Image to Text Synthesis with VGG-16 CNN and LSTM

Edward Kevin Winata
Computer Science Department Faculty of Computing and Media
Bina Nusantara International University
Central Jakarta, Indonesia 11480
edward.winata@binus.ac.id

Jude Joseph Lamug Martinez
Computer Science Department Faculty of Computing and Media
Bina Nusantara International University
Central Jakarta, Indonesia 11480
jude.martinez@binus.ac.id

***Abstract*—One of the more difficult tasks at the nexus of computer vision and natural language processing(NLP) is image decomposition into natural language, also referred to as prompts. This study presents a novel method for tackling this job by combining Long Short - Term Memory(LSTM) networks for text synthesis with Convolutional Neural Networks(CNNs) for image feature extraction. After high - level visual features from input photos are extracted using the VGG-16 CNN architecture, the information is then fed into LSTM encoder networks to produce descriptive texts. Using common evaluation measures like BLEU(Bilingual Evaluation Understudy), METEOR(Metric for Evaluation of Translation with Explicit Ordering), and CIDEr(Consensus - based Image Description Evaluation), the efficacy of the suggested method is assessed on benchmark datasets, such as other Flickr datasets with larger scales. The suggested strategy outperforms baseline techniques, as shown by experimental findings, obtaining exceptional performance indexes. Furthermore, the model's capacity to generate semantically coherent and contextually appropriate descriptions is demonstrated by quantitative examination of generated texts. The suggested methodology advances the field of multimodal artificial intelligence and establishes the groundwork for future developments in content creation, picture comprehension and human - computer interaction.**

***Index Terms*—Natural language processing, convolutional neural networks, long short - term memory, multimodal learning, benchmark datasets, Bilingual Evaluation Understudy, Metric for Evaluation of Translation with Explicit Ordering, Consensus - based Image Description Evaluation, Tensor Processing Unit.**

I. Introduction

An essential research subject within the field of computer vision and natural language processing operations involves decomposing images into natural language. Enabling robots to comprehend and convey the substance of visual situations is the aim of generating compatible prompts for each given image data, which aims to automatically generate descriptive and contextually appropriate written descriptions in the form of natural language, or text meant for input photos. Many applications, such as content - based picture retrieval systems, assistive technologies for the blind, and human-computer interfaces, will benefit greatly from this task's practical implications [1].

Implementing recent developments in convolutional neural networks and recurrent neural networks(RNNs), considerable strides have been achieved in the development of deep learning based methods for providing natural language prompts or captions from images in recent years. CNNs are commonly utilized for the extraction of high - level visual features from input images, including spatial hierarchies and semantic information [2]. On the basis of the extracted features, especially RNNs and long short - term memory networks are used to produce coherent and contextually relevant captions [3].

Manually designed feature extraction techniques and rule based caption synthesis algorithms dominated early approaches for generating text from images. But these methodologies frequently failed to grasp the depth and complexity of visual content, which prompted the creation of data - driven strategies built on deep learning technologies. The efficacy of deep neural networks for generating caption text from image tasks has been established in notable publications by Karpathy and Fei-Fei(2015), Vinyals(2015) and Xu(2015). These vital research breakthroughs set the foundation for further study in the subject.

The creation of varied and semantically significant word tokens, the alignment of visual and textual modalities and the incorporation of contextual information into the captioning process are some of the issues that still face the area of image to text tasks, despite its advancements [4]. Furthermore, evaluating the models implemented for the task presents a unique set of difficulties that call for the creation of thorough and reliable assessment measures that reliably grade the relevance and caliber of generated word tokens.

In this research instance, we present a kind of method for image decomposition into plain language by combining LSTM networks for word token generation with VGG-16 CNN to extract vital image features. By expanding on earlier research, we aim to address common issues that are found within image captioning tasks as well as develop a multimodal AI and language comprehension [5]. Experimental results show that our suggested strategy performs collectively with baseline methods in terms of metrics as well as accuracy tests [6]. The effectiveness of the approach is assessed through a split section of the used Flickr8K dataset meant for testing.

II. Related Works

This section examines a number of important research studies that employ methodologies using similar techniques or other related modus operandi to our suggested method for image decomposition into natural language in terms of problem breakdown. The following works demonstrate developments and breakthroughs in the field of image to text tasks by bridging the gap between textual and visual modalities with the help of deep learning algorithms.

The first research work that we conclude to have shared a common technique was published in 2015 by Karpathy and Fei - Fei. A deep visual - semantic alignment model that aligns phrases with image regions was introduced by the two driven researchers. Their model processes textual descriptions using bidirectional RNNs and extracts visual characteristics using a combination of CNNs. To achieve the best possible alignment between the textual and visual data, a structured objective is used, leading to a notable improvement in image captioning performance which is another area of work employing similar procedures. Our use of VGG-16 CNN for extracting detailed picture features and LSTMs for producing coherent captions is essential to our study, which lays the groundwork for how aligning visual and textual aspects can improve production of high - quality word tokens.

Another one is a work which delves within the area of visual questions, based on external attributes and knowledge. The author of the work presented a model for image captioning and visual question answering that takes characteristics and outside knowledge into account [7]. In order to provide text data that are more accurate and contextually relevant, the model takes attribute characteristics from photos and combines them with external knowledge sources. This method emphasizes how important it is to use information more than just raw visual data, which motivates us to think about using attention mechanisms and outside contextual data to enhance caption production.

Moreover, a paper from 2017 authored by Lin discussed the advantages of Feature pyramids with rich semantics at all scales, built using the proposed Feature Pyramid Networks, which improve object detection capability of the research product. While this pyramid approach feature is primarily focused on object recognition, it might be able to be integrated into image captioning works to improve the capture of multi - scale visual information and produce contextually richer and more detailed image representations. This information suggests that more opportunities may be present for including more multi - scale features into our CNN based feature extraction process in future updates.

In addition, Yang presented a different strategy by using Stacked Attention Networks(SANs) to generate replies to questions concerning specific images inside a dataset for the purpose of visual question answering. The SAN model then will iteratively focus on pertinent areas of the image by using multiple levels of attention [8]. Focus is sharpened on the areas of the image that are most important for providing an answer to the question by each attention layer. This method is pertinent to our study since it shows how numerous attention layers can be useful. We consider applying this method to update our CNN model in order to gradually sharpen the focus on pertinent image aspects, if applicable in future updates.

Last but not least, Spatial, channel - wise and word - level are the three categories of attention mechanisms that Yao presented in his Triple - Attention Network research [9]. To provide in - depth explanation, this network mostly concentrates on the finer elements found in photos. The work showcases the advantages of including several attention mechanisms to capture more subtle details and improve the specificity and depth of output captions. Our approach on the other hand employs a 16 layer CNN and a single LSTM network with basic attention.

III. Methodology

A. Information on Dataset

This paper acknowledges the fact that it implements the Flickr8K dataset, authored by Hodosh, Young and Hockenmaier [10]. Flickr8K contains 8092 images and up to five different textual caption tokens for each image. Such annotations are essential for continued progress in automatic image description and grounded language understanding. They enable us to define a new benchmark for localization of textual entity mentions in an image. We present a strong baseline for this task that combines an image - text embedding, detectors for common objects and a bias towards selecting larger objects. While our baseline prototype displays some frequent inaccuracies compared to other more complex and advanced models, we show that its gains cannot be easily parlayed into improvements on such tasks as image - sentence retrieval, thus underlining the limitations of current methods and the need for further research.

For more information, the Flickr8K dataset is significantly smaller compared to other available datasets like Flickr30K or MSCOCO, which contain 30,000 and 330,000 images respectively. The choice of Flickr8K in this study is primarily driven by the balance it strikes between dataset size and the computational resources required for training, making it an efficient option for experimenting with and training our deep learning models within a reasonable time frame, Although there was an occasion where we employed a publicly available tensor processing unit provided by Kaggle for training.

Data science researchers would frequently use the Flickr8K dataset as benchmark and other experimentations in image to text tasks. With the aim of bridging the gap between visual and textual data, the dataset was established to aid in the creation and evaluation of models that may produce descriptions in the form of word tokens for images.

Further technical details are as follows :

- Number of Images : 8,092
- Number of Caption Tokens : 40,460(5 caption tokens per image)
- Image Resolution : Varies, typically high - resolution images which have been downsampled for computational efficiency
- File Format : JPEG for images, TXT for caption tokens
- File Size : Approximately 1.05 GB

For the purposes of training, validation and testing in the task, the dataset is then split into three separate subsets :

- Training Set : 7,092 images
- Test Set : 1,000 images

These splits ensure that models can be evaluated on unseen data, providing a clear measure of their performance and generalization capabilities.

B. Pre - Processing Mechanisms

The first step in getting the dataset ready to be passed on to deep learning models is to efficiently pre - process both our word token and image meta data. This section describes the comprehensive approach used to convert the Flickr8K dataset's raw photos and word tokens into formats that can be used to train convolutional and recurrent models. We outline the process for loading, scaling and feature extraction from image data using a pre - trained VGG-16 model.

We address the tasks of cleaning and tokenizing text, mapping word tokens to images and getting sequences ready for input into an LSTM model for text data. To guarantee that the data is consistent, standardized, and prepared for successful and efficient model training, It is critical that this step is performed as efficient and flawless as possible.

Within the following, we detail the pre - processing steps applied to images and caption texts before they are fed into the neural network models.

1) **Images**
   a) Loading and Resizing Images
      i) Each image is imported using the $load_img$ method from the TensorFlow Keras preprocessing package to maintain consistent input dimensions.
      ii) Images are then scaled to 224 by 224 pixels to fit the desired input size of the VGG16 model.
   b) Conversion to Numpy Array for Further Steps
   c) Reshaping for Model Compatibility
      i) Array size dimension is incorporated into the rebuilt picture array, resulting in a shape of (1, 224, 224, 3).
   d) Convolutional Pre - processing
      i) Perform mean subtraction and scaling by implementing the $preprocess_input$ method specific only to the convolutional model.
   e) Feature Extraction and Saving
      i) To extract high - level features, pre - processed image arrays are passed into our convolutional model, omitting fully linked layers. For further use, these attributes are kept in a hashable object.
      ii) Features are then saved into a separate file using the pickle module for efficient storage and retrieval.

2) **Caption Texts**
   a) Loading Captions
      i) Caption text tokens are read from the text metadata and split into individual lines.
   a) Mapping to Images
      i) This step distinguishes image IDs from caption tokens. Commas are used to separate each line, then captions are mapped to appropriate image IDs.
   a) Cleaning
      i) A user - custom function converts caption tokens to lowercase, removes special characters and digits and replaces multiple spaces with one single space.
      ii) Suffix and prefix tokens denoting start($startseq$) and end($endseq$) are added to each caption text to denote the beginning and end of a sequence.
   a) Further Tokenization and Padding
      i) Each caption token is converted into a unique integer by the use of a class belonging to the Keras module named $Tokenizer$.
      ii) Sequences are padded to ensure they all have the same length, which is necessary for our LSTM model.
   a) Batch Allocation for Training Set
      i) Tokenized and padded sequences are split into input to output pairs, with the condition that the input is a sequence of words, and the output is the next word in the sequence.
      ii) These pairs are then generated in batches to manage memory efficiently during training. Although in cases where training sessions would employ a tensor processing unit or additional GPUs, such procedure would not be required and becomes optional.
      iii) Moreover, an in - depth analysis of individual batches was conducted to view caption tokens that appear more and less frequently compared side by side. We included this additional step to detect possibilities of outliers appearing in the end result set from our convolutional model.

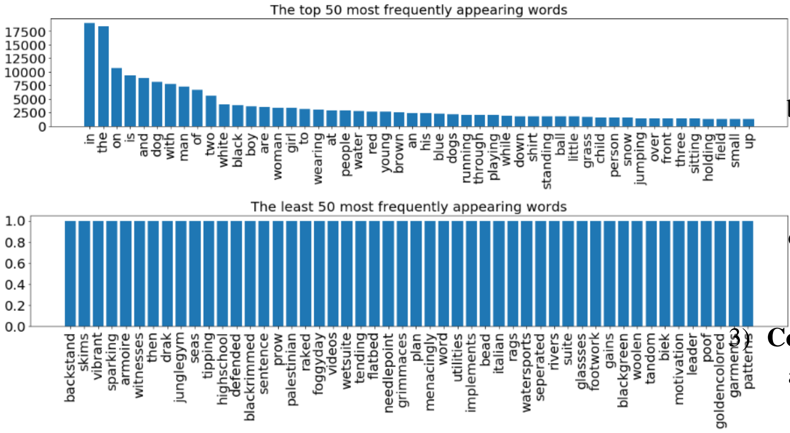

Fig. 1. The 50 Most Frequent and Least Appearing Words in A Sample Batch

We implemented this mechanism to ensure that both image and caption data are in suitable formats for the convolutional model and LSTM to efficiently learn and generate accurate natural language caption prompts based on image inputs.

C. Configurations on Encoder and Decoder Models

Essentially, the task involves two primary components, the encoder and the decoder. The encoder is responsible for extracting high - level features from the input images, while the decoder generates caption tokens in the form of natural language based on these features. This section details the configurations of the encoder and decoder models used in this task.

The encoder model is based on the VGG-16 convolutional model, pre - trained on the ImageNet dataset. This model can categorize images into 1000 different categories and has been trained on over a million JPEG individual units from that dataset. VGG-16 is a powerful model known for its depth and ability to capture intricate details in images.

On the other hand, An LSTM network serves as the decoder model in this research due to its capacity to maintain long - term relationships primarily in text data [11]. The LSTM is a particular kind of recurrent neural network that works very efficiently for sequence prediction tasks.

Our LSTM network model consists of several key components, the embedding layer, core layers and lastly dense(fully connected) layers.

1) **Embedding Layer**
   a) Word index sequences are transformed into dense vectors of a fixed size by the embedding layer. This aids in identifying the semantic connections between words.
   b) The embedding dimension is set to a value of 256. The vocabulary size depends on the tokenizer used to pre process the text data.
2) **Core Layers**
   a) The embedded word sequences are processed by the core layers, which keep track of the words that have come before in order to anticipate the words that will come after [12]. Because LSTMs have ways to regulate the information flow, they are useful for learning temporal dependencies.
   b) The core layers process the embedded vectors in a sequential manner, updating their states with each timestep. The context from earlier words is retained as it processes words one at a time.
   c) The LSTM network within this task has been modified to contain 256 units. This value establishes how many dimensions the output space has.
3) **Combining Image Features with LSTM Output**
   a) To give context for the image and the partially generated caption tokens, the encoder's output(image features) is merged with the LSTM's output after processed by the activation function in the dense layer.
4) **Dense Layer with Softmax Activation**
   a) For the next word prediction, the final dense layer generates a probability distribution across the whole vocabulary using a softmax activation function [13].
   b) The output size of this layer must match the vocabulary size.
5) **Compilation**
   a) By implementing Adam optimizer which is excellent at managing sparse gradients in big datasets, categorical cross - entropy loss is used to construct the entire model.

This decoder model design may provide caption tokens that are both coherent and contextually appropriate by utilizing the long - term dependency maintenance capabilities of LSTMs in conjunction with picture attributes. This approach makes it quite efficient for image - to - text tasks.

D. Model Training

By using a pre - trained VGG-16 architecture to exploit its learned features, we were able to drastically shorten training time and enhance feature quality without having to start from scratch by manually selecting appropriate hyperparameters and training a whole new convolutional model [14].

In addition, the second - to - last layer of the VGG-16 model, which is dense and has 4096 units, is used to output features, while the fully connected layers are eliminated. The reason for this is that we do not require the final classification layer, however only the features are accounted for and assigned to their respective image IDs within the dictionary. Afterwards, The 4096 - dimensional feature vector is extracted by running the pre processed image through the modified convolutional model [15]. The features of the image that the VGG-16 model has trained to recognize are represented by this vector.

Furthermore, we have tried to implement a callback method from Keras named $earlyStopping$. It is meant to halt the training process whenever a certain performance metric stops improving, hence preventing over fitting and cutting down on needless training time. This approach is particularly helpful for deep learning models, where training can be time consuming and computationally expensive. A parameter named $restoreBestWeights$ is set to a value of $True$, to ensure that the final model is the one that performs best on the validation split, providing an optimal balance between under fitting and over fitting.

High level image features are efficiently captured by the VGG-16 model and passed on into the LSTM network to produce caption tokens. By utilizing a pre - trained model, we can take use of the rich feature representations that have been acquired from a substantial dataset, which allows for enhanced performance even with a comparatively smaller dataset [16].

E. Performance Evaluation

The training and validation losses of the VGG-16 model can be utilized to assess the performance of our encoder architecture, particularly in the context of this task [17]. The degree to which the model's predictions agree with the actual data is measured by loss functions, and monitoring these losses throughout epochs provides insight into the model's generalization and learning processes.

With regard to the above, we present a visualization of a sample of one of our encoder models of its obtained training and validation(implementing test data) loss rates.

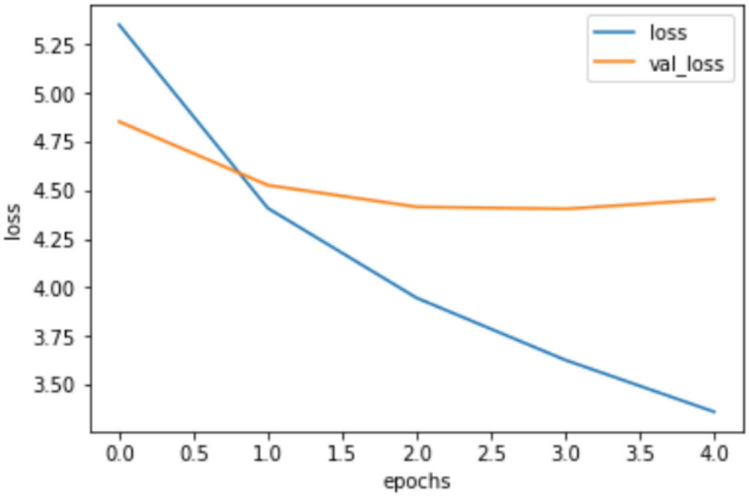

Fig. 2. Loss Rate from Training and Validation of Saved Model

As the training loss rate decreases steadily, indicating that the model is picking up the training data efficiently, the validation loss would also initially decrease, suggesting better performance on unobserved data [18]. Toward the end, it does, however, steady and slightly increase which indicates the onset of over fitting [19].

As for evaluating the quality and relevancy of generated caption tokens, we discuss a metric known as BLEU which is essential for converting visual data into comprehensible and contextually correct language. It specializes in comparison of the quality of text produced by machines to a reference set.

1) **Bilingual Evaluation Understudy Score**

a) BLEU(Bilingual Evaluation Understudy) is a quantitative natural language processing metric proposed by Papineni in *BLEU:A Method for Automatic Evaluation of Machine Translation, Proceedings of the* 40^{th} *Annual Meeting of the Association for Computational Linguistics(ACL), Philadelphia, July(2002):311-318* [20]. It measures the similarity between the generated text(predictions) and one or more target texts(ground truth). This works by figuring out how closely the generated text's n - grams — contiguous sequences of n elements from a given text sample match with the target texts' n - grams. A match is considered to be true if the produced caption token's n - gram does exist in some or all elements of the target text. In other words, BLEU determines precision in terms of any n - gram similarities found, with scores ranging from 0 to 1, where a higher score would indicate a better performance.

$$BLEU(\text{candidate}, \text{ref}) = \frac{\sum_{\forall n-\text{gram in candidate}} \text{Count clip}(n-\text{gram})}{\sum n-\text{gram in candidate}^{\text{Count}(n-\text{gram})}} \times BP$$

In the above, *Count(n - gram)* refers to the number of a given n - gram's occurences within the produced caption tokens, while

$$Count_{clip}(n\text{ - }gram)$$

instead refers to a mechanism of calculating *Count(n - gram)* given a certain n - gram value and then clipping the result of that calculation in respect to the maximum number of the n - gram's occurences.

In cases where multiple target texts are available, we pick the maximum BLEU value.

$$BLEU = \max\left(BLEU\left(\text{candidate}, \text{ref}_i\right)\right)$$

Additionally, in order to detect if the model is producing shorter phrases, a penalty term *brevity penalty*(BP) is added to penalize short phrase results and revise it simultaneously. This phenomenon may occur due to the MT system generating less caption tokens given longer target sentences while still generating relatively high BLEU score values. The value of brevity penalty is calculated by implementing the following.

$$BP = \left\{ \begin{array}{cc} 1 & \text{if } c > t \\ e^{(1-t/c)} & \text{if } c \leq t \end{array} \right\}$$

Where c would refer to the value of the length of the produced caption tokens, while t is for the target sentence's length.

Unfortunately, This also serves as one of many limitations of the BLEU metric system. BLEU often omits synonyms in sentences and has no recall feature with itself. It may perform poorly for certain languages other than english, however, in often situations we find it to correlate well with common human decisions and judgement.

b) Several forms of the BLEU score also exist, depending on fixed value of the n - gram [21]. The two that was implemented in the task are as follows.
   i) **BLEU-1**
      A) Calculates the proportion of uni gram matches between the reference and generated captions. A higher word-level accuracy in the generated captions is indicated by a higher BLEU-1 score.
   ii) **BLEU-2**
      A) Determines the bi gram match percentage to get a sense of how effectively the model generates word pairs that make sense in context. Improved contextual correctness is indicated by a higher BLEU-2 score.

In conclusion, BLEU calculates a modified n - gram precision and corrects short phrase values using the brevity penalty. This approach is sure to avoid the possibility of producing unassociated caption tokens having disproportionate BLEU score values.

IV. Results and Assessments

The following results, where BLEU-1 takes into account uni grams values and BLEU-2 takes into account bi grams values, are obtained by comparing the n - grams of the generated caption tokens to those of the target captions. The outcomes show how well the model was able to provide meaningful and cogent captions for the photographs in the dataset.

The saved decoder model instance we have selected has yielded the following BLEU scores.

1) **BLEU-1 :** 0.520913
2) **BLEU-2 :** 0.294527

A 52 percent majority of individual words in the generated caption tokens are properly predicted, according to an obtained BLEU-1 score of 0.5209. This implies the model has successfully picked up a sizable chunk of the vocabulary and is able to provide natural language prompt to input images with it.

Although it yields a lower value, the BLEU-2 score of 0.418 still shows a respectable degree of prediction accuracy for bi grams, or consecutive word pairs. This shows that the model can comprehend the basic structure and meaning of some sentences seen in the target sentences, but it might have a little bit of trouble with longer or more intricate sequences.

V. Summary and Future Updates

The model composition seems to perform generally well for producing pertinent and well - organized caption tokens for images in the dataset, as evidenced by the collected BLEU ratings. The greater BLEU-1 score demonstrates how well the model can identify and use certain words. Nonetheless, the reduced BLEU-2 score suggests that there is potential for enhancement in terms of identifying intricate phrase structures and guaranteeing the coherence of the produced caption tokens.

In order to address detected flaws in the task, future research could concentrate on addressing the constraints noted and enhancing the model's functionality. The model's capacity to generalize more images displaying more features can be improved by training it on bigger and more varied datasets, such as MSCOCO. Additionally, it would do no harm to try experimenting with more complex model designs, such Transformer - based models, which have demonstrated better results in problems requiring sequence - to - sequence navigation.

Last but not least, enhancing the pre - processing mechanisms and including more intricate attention methods could potentially help the model generate more coherent and pertinent caption tokens by allowing it to concentrate on particular and specific areas and features of the image.

VI. Supplementary Materials

Results and all source code are within the following. *https://github.com/edw9998/VGG16-demo*

Acknowledgment

Sincere thanks are extended to the supervisors who helped transform this research project a success and for their invaluable support.

Over and above that, I personally offer my deepest gratitude to BINUS International University for providing the necessary resources, facilities and a conducive environment for conducting this research. The access to computational resources and academic databases was crucial for the completion of this project.

Additionally, we acknowledge the creators and maintainers of the Flickr dataset and other publicly available datasets used in this research. Their efforts in curating and sharing such rich datasets have made significant contributions to the field of computer vision and natural language processing.

Finally, we would like to thank our families and friends for their unwavering support and understanding, which provided us with the strength and motivation to persevere through the demands of this research endeavor.

References

[1] Ren, S. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems(NIPS).

[2] He, K. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR).

[3] Donahue, J. (2015). Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR).

[4] Yang, Z. (2016). Hierarchical attention networks for document classification. In Proceedings of the NAACL-HLT.

[5] Zhang, H. (2019). Show, adapt and tell: Adversarial training of cross-domain image captioner. In Proceedings of the IEEE International Conference on Computer Vision(ICCV).

[6] Lin, T. Y. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR).

[7] Wu, Q. (2016). Image captioning and visual question answering based on attributes and external knowledge. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR).

[8] Yang, Z. (2016). Stacked Attention Networks for Image Question Answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR).

[9] Yao, T. (2017). Describing Objects in Detail with a Triple-Attention Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR).

[10] Framing Image Description as a Ranking Task : Data, Models and Evaluation Metrics, Journal of Artificial Intelligence Research, Volume 47, pages 853 - 899.

[11] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[12] Chollet, F. (2017). Deep Learning with Python. Manning Publications Co.

[13] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep Learning. MIT Press.

[14] K. Simonyan and A. Zisserman, ”Very Deep Convolutional Networks for Large-Scale Image Recognition” in Proc. of International Conference on Learning Representations(ICLR), 2015.

[15] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in neural information processing systems(pp. 1097-1105).

[16] Zhang, Z., Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems, 31.

[17] Eelbode, T., Sinonquel, P., Maes, F., Bisschops, R. (2021). Pitfalls in training and validation of deep learning systems. Best Practice and Research Clinical Gastroenterology, 52, 101712.

[18] Abbas, W., Tap, M. (2019, May). Adaptively weighted multi-task learning using inverse validation loss. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)(pp. 1408-1412). IEEE.

[19] Liu, H., Simonyan, K., Yang, Y. (2018). Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055.

[20] Papineni, K., Roukos, S., Ward, T., Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics(pp. 311-318).

[21] Yang, M., Zhu, J., Li, J., Wang, L., Qi, H., Li, S., Daxin, L. (2008, November). Extending BLEU evaluation method with linguistic weight. In 2008 The 9th International Conference for Young Computer Scientists(pp. 1683-1688). IEEE.