Jurnal Ilmiah
KURSOR
Menuju Solusi Teknologi Informasi

Vol. 12, No. 3, Juli 2024

ISSN 0216 – 0544
e-ISSN 2301– 6914

IMAGE CAPTIONING USING TRANSFORMER WITH IMAGE FEATURE EXTRACTION BY XCEPTION AND INCEPTION-V3

[a]Jasman Pardede, [b]Fandi Ahmad

[a,b] Department of Informatics, Institut Teknologi Nasional (Itenas) Bandung, Indonesia
E-mail: jasman@itenas.ac.id, fandi98ahmad@mhs.itenas.ac.id

Abstract

Image captioning is a task in image processing that involves creating text descriptions that can describe the image content. The formation of the image captioning system model is influenced by image interpretation related to the given image caption. Image interpretation is influenced by the feature extraction used. This research proposes feature extraction with Xception and Inception-V3 by generating an image captioning model using Transformer. Model performance is measured based on BLUE and METEOR values. Based on the results of research conducted on the Flickr8k Dataset, it shows that the best model performance is using Xception feature extraction and batch_size = 256. The image captioning performance of Xception feature extraction for BLUE-1, BLUE-2, BLUE-3, BLUE-4, and METEOR when compared with Inception-V3 achieves increasing of 13.15%, 18.03%, 18.71%, 27.27%, and 15.43% respectively. The performance for Xception feature extraction with batch_size = 256 compared with batch_size = 128, increasing BLUE-1, BLUE-2, BLUE-3, BLUE-4, and METEOR namely 19.81%, 41.84%, 52.23%, 53.14%, and 31.56% respectively.

Key words: batch_size, image captioning, Inception-V3, Xception, Transformer.

INTRODUCTION

Giving descriptive photos a story is the aim of picture captioning, a field that combines computer vision and natural language processing. It is a two-step procedure that depends on precise visual interpretation and accurate syntactic and semantic language understanding [1][2]. Image captioning has a wide range of uses, including human-computer interaction, virtual assistants, assistive technology for the blind, and several more natural language processing (NLP) uses [3], [4]. Prior solutions to this issue have only been able to solve the problem up until the point of providing appropriate image labels for an input image. That, though, is a superficial interpretation of the picture. comprehension the image holistically, rather than simply the different objects that make it up, is referred to as having a high-level comprehension of the image [5]. The challenge of comprehending a picture as a high-level input and automatically producing appropriate captions for the image is addressed by automatic image captioning [6]. Image captioning, like any area of artificial intelligence, is not without its obstacles and problems, but it also has great potential for greatly improving our ability to comprehend and describe visual material [7]. A significant obstacle lies in the system's capacity to provide precise and educational depictions, necessitating a profound comprehension of the visual environment, acceptable sentence construction, and the utilization of suitable terminology [8]. Evaluating caption quality becomes essential to determining how effectively an image captioning system achieves its goals. The BLEU (Bilingual Evaluation Understudy) score is one evaluation measure that is frequently employed. Based on how closely the system-generated descriptions resemble those written by humans, BLEU assigns a numerical score. The degree of similarity between machine-generated translations and human reference texts may be quantified by BLEU scores [9]. Convolutional neural networks are a common method for creating image captioning (CNN)[10] such as the Transformer and Xception models, followed by the Recurrent Neural Network (RNN) model. While RNNs are used to create word sequences that make up the final description, CNNs are used to extract visual information from photos [11]. By combining the two, the model is able to understand visual material similarly to humans, which includes the ability to identify and characterize objects as well as the connections between them. This strategy has been effectively used in a number of situations, opening the door for the creation of more complex and dependable picture captioning systems [12].

Previous research on image captioning by Pal [13] carried out using various models for image feature extraction, namely VGG16 and VGG19, Inception-V3, and Inception-ResNetV2. Meanwhile, the model used for caption generation is LSTM. The aim of this research is to compare image feature extraction models and find out which model has the best BLEU score. The results of the research concluded that Inception-V3 was the best model among the models used, because the BLEU score range obtained from Inception-V3 was the smallest. The lowest BLEU score of Inception-V3 is 0.295, and the highest BLEU score is 0.895. Next research by Jatmiko [12] employed four CNN model architecture methods: VGG-16, LeNet, Xception, and MobileNet. The Xception model yielded the best accuracy and lowest loss, with 0.93 accuracy and 0.19 loss, precision 0.93, recall 0.935, and f1-score 0.93. In contrast, LeNet achieved 0.897 accuracy and 0.28 loss, mobileNet achieved 0.908 accuracy and 0.225 loss, while the other model, VGG-16, yielded 0.90 accuracy and 0.27 loss. Next research by Rizki [8] CNN accuracy was 0.94, VGG16 accuracy was 0.88, and ResNet50 accuracy was 0.72 when classifying breast cancer histopathology pictures to identify breast tumors utilizing Transfer Learning techniques. Next research by Sharma [14] With an accuracy of 0.9625, 0.9625, 0.9574, and 0.9411 for 40X, 100X, 200X, and 400X level of magnification, respectively, the pre-trained Xception model and SVM classifier with the "radial basis function" kernel have produced the best and most consistent results when used for magnification-dependent breast cancer histopathological image classification.

This study differs from the others in that it attempts to use Transformer to apply the Xception approach. Convolutional neural networks, such as the Xception architecture, learn filtering in three dimensions: two spatial dimensions (width and height) and one dimensional channel. Xception is a residually connected linear depthwise separable convolution stack [15]. Building and changing this architecture is therefore simple. The three channels that make up the Xception model's implementation are the entry flow, middle flow, and exit flow. After entering the entrance flow and passing through the middle flow, which is repeated eight times, the data will eventually depart the entry flow. [12]. Furthermore, a comparison between the Transformer and Inception-V3 approaches will be conducted in this study. A deep convolutional architecture known as Inception-V3 was created as a consequence of study into the Inception-v1 or GoogleNet model [6]. Two name changes and developments in this architecture have been made to this method in order to reduce the number of connections or parameters without reducing the size of the network used: batch normalization (BN) and additional factorization at the convolution stage, which has been named Inception-V3 [6]. Moreover, the Transformer approach is applied in text feature extraction. A model called Transformer is capable of word sequence prediction. Recurrent Neural Network (RNN)-like encoder-decoder architecture is used by this transformer [16]. The primary distinction is that, although the RNN can only accept input up to one word at a time, the Transformer may receive phrase or sequence input in parallel, meaning there is no temporal lag connected with the input and all words in the sentence can be skipped concurrently [17]. Then for the evaluation of caption results using BLEU and METEOR, Bilingual Evaluation Understudy (BLEU) is a translation evaluation metric that measures the similarity between translation and reference translation. The n-gram matching notion, which determines the degree of similarity between a translation and its reference translation at the n-gram level, is the foundation of BLEU [7]. Then, the translation assessment metric known as METEOR, or Metric for assessment of Translation with Explicit Ordering, quantifies the degree of similarity between the translated and reference translations. Unigram matching, the foundation of METEOR, determines the degree of similarity between a translation and its reference translation at the unigram level [18]. Higher numbers indicate more similarity between the translation and the reference translation. BLEU and METEOR scores range from 0 to 1. The Flickr8k dataset, which was utilized in this study, was downloaded from the Kaggle website. There are 40,455 captions and 8,091 picture data in the Flickr8k dataset. The photographs in this collection are hand-picked to represent a variety of scenarios and events, and most don't include well-known persons or places. Every image in the dataset includes five produced reference captions, each of which is a phrase that explains the items and events of the image data.

MATERIAL AND METHODS

Xception

Xception: Extreme version of The Inception model's development led to the initial introduction of Inception [15]. Convolutional neural networks are used in Inception, an architectural model that examines filtering in three dimensions: width, height, and channel dimensions. Correlation and cross-channel mapping are handled by a convolutional kernel. divides the process factorially into a series of activities that each individually monitor cross-channel correlations and spatial correlations in an effort to streamline and improve efficiency. Xception enhances the Inception model by mapping cross-channel correlations using 1x1 convolution and isolating the mapping spatial correlations from each output channel [15], as showed on Figure 1.

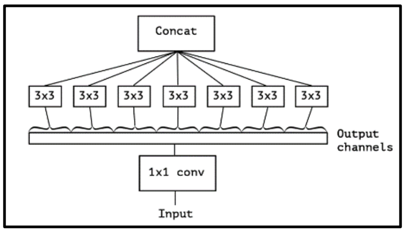


Fig 1. Block diagram of Inception's "extreme"

Eventually, this design was refined even further, with depthwise separable convolution—also known as separable convolution—replacing the Inception module. The foundation for feature extraction from the network is Xception's 36 convolution layers. With the exception of the beginning and end of each module, each of the 14 modules that these 36 layers are arranged into has a linear residual link surrounding it. To put it simply, Xception is a residually connected stacked linear depthwise separable convolution. Thus, it is simple to form and alter this architecture. The three channels that make up the Xception architecture are the entry flow, middle flow, and exit flow [14], as showed on Figure 2. After entering the input flow and passing through the middle flow, which is repeated eight times, the data will finally enter the exit flow.

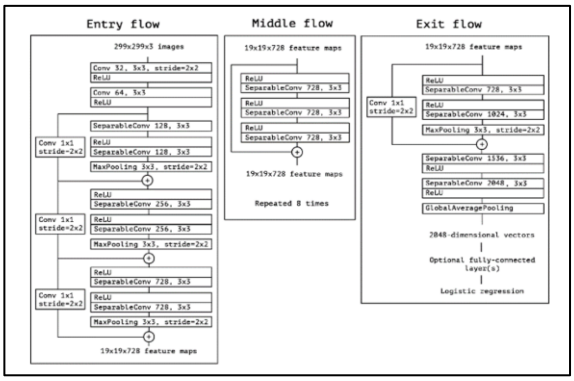


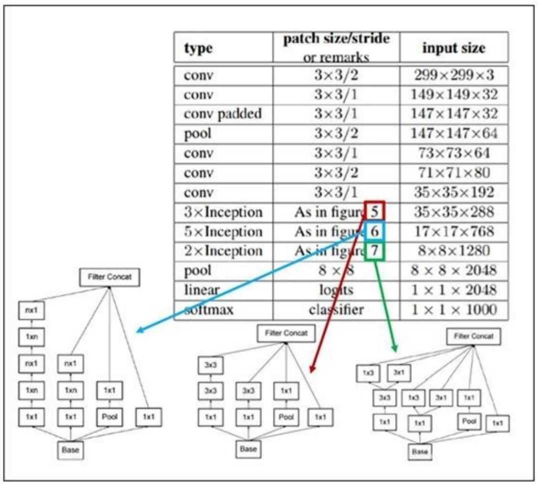
Fig 2. Xception architecture

The design consists of a block of convolution layers, MaxPooling layer, Separable Convolution layer, and ReLu (Rectified Linear Unit) activation. When the two tensors are combined, use 'ADD' to see the input sizes of each tensor. For instance, the image size is 299x299x3, and then it enters the middle flow, yielding a size of I9xl9x728 for Global Average Pooling, depending on the number of repetitions and whether to add a fully connected layer [15].

Inception-V3

The Google team created the convolutional neural network (CNN) architecture known as Inception-V3. This architecture is an evolution of the Inception-v1 model, which was improved upon in several ways to address issues with better feature extraction from pictures with different levels of complexity. The Inception-V3 model is widely recognized for its inventive method of merging distinct convolution filter sizes into a solitary layer, hence facilitating the effective detection of information at several spatial scales [19]. Furthermore, Inception-V3 employs a more profound and intricate architecture to execute the Inception module, enabling this model to comprehend more intricate and profound feature hierarchies in pictures [8].

The usage of an *atrous* convolution module, which enables the network to have a broader field of vision without compromising spatial resolution, is one of Inception-v3's unique characteristics. This enhances the model's capacity to identify big items and more extensive context in pictures. Furthermore, Inception-V3 uses sophisticated regularization techniques including dropout and batch normalization, which lessen overfitting and enhance model generalization to never-before-seen data. Inception-V3's design is divided into three sections, as shown on Figure 3.



type	patch size/stride or remarks	input size
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	As in figure 5	35×35×288
5×Inception	As in figure 6	17×17×768
2×Inception	As in figure 7	8×8×1280
pool	8 × 8	8 × 8 × 2048
linear	logits	1 × 1 × 2048
softmax	classifier	1 × 1 × 1000

Fig 3. Inception-V3 architecture configuration

First, the fundamental convolutional layer (stem), which is made up of conv0 through conv4 and comprises 5 layers with permissible padding types. ReLu (BN) activation comes after every convolution process. Second, the inception module's layer section, which has 11 layers (mixed0 through mixed10) with the same kind of padding in each convolution operation. Convolution factorization with dimensions of 1x1, 3x3, 5x5, 1x7, and 7x7 is used in the construction of every layer in the module block [19]. Third, ImageNet datasets were used to train the classification model section.

Transformer

Transformer employs a *self-attention* mechanism in constructing its architecture model, which was previously described by Ashish Vaswani in his research paper "Attention is All You Need" [20]. An encoder-decoder helps a transformer model anticipate or recover words consecutively and convert one moving sequence to another.

Self-attention, point-wise, add & norm, linear, *softmax*, multi-head attention, and feed-forward are all used in the Transformer design. After that, fully connected layers are used by the encoder and decoder [20], as shown in Figure 4.

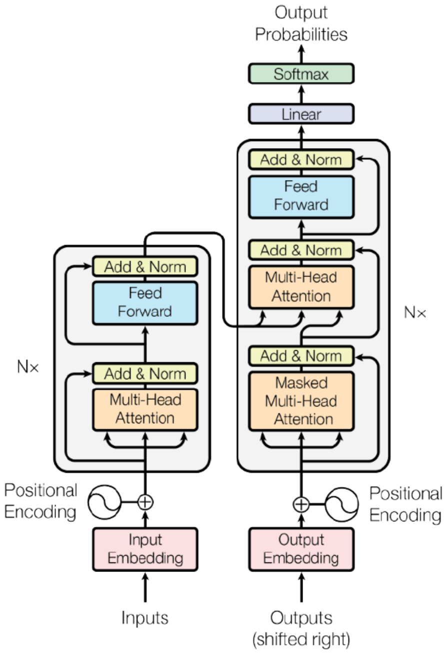


Fig 4. Transformer architecture

BLEU (Bilingual Evaluation Understudy)

The precision of the MT output in *n*-grams relative to the reference is measured by BLEU (Bilingual Evaluation Understudy), which is weighted by a brevity penalty to penalize translations that are too brief. High variation exists across various hyperparameters and pre-processing techniques for BLEU [9]. This is one of the methods used by bleu to assess the quality of generated text; to do this, each text is compared to a set of reference texts that were written by humans. There is no need to focus on syntactical accuracy in order to assess how closely machine-generated text resembles ground truth and how each one is scored. In order to assess the overall quality of the created text, an average score is finally calculated. The size of the produced text and the quantity of reference text affect the BLEU metric's performance. There are certain restrictions. For example, a high BLEU score does not always guarantee that the value of the created text will be good. BLEU scores are only beneficial if the generated text is exact [21]. The BLEU score is calculated using Equation (1).

$$BLEU = BP * \exp\left(\sum_{n=1}^{N} w_n * \log(p_n)\right) \quad (1)$$

where:

BP: brevity penalty reduction factor (short reduction).

N : maximum n-gram level used (usually up to 4-grams).

w_n : weights for each n-gram (usually equivalent for all n-grams $w_n = 1/N$).

p_n : n-gram precision, which measures how much the machine translated n-gram matches the reference n-gram.

BLEU-1 to BLEU-4 measurements is necessary in image captioning evaluation for measuring word-level similarity. BLEU-1 measures the unigram (single word) similarity between the model-generated sentence and the reference sentences. BLEU-2 measures bigram (two consecutive words) similarity, BLEU-3 measures trigram (three consecutive words) similarity, and BLEU-4 measures four-gram similarity. The higher the BLEU score, the more similar the model-generated sentence is to the reference sentences [9].

METEOR (Metric for Evaluation of Translation with Explicit Ordering)

Another measure that aids in the computation of machine produced language is METEOR. A generalized unigram match is made between text produced by machines and references written by humans. Also, the most excellent score is chosen from each reference's separately evaluated ones based on similarity, a score assessed in the event of several references [21]. METEOR, on the other hand, uses exact match, stem match, or synonymy match to evaluate translation while taking semantic information into account. But under the scenario of our previous example, the METEOR weighting system would not permit a significant penalty of the missing negation marker [18].

The METEOR metric for machine translation evaluation is based on the harmonic mean of unigram precision and recall, with recall weighted higher than precision. It also has several features that are not found in other metrics, such as stemming and synonymy matching, along with the standard exact word matching. The METEOR score is typically reported on a scale of 0 to 1, with higher scores indicating better translations [18].

The METEOR score is calculated using Equation (2)[22]. Meanwhile, *P_en*, *F-mean*, The unigram *Precision,* and *Recall* are computed using Equation (3), Equation (4), Equation (5), and Equation (6), respectively.

$$METEOR = (1 - P\_en) * F\_mean \quad (2)$$

$$P\_en = \text{gamma} * (1 - \text{FCM}) \quad (3)$$

$$F\_mean = \frac{(2 * \text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (4)$$

$$Precision = C / U\_c \quad (5)$$

$$Recall = C / U\_r \quad (6)$$

where:

P_{en} : the chunk penalty, which measures how fragmented the translation is compared to the reference

F_mean : measures how well the translation matches the reference in terms of individual words.

$gamma$: a parameter that controls the severity of the penalty. It is typically set to 0.5

FCM : the proportion of chunks in the reference that are matched by the translation

C : the number of unigram matches between the translation and the reference.

U_c : the number of unigrams in the translation

U_r : the number of unigrams in the reference.

Dataset

This study uses the Flickr8k dataset, which was downloaded from the Kaggle website. There are 40,455 captions and 8,091 picture data in the Flickr8k dataset. The photographs in this collection are hand-picked to represent a range of scenarios and events, and they often don't feature famous persons or places. Every image in the dataset contains five personally developed and chosen reference captions, each of which is a phrase that may be used to explain various items and events from the image data.

Block Diagram

The general research system workflow will be explained which is described using block diagrams in the Xception and Inception-V3 methods for image feature extraction, then the Transformer method for text feature extraction will be stated in Figure 5.

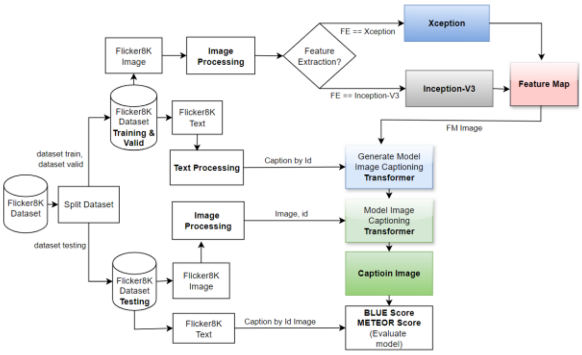


Fig 5. Block diagram of Xception and Inception-V3 with Transformers

Based on Figure 5, there are 6 parts of the research stages. In the stages section there is an explanation of the stages as follows:

1. Initialisation
   Prepare the Flickr8k dataset, which is made up of two different sets of data: the image-only Flicker8k_Dataset and the caption-only Flicker8k_Text.
2. Text Processing
   a) Tokenizing is the process of separating each word from the sentence. The tokenizing process is carried out, namely taking words from existing sentences, then the word will be entered into the vocabulary.
   b) Mark caption is the process of adding new tokens. The addition of the token "<start>" which is placed at the beginning of the sentence to mark the beginning of the caption sentence, and the addition of the token "<end>" which is placed at the end of the sentence to mark the end of the caption sentence.
3. Image Preprocessing: Resize image to 299x299x3.
4. Feature Extraction: Feature extraction in this research uses Inception-V3 or Xception.
   a. Inception-V3
      In Inception-V3 feature extraction only need to extract map features, so the linear and *softmax* layers from the Inception-V3 model in Figure 3 are removed. Before being given to the model, all images must be preprocessed in stage 3. The output form of this model is 8x8x2048.
   b. Xception
      In Xception feature extraction the Optional fully-connected and Logistic regression layers of the model are removed. Before extracting features by Xception, pre-processing is carried out first. The output of this model is 8x8x2048.
5. Generating Model by Transformers
   Transformer is a model used for the caption generation stage or for generating captions. The model is created from layers of self-attention layers, a stacked point-wise, feed-forward network, fully connected for the encoder and decoder. Feature extraction results by Xception or Inception are used as input and text processing results are used to target the Transformer as stated in Figure 6.

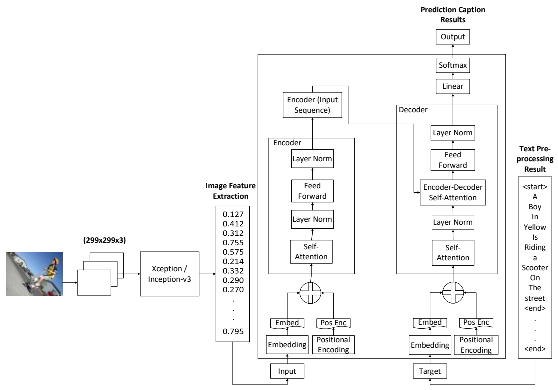


Fig 6. Image captioning system design

6. Inference
   a. Model fitting is the training process of a model that has been created and carried

out 150 epochs or 150 training times on the model.

The model that has been trained is then evaluated using the BLEU and the METEOR score to find out how well the model created is for the image captioning that be generated.

RESULT AND DISCUSSION

In this study, to show the effect of image feature extraction on the performance of the image captioning transformer model were used, batch_size = 128 or batch_size = 256 and training = 150 epochs for each proposed feature extraction. The performance assessment of the model used is based on the loss and accuracy values. The performance model Xception and Transformer using batch_size = 128 is stated in Figure 7. Whereas, the performance model Inception-V3 and Transformer using batch_size = 128 is stated in Figure 8.

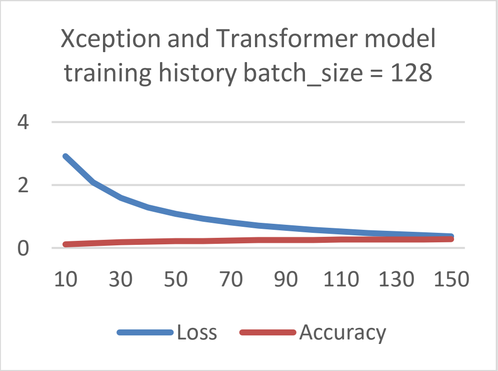


Figure 7. Xception and Transformer model training graph batch_size = 128

The research results reveal that Xception and Inception-V3 feature extraction at batch_size = 128 have almost the same accuracy performance result for each epoch. At the 10th epoch, feature extraction has accuracy for Xception and Inception-V3 is 0.1143 and 0.1107 respectively. Meanwhile, the loss values for Xception and Inception-V3 are 2.9143 and 2.9882, respectively.

The formation of the image captioning model was carried out until the 150th epoch. The accuracy performance at the 150th epoch for Xception feature extraction reached 0.2789 with a loss value of 0.3677. Meanwhile, for Inception-V3 feature extraction, the accuracy and loss values are 0.2613 and 0.5398 respectively.

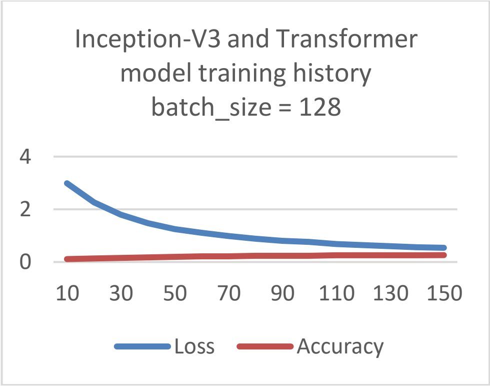


Fig 8. Inception-V3 and Transformer model training graph batch_size = 128

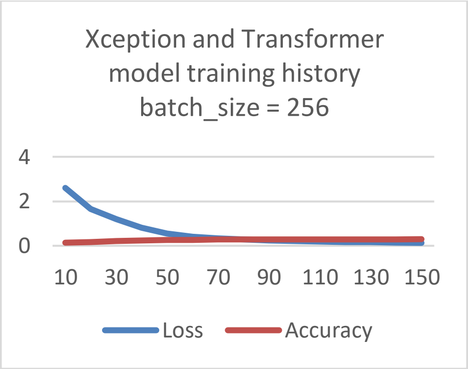


Fig 9. Xception and Transformer model training graph batch_size = 256

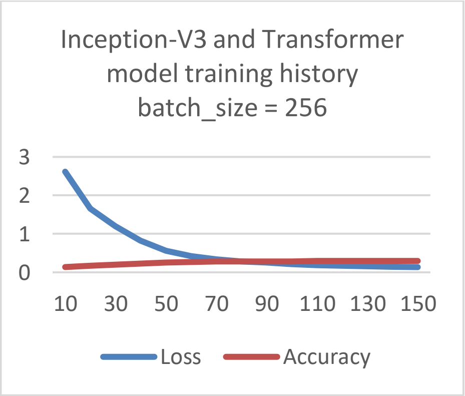


Fig 10. Inception-V3 and Transformer model training graph batch_size = 256

Furthermore, the performance model Xception and Transformer using batch_size = 256 is stated in Figure 9. Whereas, the performance model Inception-V3 and Transformer using batch_size = 256 is stated in Figure 10. At the 10th epoch, feature extraction has accuracy for Xception and Inception-V3 is 0.1381 and 0.1373 respectively. Meanwhile, the loss values for Xception and Inception-V3 are 2.6005 and 2.6171, respectively.

Meanwhile, the accuracy and loss value on Xception at the 150th epoch is 0.2971 and 0.1301 respectively. Meanwhile, the Inception-V3 feature extraction is 0.2964 and 0.1353 respectively.

Based on the experimental results, it was disclosed that feature extraction influenced the performance image captioning system. The Xception feature extraction was better than Inception-V3. At batch_size = 128 Xception feature extraction increases accuracy by 6.3105% compared to Inception-V3. Model performance is also affected by the given batch_size. The best model performance is at batch_size = 256. At batch_size = 256, Xception feature extraction increased accuracy by 6.1259% compared with batch_size = 128.

Table 1. The Xception and Transformer Methods Batch_size = 128.

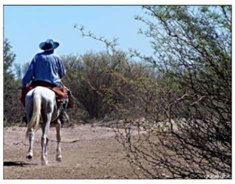
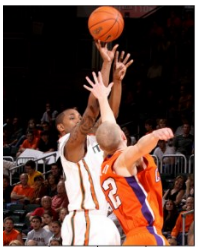
No	Image	BLEU & METEOR Score	Real Caption	Predicted Caption
1	 405331006_4e94e07698.jpg	BLEU 1: 0.747082 BLEU 2: 0.547860 BLEU 3: 0.202911 BLEU 4: 0.114138 METEOR: 0.6915	1. A man in a blue cowboy hat is riding a white horse 2. A man in blue is riding a horse on a dirt road 3. A man wearing a blue hat and shirt is riding a white horse 4. A person in a blue cowboy hat rides a horse down a dirt trail 5. The person in the blue shirt and blue hat is riding a white horse	a man in a blue shirt is sitting on a dock
2	 3126752627_dc2d6674da.jpg	BLEU 1: 1.000000 BLEU 2: 0.833333 BLEU 3: 0.800000 BLEU 4: 0.750000 METEOR: 0.9985	1. A basketball player shooting while another player is trying to block his shot 2. A basketball player tries to block another 3. Two basketball players reaching for a ball 4. Two men are playing Basketball 5. two men jump for the basketball	two basketball players reaching for a ball

Table 2. The Inception-V3 and Transformer Methods Batch_size = 128.

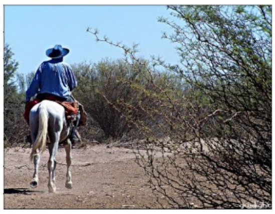
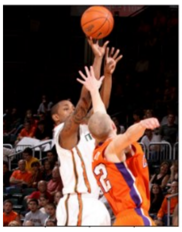
No	Image	BLEU & METEOR Score	Real Caption	Predicted Caption
1	 405331006_4e94e07698.jpg	BLEU 1: 0.818731 BLEU 2: 0.636791 BLEU 3: 0.614048 BLEU 4: 0.584808 METEOR: 0.8441	1. A man in a blue cowboy hat is riding a white horse 2. A man in blue is riding a horse on a dirt road 3. A man wearing a blue hat and shirt is riding a white horse 4. A person in a blue cowboy hat rides a horse down a dirt trail 5. The person in the blue shirt and blue hat is riding a white horse	a man is riding a horse on a dirt road
2	 3126752627_dc2d6674da.jpg	BLEU 1: 0.153341 BLEU 2: 0.000000 BLEU 3: 0.000000 BLEU 4: 0.000000 METEOR: 0.1333	1. A basketball player shooting while another player is trying to block his shot 2. A basketball player tries to block another 3. Two basketball players reaching for a ball 4. Two men are playing Basketball 5. two men jump for the basketball	a man in a white uniform holds a basketball during a game

Table 3. The Xception and Transformer Methods Batch_size = 256.

No	Image	BLEU & METEOR Score	Real Caption	Predicted Caption
1	 405331006_4e94e07698.jpg	BLEU 1: 0.928571 BLEU 2: 0.769231 BLEU 3: 0.583333 BLEU 4: 0.363636 METEOR: 0.8800	1. A man in a blue cowboy hat is riding a white horse 2. A man in blue is riding a horse on a dirt road 3. A man wearing a blue hat and shirt is riding a white horse 4. A person in a blue cowboy hat rides a horse down a dirt trail 5. The person in the blue shirt and blue hat is riding a white horse	a man in a blue hat and blue jacket is riding a white horse
2	 3126752627_dc2d6674da.jpg	BLEU 1: 1.000000 BLEU 2: 0.833333 BLEU 3: 0.800000 BLEU 4: 0.750000 METEOR : 0.9985	1. A basketball player shooting while another player is trying to block his shot 2. A basketball player tries to block another 3. Two basketball players reaching for a ball 4. Two men are playing Basketball 5. two men jump for the basketball	two basketball players reaching for a ball

Table 4. The Inception-V3 and Transformer Methods Batch_size = 256.

No	Image	BLEU & METEOR Score	Real Caption	Predicted Caption
1	 405331006_4e94e07698.jpg	BLEU 1: 0.913101 BLEU 2: 0.821791 BLEU 3: 0.811645 BLEU 4: 0.570688 METEOR : 0.9216	1. A man in a blue cowboy hat is riding a white horse 2. A man in blue is riding a horse on a dirt road 3. A man wearing a blue hat and shirt is riding a white horse 4. A person in a blue cowboy hat rides a horse down a dirt trail 5. The person in the blue shirt and blue hat is riding a white horse	a man in a blue hat is riding a white horse
2	 3126752627_dc2d6674da.jpg	BLEU 1: 1.000000 BLEU 2: 0.833333 BLEU 3: 0.800000 BLEU 4: 0.750000 METEOR : 0.9985	1. A basketball player shooting while another player is trying to block his shot 2. A basketball player tries to block another 3. Two basketball players reaching for a ball 4. Two men are playing Basketball 5. two men jump for the basketball	two basketball players reaching for a ball

To test the image captioning model that has been generated, 10% test data or 810 images are used. An example of the results of image caption generation for Xception feature extraction and Transformer with batch_size = 128 is as stated in Table 1. Image caption generation for Inception-V3 extraction is stated in Table 2. Meanwhile, image caption generation for batch_size = 256 for feature extraction Xception and Inception-V3 are stated in Table 3 and Table 4 respectively.

System performance in this study was measured based on the BLUE and METEOR values. System performance is considered based on feature extraction and proposed batch_size. The result of generating the image captioning model with Xception feature extraction and batch_size = 128 on image

405331006_4e94e07698.jpg is '*a man in a blue shirt is sitting on a dock*', as stated in Table 1. Meanwhile the real caption is as stated in the Real Caption column. Based on the values of BLUE-1, BLUE-2, BLUE-3, BLUE-4, and METEOR are 0.747082, 0.547860, 0.202911, 0.114138, and 0.6915, respectively. The result of generating the image captioning model with Xception feature extraction and batch_size = 256 on image 405331006_4e94e07698.jpg is '*a man in a blue hat and blue jacket is riding a white horse*', as stated in Table 3. So that, BLUE-1, BLUE-2, BLUE-3, BLUE-4, and METEOR are 0.928571, 0.769231, 0.583333,0.363636, and 0.8800, respectively.

Image captions generated by the system can produce different or same caption. In the testing image 3126752627_dc2d6674da.jpg by batch_size = 256 with Exception and Inception feature extraction produced the same image caption, as stated in Table 3 and Table 4. However, image 405331006_4e94e07698.jpg produces different captions.

Table 5. System Performance Comparison

	Xception and Transformer		**Inception-v3 and Transformer**	
	batch_size = 128	**batch_size = 256**	**batch_size = 128**	**batch_size = 256**
BLUE-1	0.7771	0.9691	0.5606	0.8416
BLUE-2	0.5151	0.8857	0.2948	0.7260
BLUE-3	0.4006	0.8385	0.2107	0.6816
BLUE-4	0.3595	0.7672	0.1468	0.5580
METEOR	0.6677	0.9756	0.4780	0.8251

Based on the experiments carried out, the comparison of the performance image captioning system with batch_size and the extraction features used is shown in Table 5. Based on experimental results, it proves that the best feature extraction is Xception with batch_size = 256. The image captioning performance of Xception for BLUE-1, BLUE-2, BLUE-3, BLUE-4, and METEOR when compared with Inception-V3 achieves an increase of 13.15%, 18.03%, 18.71%, 27.27%, and 15.43% respectively. Meanwhile, if we pay attention to the effect of batch_size = 256 against batch_size = 128 with Xception feature extraction, we get an increase in the performance of BLUE-1, BLUE-2, BLUE-3, BLUE-4, and METEOR respectively, namely 19.81%, 41.84%, 52.23%, 53.14 %, and 31.56%.

CONCLUSION

This research has shown the effect of image feature extraction and batch_size on the performance of the image captioning system. Image captioning performance is measured based on BLUE and METEOR values. Based on the research results, it reveal that Xception feature extraction produces the best performance when compared to Inception-V3. The best batch_size for Xception and Inception-V3 is 256. Based on the increase in the BLUE value for Xception feature extraction with Inception-V3, it was found that the highest increase was in BLUE-4, namely 18.71%. This means that Xception and batch_size = 256 are better 18.71% at arranging 4-words in sequence compared to Inception-V3.

REFERENCES

[1] A. Elhagry and K. Kadaoui, “A Thorough Review on Recent Deep Learning Methodologies for Image Captioning,” Jul. 2021, [Online]. Available: http://arxiv.org/abs/2107.13114

[2] U. L. Yuhana, I. Imamah, C. Fatichah, and B. J. Santoso, “Effectiveness Of Deep Learning Approach For Text Classification In Adaptive Learning,” *Jurnal Ilmiah Kursor*, vol. 11, no. 3, p. 137, Jul. 2022, doi: 10.21107/kursor.v11i3.285.

[3] Q. Wang, J. Wan, and A. B. Chan, “On Diversity in Image Captioning: Metrics and Methods,” *IEEE Trans Pattern Anal Mach Intell*, vol. 44, no. 2, pp. 1035–1049, Feb. 2022, doi: 10.1109/TPAMI.2020.3013834..

[4] K. R. Chowdhary, “Natural Language Processing,” in *Fundamentals of Artificial Intelligence*, New Delhi: Springer India, 2020, pp. 603–649. doi: 10.1007/978-81-322-3972-7_19.

[5] A. Mathew, P. Amudha, and S. Sivakumari, “Deep Learning Techniques: An Overview,” 2021, pp. 599–608. doi: 10.1007/978-981-15-3383-9_54.

[6] A. E. Minarno, L. Aripa, Y. Azhar, and Y. Munarko, “Classification of Malaria Cell Image using Inception-V3 Architecture,” JOIV : *International Journal on Informatics Visualization*, vol. 7, no. 2, pp. 273–278, May 2023, doi: 10.30630/joiv.7.2.1301.

[7] I. Fahruzi, “Sleep Disorder Identification From Single Lead ECG By Improving Hyperparameters Of 1D-CNN,” *Jurnal Ilmiah Kursor*, vol. 11, no. 4, pp. 157–164, Jan. 2023, doi: 10.21107/kursor.v11i4.302.

[8] N. Jethwa, H. Gabajiwala, A. Mishra, P. Joshi, and P. Natu, “Comparative Analysis between InceptionResnetV2 and InceptionV3 for Attention based Image Captioning,” in 2021 2nd *Global Conference for Advancement in Technology (GCAT), IEEE*, Oct. 2021, pp. 1–6. doi: 10.1109/GCAT52182.2021.9587514.

[9] N. Mathur, T. Baldwin, and T. Cohn, “Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics,” Jun. 2020, [Online]. Available: http://arxiv.org/abs/2006.06264

[10] Fawaidul Badri, M. Taqijuddin Alawiy, and Eko Mulyanto Yuniarno, “Deep Learning Architecture Based On Convolutional Neural Network (CNN) In Image Classification,” *Jurnal Ilmiah Kursor*, vol. 12, no. 2, pp. 83–92, Dec. 2023, doi: 10.21107/kursor.v12i2.349.

[11] D. Rizki Chandranegara, F. Haidar Pratama, S. Fajrianur, M. Rizky Eka Putra, and Z. Sari, “Automated Detection of Breast Cancer Histopathology Image Using Convolutional Neural Network and Transfer Learning,” vol. 22, no. 3, pp. 455–468, 2023, doi: 10.30812/matrik.v22i3.xxx.

[12] R. H. Jatmiko and Y. Pristyanto, “Investigating The Effectiveness of Various Convolutional Neural Network Model Architectures for Skin Cancer Melanoma Classification,” *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 23, no. 1, pp. 1–16, Oct. 2023, doi: 10.30812/matrik.v23i1.3185.

[13] A. Pal, S. Kar, A. Taneja, and V. Kumar Jadoun, “Image Captioning and Comparison of Different Encoders,” *J Phys Conf Ser*, vol. 1478, no. 1, p. 012004, Apr. 2020, doi: 10.1088/1742-6596/1478/1/012004.

[14] S. Sharma and S. Kumar, “The Xception model: A potential feature extractor in breast cancer histology images classification,” *ICT Express*, vol. 8, no. 1, pp. 101–108, Mar. 2022, doi: 10.1016/j.icte.2021.11.010.

[15] X. Wu, R. Liu, H. Yang, and Z. Chen, “An Xception Based Convolutional Neural Network for Scene Image Classification with Transfer Learning,” in 2020 2nd *International Conference on Information Technology and Computer Application (ITCA), IEEE*, Dec. 2020, pp. 262–267. doi: 10.1109/ITCA52113.2020.00063.

[16] R. F. Hadi, S. Sa’adah, and D. Adytia, “Forecasting of GPU Prices Using Transformer Method,” *Jurnal Sisfokom* (*Sistem Informasi dan Komputer*), vol. 12, no. 1, pp. 136–144, Mar. 2023, doi: 10.32736/sisfokom.v12i1.1569.

[17] Z. Wang, Y. Ma, Z. Liu, and J. Tang, “R-Transformer: Recurrent Neural Network Enhanced Transformer,” Jul. 2019, [Online]. Available: http://arxiv.org/abs/1907.05572

[18] H. Saadany and C. Orasan, “BLEU, METEOR, BERTScore: Evaluation of Metrics Performance in Assessing Critical Translation Errors in Sentiment-oriented Text,” Sep. 2021, doi: 10.26615/978-954-452-071-7_006.

[19] N. Dong, L. Zhao, C. H. Wu, and J. F. Chang, “Inception v3 based cervical cell classification combined with artificially extracted features,” *Appl Soft Comput*, vol. 93, Aug. 2020, doi: 10.1016/j.asoc.2020.106311.

[20] A. Vaswani, et al.,"Attention is all you need, " in *31st Conference on Neural Information Processing Systems* (NIPS 2017), Long Beach, CA, USA.

[21] H. Sharma, M. Agrahari, S. K. Singh, M. Firoj, and R. K. Mishra, “Image Captioning: A Comprehensive Survey,” in *2020 International Conference on Power Electronics and IoT Applications in Renewable Energy and its Control, PARC 2020*, Institute of Electrical and Electronics Engineers Inc., Feb. 2020, pp. 325–328. doi: 10.1109/PARC49193.2020.236619.

[22] A. Lavie, A. Agarwal, Meteor: "An automatic metric for mt evaluation with improved correlation with human judgments," in *The Second Workshop on Statistical Machine Translation*, 2007, pp. 228–231

.