
B@1=1.00,    B@2=1.00,    B@3=1.00,
B@4=1.00

VGG16+LSTM (baseline)
(translate)  limestone  with  shale
and light gray sandstone
B@1=1.00,    B@2=1.00,    B@3=1.00,
B@4=1.00
```

(b)

Figure 8. Examples of generated captions for geological captioning, translation into English. B@N is the BLEU score. (**a**) Sedimentary rock image; (**b**) limestone image on a different surface.

In Figure 8b, we compared five geologist captions by referring to the ground truth: Ref-1: batugamping dengan sisipan serpih—(translation) limestone with shale insert; Ref-2: batugamping pasiran berwarna merah bata/kecoklatan—(translation) brick red/brown sandy limestone; Ref-3: batugamping pasiran berwarna merah bata/kecoklatan menindih batugamping pasiran berwarna kelabu cerah—(translation) brick red/brown sandy limestone on top of light gray sandy limestone; Ref-4: Batugamping bersisipan serpih dan batugamping pasiran berwarna kelabu cerah—(translation) limestone with shale inserts and light gray sandy limestone. Ref-5: Batugamping bersisipan serpih dan batugamping pasiran berwarna kelabu cerah cenderung menyerpih—(translation) limestone with shale inserts and light gray sandy limestone tends to shale.

4. Discussion

There are several discussions to be presented. The objective was to find the best captions by obtaining a high BLEU score. We followed the proposed method as a strategy for observing the object in the background [60]. The experiments proved to have different outcomes. A change in the layers can cause various outcomes, such as in the resulting caption, training parameters, and time. An event in the convolution process and size filtering can result in different feature maps. The number of channels the CNN has influences the feature maps' colors, shapes, and sizes. We set a significant number of channels, which provided opportunities for various gradations of colors. On rock images, observing the color is necessary to distinguish the content of the rocks; even if the rocks have a similar color, the content is probably different [53].

VGG16 successfully identified the object and predicted the classification [63–65]. Meanwhile, our models successfully classified the rock object within an efficient time, particularly CNN-B and CNN-D [66]. Table 1 shows that VGG16 is suitable for our rock images, but the model has a longer processing time compared to the CNN-B and CNN-D models.

Table 3 compares the accuracy and loss of the models. The results showed that VGG16 achieved the best accuracy and that several layers and structure processes applied to the CNN were robust in differentiating color.

Table 3. Comparison of the accuracy and loss between our models and the baseline.

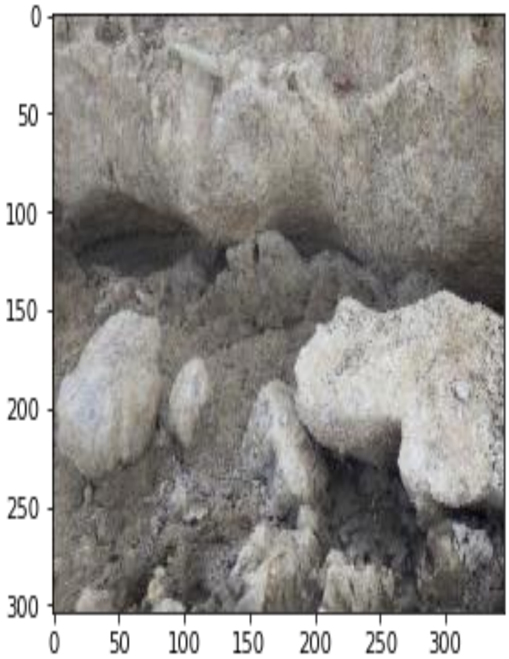
Evaluation	CNN-A	CNN-B	CNN-C	CNN-D	VGG16
Accuracy	0.9137	0.9148	0.9178	0.9206	0.9228
Loss	0.1763	0.1764	0.1638	0.1457	0.1464

The theoretical concept of VGG16 tries to extract the image from deeper layers and provide outcomes with various features [65]. In image recognition, the detection of similar features is an essential task to achieve the best accuracy. In our problem domain, if the image features concatenate with text features, then many pairwise mistakes can occur and cause the wrong predicted word. Many feature map differences cannot guarantee that the captions will be optimized. The spread of various colors causes a bias when the object area pairs with the sentence. The effect that many features create causes probable bias in predicting words. We used a limited object in our domain, which is why the results of pairwise testing a sequence of image descriptors and feature vector text is important.

LSTM, as a word generator, was chosen as the language model because of its ability to separate outcomes into three gates. The LSTM generates words based on previous words by relating to the corpus and selecting the best probability word from many produced words. The previous word makes the LSTM machine more powerful in predicting the next word [67]. This study found error captions for the VGG16+word2vec+LSTM model when predicting a word. Simonyan models have successfully generated a word when the object is a person, an animal, a car, etc. [57]. In contrast, the models experience an error in predicting a word when the object appears in the background.

The likelihood function $\log P(w_t|w_{1:t-1}, I; \theta)$ is important for making a true predicted word when generating a word. Re-engineering of the log-likelihood function will result in different captions. We acquired a loss value of $0 \leq \sum_{I,S\epsilon X} \sum_{t=1}^{|s|} \log P(w_t|w_{1:t-1}, I; \theta) \leq 0.1358$. This means that the function was successful in maximizing the detection of images and predicting the words by minimizing the residual loss values [68].

Figure 9 shows an error caption and compares VGG16 + Word2Vec + LSTM to our models [69]. This study successfully presented an error result from the baseline. It proves that many CNN layers do not always acquire the best outcome; indeed, the result is sometimes a mistake. The image in Figure 9 is of sandstone. Nevertheless, the baseline presented a mistake caption of chert. Many parameters produced the feature that caused VGG16 difficulty to pairwise between the word and the feature maps. Geological rock images always show a variety of color within the rocks and color differentiation is necessary to separately identify the rocks.



CNN-A+LSTM:
(translation) outcrop alternating chert and shale sandstones
B@1=0.67, B@2=0.52, B@3=0.44, B@4=0.51

CNN-B+LSTM:
(translation) gray andesite B@1=0.195, B@2=0.394, B@3=0.514, B@4=0.550

CNN-C+LSTM:
(translation) alternating fine sandstone with siltstone
B@1=0.60, B@2=0.50, B@3=0.66, B@4=0.622

CNN-D+LSTM:
(translation) compact sandstone gray cracks there is also a layer of reddish color
B@1=0.10, B@2=0.32, B@3=0.50, B@4=0.56

VGG16+LSTM (baseline):
(translation) black chert cracks filled with silica mineral
B@1=0.00, B@2=0.00, B@3=0.00, B@4=0.00

Figure 9. An error created by the baseline model, producing a different rock type. The dominant rock in the image is sandstone, whereas the baseline produced chert and CNN-B produces gray andesite.

Some images could not be adequately classified by VGG16, thus, resulting in a low BLEU score. However, it is understandable to assume that VGG16 is accurate because it has a deeper layer than our models. The impact of this is that many feature maps eventually become biased when rephrasing image feature extraction and word embedding. It is known that image feature extraction becomes decisive regarding feature identification. The number of CNN layers can convey the success of feature extraction and many receptive field channels supply more space for feature assortments [70,71]. It can help to recognize objects to a certain degree in an image. Meanwhile, the pooling layer also helps the feature map values to avoid overfitting.

This study shows a need to recognize and analyze the image relating to captions by using image preprocessing, such as reducing image size. In the caption, it is necessary to pay attention to the text marks when reading. Sometimes, captions use the "-" sign to make adjectives or derivatives of rock types. In text preprocessing, the text is cleaned by removing the stop word, sign, number, and symbols. Figure 3 shows the word vectorizing using Word2Vec as a feature text. The difference between one hot vector and word2vec lies in the matrix values. One hot vector consists of a 0 or 1 value and has a length the same as the defined word length [28,31]. On the other hand, word2vec creates a decimal value with the defined length and dimensions [25,69].

Regarding the process, the annotation of the rocks was accompanied by their properties, such as carbonate mudstone and clay sandstone [72]. The classification and interpretation of images created a caption based on feature maps and the text feature. This scheme is an essential part of captioning because this research's target was a caption similar to the geologist's description [53].

5. Conclusions

Our models outperform the architecture of the baseline model. A CNN (32,5) with a 5 × 5 filter and 32 channels produced a meaningful caption. The metric of the model was directed more toward the precision of the caption than accuracy. The accuracy is just needed to measure the image classification and how similar the factual feature map extraction compares with the actual feature maps. The experiments proved that shallow layers effectively solved our domain problem. Our proposed CNN-A, CNN-B, CNN-C, and CNN-D models used BLEU score methods for the N-gram. The BLEU scores achieved for B@1 were 0.5515, 0.6463, 0.7012, 0.7620, and 0.5620, respectively. B@2 showed scores of 0.6048, 0.6507, 0.7083, 0.8756, and 0.6578, respectively. B@3 had scores of 0.6414, 0.6892, 0.7312, 0.8861, and 0.7307, respectively. Finally, B@4 showed scores of 0.6526, 0.6504, 0.7345, 0.8250, and 0.7537, respectively. The CNN-D architecture encouraged our model to produce a high B@4 score of 0.8250 but it had a time deficiency. The BLEU score measurement was dependent on precision and word embedding. The combination of the CNN and Word2vec embedding increased the speed and produced precision words. Construction of the caption using the beam search supported the creation of proper sentences for the caption. There are several considerations for building the architectures, such as the optimum number of layers, the precise ReLu function, a suitable SoftMax function, and an ADAM optimizer tuned to acquire good results. On the other hand, the accuracy score is used to measure how precisely the image detection matches the image references. The metrics used for measuring the success of image detection are similar for captioning.

Relating to the results, we discovered several challenges for future research. This study did not just find layers, filters, strides, and pooling methods but also proposed language generators. Language models, such as structured language and paraphrasing models, are important subjects of research. Assembling captioning based on geological sentence arrangement, geological sentence representations, and assembling words by geological sentence tagging is a challenging topic. In all captioning models, the target is word precision, which is an indicator of success when generating a caption, with a high BLEU score or other language metric.

Author Contributions: Conceptualization, A.N., R.M. and M.L.K.; methodology, A.N.; software, A.N.; validation, R.M. and M.L.K.; supervision, R.M. and M.L.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Ministry of Research, Technology and Higher Education, Republic of Indonesia, grant number 083/E5/PG.02.00.PT/2022.

Acknowledgments: The authors would like to express their great appreciation to Joko Wahyudiono and Fitriani Agustin for their valuable and constructive suggestions during the planning and development of this research work. Their willingness to give their time so generously has been very much appreciated in supplying the image materials and captions of geological rocks.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, NV, USA, 3–6 December 2012; Volume 25, pp. 1–9. [CrossRef]
2. Karpathy, A.; Fei-Fei, L. Deep Visual-Semantic Alignments for Generating Image Descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 664–676. [CrossRef]
3. Lebret, R.; Pinheiro, P.O.; Collobert, R. Phrase-Based Image Captioning. In Proceedings of the 32nd International Conference on Machine Learning, ICML, Lille, France, 6–11 July 2015; pp. 2085–2094.
4. Boureau, Y.; Ponce, J.; Fr, J.P.; Lecun, Y. A Theoretical Analysis of Feature Pooling in Visual Recognition. In Proceedings of the International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 111–118.
5. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
6. Batra, V.; He, Y.; Vogiatzis, G. Neural Caption Generation for News Images. In Proceedings of the LREC 2018—11th International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2019; pp. 1726–1733.
7. Chen, X.; Zhang, M.; Wang, Z.; Zuo, L.; Li, B.; Yang, Y. Leveraging Unpaired Out-of-Domain Data for Image Captioning. *Pattern Recognit. Lett.* **2020**, *132*, 132–140. [CrossRef]
8. Qi, M.; Qin, J.; Li, A.; Wang, Y.; Luo, J.; Van Gool, L. StagNet: An Attentive Semantic RNN for Group Activity Recognition. *Lect. Notes Comput. Sci. (Incl. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinform.)* **2018**, *11214 LNCS*, 104–120. [CrossRef]

9. He, X.; Yang, Y.; Shi, B.; Bai, X. Neurocomputing VD-SAN: Visual-Densely Semantic Attention Network for Image Caption Generation. *Neurocomputing* **2019**, *328*, 48–55. [CrossRef]
10. Chen, S.; Zhao, Q. Boosted Attention: Leveraging Human Attention for Image Captioning. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Volume 1, pp. 68–84.
11. Ghosh, S.; Das, N.; Das, I.; Maulik, U. Understanding Deep Learning Techniques for Image Segmentation. *ACM Comput. Surv.* **2019**, *52*, 1–35. [CrossRef]
12. Armi, L.; Fekri-ershad, S. Texture image analysis and texture classification methods—A review. *arXiv* **2019**, arXiv:1904.06554.
13. Soh, M. *Learning CNN-LSTM Architectures for Image Caption Generation*; Stanford University: Stanford, CA, USA, 2016; pp. 1–9.
14. Bhatia, Y.; Bajpayee, A.; Raghuvanshi, D.; Mittal, H. Image Captioning Using Google's Inception-Resnet-v2 and Recurrent Neural Network. In Proceedings of the 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 8–10 August 2019; Volume 2019, pp. 1–6.
15. Mao, J.; Yuille, A. Deep captioning with multimodal recurrent neural networks (M-RNN). *arXiv* **2015**, arXiv:1412.6632.
16. Junhua, M.; Wei, X.; Yi, Y.; Jiang, W.; Zhiheng, H.; Yuille, A. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). In Proceedings of the ICLR, San Diego, CA, USA, 7–9 May 2015; pp. 1–17.
17. Xiao, X.; Wang, L.; Ding, K.; Xiang, S.; Pan, C. Dense Semantic Embedding Network for Image Captioning. *Pattern Recognit.* **2019**, *90*, 285–296. [CrossRef]
18. Xu, K.; Ba, J.L.; Kiros, R.; Courville, A. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6 July–11 July 2015; pp. 2048–2057.
19. Donahue, J.; Hendricks, L.A.; Rohrbach, M.; Venugopalan, S.; Guadarrama, S.; Saenko, K.; Darrell, T. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–14.
20. He, X.; Shi, B.; Bai, X.; Xia, G.; Zhang, Z. Image Caption Generation with Part of Speech Guidance. *Pattern Recognit. Lett.* **2017**, *119*, 229–237. [CrossRef]
21. Wang, L.; Chu, X.; Zhang, W.; Yiwei, W.; Weichen, S.; Chunlei, W. Social Image Captioning: Exploring Visual Attention and User Attention. *Sensors* **2018**, *18*, 646. [CrossRef]
22. Lee, H.; Yoon, S.; Dernoncourt, F.; Kim, D.S.; Bui, T.; Jung, K. ViLBERTScore: Evaluating Image Caption Using Vision-and-Language BERT. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*; Eval4NLP; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 34–39.
23. Weijie, S.; Xizhou, Z.; Yue, C.; Bin, L.; Lewei, L. Vl-Bert: P Re-Training of G Eneric V Isual. In Proceedings of the ICLR, Addis Ababa, Ethiopia, 26–30 April 2020; pp. 1–16.
24. Plummer, B.A.; Liwei, W.; Cervantes, C.M.; Caicedo, J.C.; Hockenmaier, J.; Lazebnik, S. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–17 December 2015; pp. 2641–2649.
25. Yao, T.; Pan, Y.; Li, Y.; Qiu, Z.; Mei, T. Boosting Image Captioning with Attributes. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4904–4912.
26. Nur, K.; Effendi, J.; Sakti, S.; Adriani, M.; Nakamura, S. Corpus Construction and Semantic Analysis of Indonesian Image Description. In Proceedings of the 6th Workshop on Spoken Language Technologies for Under-Resourced Languages, Gurugram, India, 29–31 August 2018; pp. 20–24.
27. Su, J.; Tang, J.; Lu, Z.; Han, X.; Zhang, H. A Neural Image Captioning Model with Caption-to-Images Semantic Constructor. *Neurocomputing* **2019**, *367*, 144–151. [CrossRef]
28. Wang, C.; Yang, H.; Meinel, C. Image Captioning with Deep Bidirectional LSTMs and Multi-Task Learning. *ACM Trans. Multimed. Comput. Commun. Appl.* **2018**, *14*, 3115432. [CrossRef]
29. Ordenes, F.V.; Zhang, S. From Words To Pixels: Text And Image Mining Methods For Service Research. *J. Serv. Manag.* **2019**, *30*, 593–620. [CrossRef]
30. Nezami, O.M.; Dras, M.; Wan, S.; Nov, C.V. SENTI-ATTEND: Image Captioning Using Sentiment and Attention. *arXiv* **2018**, arXiv:1811.09789.
31. Aneja, J.; Deshpande, A.; Schwing, A.G. Convolutional Image Captioning. In *Computer Vision and Pattern Recognition*; Computer Vision Foundation, 2017; pp. 5561–5570. Available online: https://arxiv.org/abs/1711.09151 (accessed on 12 September 2022).
32. Wang, A.; Hu, H.; Yang, L. Image Captioning with Affective Guiding and Selective Attention. *ACM Trans. Multimed. Comput. Commun. Appl.* **2018**, *14*, 1–15. [CrossRef]
33. Tan, Y.H.; Chan, C.S. Phrase-Based Image Caption Generator with Hierarchical LSTM Network. *Neurocomputing* **2019**, *333*, 86–100. [CrossRef]
34. Li, N.; Chen, Z. Image Captioning with Visual-Semantic LSTM. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 793–799.
35. Tan, E.; Lakshay, S. "Neural Image Captioning". Available online: https://arxiv.org/abs/1907.02065 (accessed on 12 September 2022).
36. Zhu, Z.; Xue, Z.; Yuan, Z. Think and Tell: Preview Network for Image Captioning. In Proceedings of the British Machine Vision Conference 2018 (BMVC 2018), Newcastle, UK, 3–6 September 2018; pp. 1–12.
37. He, C.; Hu, H. Image Captioning with Visual-Semantic Double Atention. *ACM Trans. Multimed. Comput. Commun. Appl.* **2019**, *15*, 1–16. [CrossRef]

38. Mullachery, V.; Motwani, V. Image Captioning. *arXiv* **2018**, arXiv:1805.09137.
39. Li, X.; Song, X.; Herranz, L.; Zhu, Y.; Jiang, S. Image Captioning with Both Object and Scene Information. In *Proceedings of the 24th ACM International Conference on Multimedia*; ACM: Amsterdam, The Netherlands, 2016; pp. 1107–1110.
40. Mathews, A. *Automatic Image Captioning with Style*; ANU Open Research, 2018. Available online: https://openresearch-repository.anu.edu.au/bitstream/1885/151929/1/thesis_apm_01_11_18.pdf (accessed on 12 September 2022).
41. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 652–663. [CrossRef] [PubMed]
42. Mun, J.; Cho, M.; Han, B. Text-Guided Attention Model for Image Captioning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4233–4239.
43. Tran, A.; Mathews, A.; Xie, L. Transform and Tell: Entity-Aware News Image Captioning. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13032–13042. [CrossRef]
44. Herdade, S.; Kappeler, A.; Boakye, K.; Soares, J. Image Captioning: Transforming Objects into Words. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–11.
45. Zhu, Y.; Li, X.; Li, X.; Sun, J.; Song, X.; Jiang, S. Joint Learning of CNN and LSTM for Image Captioning. In Proceedings of the CEUR Workshop Proceedings, Évora, Portugal, 5–8 September 2016; Volume 1609, pp. 421–427.
46. Gan, C.; Gan, Z.; He, X.; Gao, J. StyleNet: Generating Attractive Visual Captions with Styles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 955–964.
47. Kinghorn, P.; Zhang, L.; Shao, L. A Region-Based Image Caption Generator with Refined Descriptions. *Neurocomputing* **2018**, *272*, 416–424. [CrossRef]
48. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef]
49. Kocarev, L.; Makraduli, J.; Amato, P. Rock classification in petrographic thin section images based on concatenated convolutional neural networks. *Earth Sci. Inform.* **2005**, *9*, 497–517.
50. Lepistö, L. Rock Image Classification Using Color Features in Gabor Space. *J. Electron. Imaging* **2005**, *14*, 040503. [CrossRef]
51. Lepistö, L.; Kunttu, I.; Autio, J.; Visa, A. Rock Image Classification Using Non-Homogenous Textures and Spectral Imaging. WSCG. 2003, pp. 1–5. Available online: http://wscg.zcu.cz/wscg2003/Papers_2003/D43.pdf (accessed on 12 September 2022).
52. Nursikuwagus, A. Multilayer Convolutional Parameter Tuning Based Classification for Geological Igneous Rocks. In Proceedings of the International Conference on ICT for Smart Society (ICISS); Information Technology Research Group of the School of Electrical Engineering and Informatics, Bandung, Indonesia, 10–11 August 2021.
53. Ran, X.; Xue, L.; Zhang, Y.; Liu, Z.; Sang, X.; He, J. Rock Classification from Field Image Patches Analyzed Using a Deep Convolutional Neural Network. *Mathematics* **2019**, *7*, 755. [CrossRef]
54. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the 1st International Conference on Learning Representations, ICLR 2013—Workshop Track Proceedings, Scottsdale, AZ, USA, 2–4 May 2013; pp. 1–12.
55. David, T.A. *The University of South Alabama GY480 Field Geology Course*; University of South Alabama: Mobile, AL, USA, 2021.
56. Chen, J.; Yang, T.; Zhang, D.; Huang, H.; Tian, Y. Deep Learning Based Classification of Rock Structure of Tunnel Face. *Geosci. Front.* **2021**, *12*, 395–404. [CrossRef]
57. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arxiv* **2015**, arXiv:1409.1556.
58. Ren, W.; Zhang, M.; Zhang, S.; Qiao, J.; Huang, J. Identifying Rock Thin Section Based on Convolutional Neural Networks. In Proceedings of the 2019 9th International Workshop on Computer Science and Engineering (WCSE 2019), Hong Kong, China, 15–17 June 2019; pp. 345–351. [CrossRef]
59. Wu, C.; Wei, Y.; Chu, X.; Su, F.; Wang, L. Modeling Visual and Word-Conditional Semantic Attention for Image Captioning. *Signal Process. Image Commun.* **2018**, *67*, 100–107. [CrossRef]
60. Papineni, K.; Roukos, S.; Ward, T.; Wei-Jing, Z. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
61. Wang, C.; Yang, H.; Bartz, C.; Meinel, C. Image Captioning with Deep Bidirectional LSTMs. In Proceedings of the MM 2016-ACM Multimedia Conference, New York, NY, USA, 15–19 October 2016; pp. 988–997. [CrossRef]
62. Szegedy, C.; Vanhoucke, V.; Shlens, J. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the Computer Vision Fundation, Columbus, OH, USA, 23–28 June 2014; pp. 2818–2826.
63. Fan, G.; Chen, F.; Chen, D.; Li, Y.; Dong, Y. A Deep Learning Model for Quick and Accurate Rock Recognition with Smartphones. *Mob. Inf. Syst.* **2020**, *2020*, 7462524. [CrossRef]
64. Robson, B.A.; Bolch, T.; MacDonell, S.; Hölbling, D.; Rastner, P.; Schaffer, N. Automated Detection of Rock Glaciers Using Deep Learning and Object-Based Image Analysis. *Remote Sens. Environ.* **2020**, *250*, 112033. [CrossRef]
65. Feng, J.; Qing, G.; Huizhen, H.; Na, L. Feature Extraction and Segmentation Processing of Images Based on Convolutional Neural Networks. *Opt. Mem. Neural Netw. (Inf. Opt.)* **2021**, *30*, 67–73. [CrossRef]
66. Nursikuwagus, A.; Munir, R.; Khodra, M.L. Multilayer Convolutional Parameter Tuning Based Classification for Geological Igneous Rocks. In Proceedings of the ICISS, Patna, India, 16–20 December 2021.

67. Wu, Q.; Shen, C.; Wang, P.; Dick, A.; Van Den Hengel, A. Image Captioning and Visual Question Answering Based on Attributes and External Knowledge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1367–1381. [CrossRef]
68. Kingma, D.P.; Ba, J.L. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015-Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
69. You, Q.; Jin, H.; Wang, Z.; Fang, C.; Luo, J. Image Captioning with Semantic Attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4651–4659.
70. Ding, G.; Chen, M.; Zhao, S.; Chen, H.; Han, J. Neural Image Caption Generation with Weighted Training and Reference. *Cogn. Comput.* **2018**, *11*, 763–777. [CrossRef]
71. Cao, P.; Yang, Z.; Sun, L.; Liang, Y.; Yang, M.Q.; Guan, R. Image Captioning with Bidirectional Semantic Attention-Based Guiding of Long Short-Term Memory. *Neural Process. Lett.* **2019**, *50*, 103–119. [CrossRef]
72. Contreras, J.V. *Supervised Learning Applied to Rock Type Classification in Sandstone Based on Wireline Formation Pressure Data*; AAPG/Datapages, Inc., 2020. Available online: https://www.searchanddiscovery.com/pdfz/documents/2020/42539contreras/ndx_contreras.pdf.html (accessed on 12 September 2022). [CrossRef]