mathematics

MDPI

Article

Exploring Spatial-Based Position Encoding for Image Captioning

Xiaobao Yang [1,2], Shuai He [2], Junsheng Wu [3,*], Yang Yang [2], Zhiqiang Hou [2] and Sugang Ma [2]

[1] School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China; y78h11b09@xupt.edu.cn
[2] School of Computer Science & Technology, Xi'an University of Posts and Telecommunications, Xi'an 710061, China; hs670136496@stu.xupt.edu.cn (S.H.); yyang@stu.xupt.edu.cn (Y.Y.); hzq@xupt.edu.cn (Z.H.); msg@xupt.edu.cn (S.M.)
[3] School of Software, Northwestern Polytechnical University, Xi'an 710072, China
* Correspondence: wujunsheng@nwpu.edu.cn

Abstract: Image captioning has become a hot topic in artificial intelligence research and sits at the intersection of computer vision and natural language processing. Most recent imaging captioning models have adopted an "encoder + decoder" architecture, in which the encoder is employed generally to extract the visual feature, while the decoder generates the descriptive sentence word by word. However, the visual features need to be flattened into sequence form before being forwarded to the decoder, and this results in the loss of the 2D spatial position information of the image. This limitation is particularly pronounced in the Transformer architecture since it is inherently not position-aware. Therefore, in this paper, we propose a simple coordinate-based spatial position encoding method (CSPE) to remedy this deficiency. CSPE firstly creates the 2D position coordinates for each feature pixel, and then encodes them by row and by column separately via trainable or hard encoding, effectively strengthening the position representation of visual features and enriching the generated description sentences. In addition, in order to reduce the time cost, we also explore a diagonal-based spatial position encoding (DSPE) approach. Compared with CSPE, DSPE is slightly inferior in performance but has a faster calculation speed. Extensive experiments on the MS COCO 2014 dataset demonstrate that CSPE and DSPE can significantly enhance the spatial position representation of visual features. CSPE, in particular, demonstrates BLEU-4 and CIDEr metrics improved by 1.6% and 5.7%, respectively, compared with a baseline model without sequence-based position encoding, and also outperforms current sequence-based position encoding approaches by a significant margin. In addition, the robustness and plug-and-play ability of the proposed method are validated based on a medical captioning generation model.

Keywords: position encoding; image captioning; transformer

MSC: 68T45

check for updates

Citation: Yang, X.; He, S.; Wu, J.; Yang, Y.; Hou, Z.; Ma, S. Exploring Spatial-Based Position Encoding for Image Captioning. *Mathematics* **2023**, *11*, 4550. https://doi.org/10.3390/math11214550

Academic Editor: Konstantin Kozlov

Received: 30 September 2023
Revised: 30 October 2023
Accepted: 2 November 2023
Published: 4 November 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Image captioning is a hot topic in artificial intelligence research and spans the fields of Computer Vision (CV) and Natural Language Processing (NLP). Its main task is to enable a computer to describe the visual scene content of an image using natural sentences that conform to human description habits, and has been applied in many fields, such as image retrieval [1], automatically generating radiology reports [2], surgical captioning [3], etc. With the development of deep learning, great progress has been made in image captioning. However, some challenges remain that impact the quality of the generated sentences, e.g., the spatial position representation of the visual feature.

Most recent image caption models have adopted the pipeline structure of "encoder + decoder". Encoders are used to extract and refine the visual features of the image, which usually uses a visual feature extraction network (e.g., convolutional neural network, CNN),

while decoders are used to decode visual features and generate natural sentences, and usually adopt natural language models (e.g., Long Short-Term Memory, LSTM). In early approaches, there was a tendency to use LSTM as the sequence encoder, such as in the NIC [4], Attention [5], and Adaptive [6] models. In recent studies, such as W. Zhang et al. [7], G. Li et al. [8], and Wei Liu et al. [9], Transformer [10] has become the mainstream sequence decoder because of its powerful parallel computing capabilities.

However, no matter which natural language model is used, the problem of spatial position loss cannot be avoided. The image captioning task requires visual features to be fed into the decoder to generate natural sentences, but only sequence features can be recognized and decoded by the decoder. A sequence feature has only one dimension in space, namely length, whereas an image is a 2D feature, with two dimensions, rows and columns (also known as width and height). In order to make the visual features recognized by the decoder, the visual features need to be segmented and then concatenated in rows to achieve serialization. Obviously, this process leads to a lack of spatial position of visual features, which makes the decoder unable to make full use of the necessary spatial positional information of the image and eventually leads to the generated sentence being semantically incomplete.

This limitation is particularly noticeable in Transformer compared to LSTM. LSTM is a recursive model that can solve the problem of long dependencies. Its internal structure is composed of multiple memory cells. The memory unit can retain the necessary historical information in a gated manner and forward it to the next time step as needed. That is, in LSTM, each token in the sequence has access to the necessary information about all tokens that came before it. Thus, for the serialized visual features, the LSTM can implicitly provide information about the positions along the row direction. However, Transformer is an architecture designed in parallel to improve computational efficiency. It gives equal attention to each token in the input sequence and does not have the ability to distinguish positional differences. As a result, Transformer cannot take advantage of either row spatial positional information or column spatial positional information of visual features, which has a certain impact on the quality of the generated sentences. In the original work on Transformer [10], the authors designed a sequence-based positional encoding (SPE) to compensate for the shortcomings of position awareness. This method effectively provides the position information for the sequence by encoding the position with sine and cosine functions alternately. However, for visual features, this method can only provide position information in the row direction, and still cannot solve the problem of column spatial position loss caused by the serialization, as shown in Figure 1a.

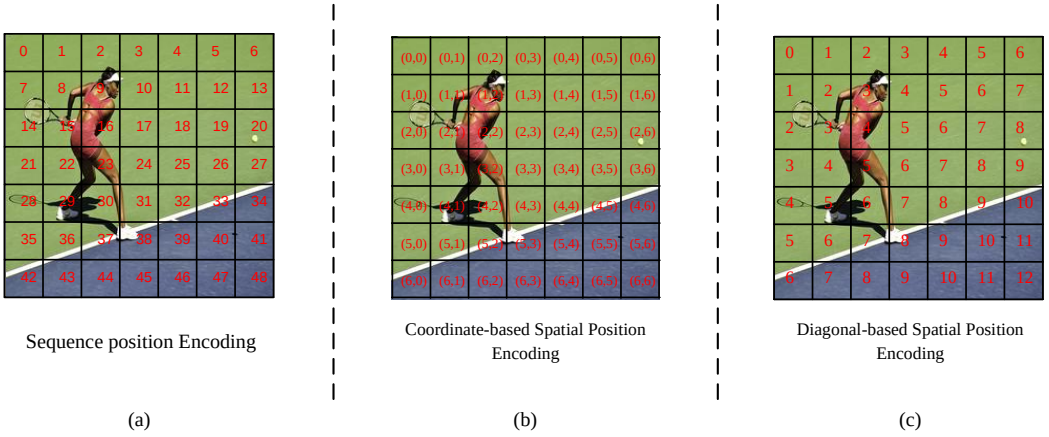


Figure 1. Diagrams of three types of position encoding. (**a**) Sequence position encoding. Position encoding is directly performed sequentially for each pixel after flattening the visual feature map. (**b**) Coordinate-based Spatial Position Encoding encodes the feature pixel's position separately by row and column coordinates in the feature map. (**c**) Diagonal-based Spatial Position Encoding encodes the position along the diagonal direction of the feature map. Here, H and W denote the height and width of the feature map, respectively, and they are set to 7.

Inspired by the insight gained by the introduction of the anchor query in object detection [11], in this paper, we propose a coordinate-based spatial position encoding method (CSPE) to address the limitation discussed above. CSPE creates coordinates for each pixel in the visual feature in both row and column directions, i.e., (x, y), as shown in Figure 1b. CSPE then encodes the row and column coordinates via trainable or hard encoding to represent the row and column position information of the visual feature. Eventually, row and column positional information are incorporated to obtain spatially complete positional information via addition or concatenation operations. Extensive experiments prove that CSPE provides effective spatial position information for visual features, making the decoder pay more attention to the details in visual features and generate sentences with higher granularity. At the same time, CSPE needs to encode the row and column coordinates separately, which has a high time cost. However, when CSPE uses the addition operation, the pixels in images that are symmetric about the diagonal obtain the same position encoding value. For example, the row and column position encoding of the pixels in locations (1,2) and (2,1) in Figure 1b add up to the same value (1 + 2 = 2 + 1). Based on this insight, we designed the diagonal-based spatial encoding (DSPE) approach, aiming to explore whether the time complexity of the model can be reduced by reducing the number of encoding instances while maintaining the same performance. DSPE builds positional markers in the diagonal direction, as shown in Figure 1c, and encodes positional markers via trainable or hard encoding to generate spatial positional information. Experimentally, DSPE performs slightly worse than CSPE but has a faster computation speed. Our overall architecture is shown in Figure 2.

On the MS COCO 2014 dataset [12], CSPE and DSPE outperform a baseline model without sequence-based position encoding in terms of CIDEr metric by 5.7% and 5.4%, respectively, demonstrating a boost in the quality of the generated description sentences, making them closer to the real annotations. In addition, we verify the robustness and plug-and-play ability of the proposed methods on the IU X-RAY12 dataset [13].

In summary, the main contributions of this paper are as follows.

- Firstly, we analyze the reasons for the lack of spatial positional information and the shortcomings of sequence position encoding in image captioning. Furthermore, we clarify the necessity of using spatial-based position encoding.
- We propose a new spatial-based position encoding approach, CSPE, that encodes the position information of visual feature pixels of both the rows and columns of the visual feature map, effectively enhancing the position representation of the visual feature. In addition, in order to improve computational efficiency, we also explore another approach, diagonal-based spatial positional encoding (DSPE).
- We conduct extensive experiments on the MS COCO 2014 and IU X-RAY datasets to validate the effectiveness and robustness of our method, respectively.

The rest of this paper is organized as follows. Related works are reviewed in Section 2. Section 3 details the overall architecture of the baseline , CSPE, and DSPE models for spatial-based position encoding. Section 4 describes the experimental data and setup, presents the ablation studies and results, and then discusses some of the issues that affect performance. Finally, a concise summary and outlook for future work are drawn in Section 5.

2. Related Work

2.1. Models

Many current advances in image captioning have benefited from the adoption of the "encoder + decoder" pipeline. One type of approach, called CNN + RNN (or LSTM), adopted a CNN [14] to encode a semantic image representation and a Recurrent neural networks (RNN) [15] or LSTM to decode the representation into a descriptive sentence. For example, NIC [4] used a GoogLeNet to extract an image feature, which was then fed into a decoder LSTM to generate the sentence, which became the pioneering work of the CNN+RNN model. Later, K. Xu et al. [5] introduced an attention mechanism to enhance the understanding of image semantic information, which improves the richness of the

description vocabulary. However, these methods face image information loss problems and ignore spatial information relevant to captions. A second type of approach, called Detection + RNN (or LSTM), uses object features generated by a region proposal network [16]. Other models, such as [17], further learned to locate attention regions highly related to the semantic content to help prediction. This method is able to capture more objects contained in the image and improve the frequency of these objects in the description sentence. Some authors (e.g., [18,19]) managed to explore visual relationships based on Faster-RCNN, which can locate the relationship between objects more accurately and reflect this in the generated sentences. However, both of the above types of approach decode the visual information used to generate sentences step by step, and are, therefore, slow in both training and inference.

Recently, the application of Transformer in the field of image captioning has attracted much attention. The authors of [9] proposed CPTR, a full Transformer network that is totally convolution-free, to replace the CNN in the encoder, which simultaneously improves model performance and training speed. Wei et al. [20] and Liang et al. [21] implanted repetitive attention and double attention into the Transformer, respectively. These methods make the model focus more accurately on the position of interest, and ultimately improve the accuracy of the description sentence. Lorenzo et al. [22] proposed a pure transformer (CapFormer) architecture for remote sensing image captioning. Specifically, a scalable vision transformer was adopted for image representation, which enhances the generator's capture of tiny goals and their relations and is embodied in the description sentences. However, Transformer is not position-aware unlike LSTM. To take advantage of Transformer and remedy this limitation, in our work, we propose spatial positional encoding and verify its effectiveness based on the CNN+Transformer pipeline.

2.2. Position Encoding

Historically, in the context of NLP, RNN and LSTM inherently take the order of words into account by parsing a sentence word by word in a sequential manner. Many researchers used RNN or LSTM to keep track of ordinal dependencies in sequential data [23]. More recently, researchers incorporated attention mechanisms into the RNN or LSTM encoder–decoder architectures to better extract contextual information for each word in salient position regions. In essence, RNN and LSTM have an inbuilt mechanism that deals with the word order in sentences, thus LSTM and its variants are still used in many state-of-the-art models like LM-BPNN [24] and PF-BiGRU-TSAM [25]. However, the Transformer model does not use recurrence or convolution and treats each data point as independent of all other data points, so its authors adopted explicitly a sequence-based position encoding (SPE) scheme to maintain the knowledge of the order of objects in a sequence. In recent years, Shaw et al. [26] focused on the relative relationship between objects and proposed a relative position encoding to replace the absolute position encoding in the original Transformer. Compared with absolute position encoding, this method can be applied to sequences of any length, which improves the fluency in machine translation. However, this method needs to modify the APIs of Transformer and increases the computational time cost. In the field of computer vision, unlike with SPE in NLP, a Transformer-based model needs to flatten the visual feature map into the sequence shape, so position encoding is a necessary component. For example, image classification models like ViT [27] and Swin [28] used a learnable position encoding. This approach builds a matrix with the same scale as the visual features and optimizes its parameters during model training to refine the representation of positional features. The method is simple to implement and effectively improves the accuracy of image classification. The drawback is that this method may provide the same interpolation at different locations due to random initialization. Recently, the object detection model DETR [11] migrated the spatial position embedding of the anchor query from its center coordinates (x, y). This embedded encoding method avoids the problem of repeated interpolation that may occur in the learnable method and

effectively improves the identification of the target. Similarly, we adopt the coordinate encoding of visual pixels to represent the position information of the feature map.

3. Methods

In this section, we firstly describe the preliminary steps of our work, which constitutes our baseline. Then, we describe the implementation of coordinate-based spatial position encoding. Finally, we explore another, diagonal-based, position encoding approach.

3.1. Preliminary

Our model follows the "encoder and decoder" pipeline architecture of CNN + Transformer, where the encoder consists of the feature extraction network ResNet101 as the CNN backbone and the original standard Transformer's encoder network as the visual feature enhancement module, while the standard Transformer's decoder network is our decoder to generate a sentence. In our experiment, our baseline selects stage-4 of ResNet101 as the backbone output, the number of channels is set to 2048, and the shape is $12 * 12$. For better performance, the enhancement module is stacked by 9 layers of Transformer blocks while the decoder is set to 3 layers. Thus, the final output of the encoder is $(12 * 12, 512)$. We maintain the original absolute SPE in both the enhanced module for visual feature extraction and the decoder for text query embedding, the numbers of channels of which are 2048. The shape of visual SPE is identified as the sequential visual feature as in language embedding. The hard coding formula of the absolute SPE formula [10] is as follows:

$$\begin{aligned} PE_{(pos,2i)} &= sin(pos/10{,}000^{2i/dim}) \\ PE_{(pos,2i+1)} &= cos(pos/10{,}000^{2i/dim}). \end{aligned} \tag{1}$$

Here, pos denotes the sequence position of the visual pixel, i denotes the vector dimension corresponding to the sequence position, where the odd bits correspond to a cosine curve and the even bits correspond to a sine curve, and their wavelengths present a geometric progression from 2 Π to 10,000 Π, dim denotes the feature dimension. The value domain of the sine and cosine functions is $[-1,1]$, which limits the size of the position encoding and makes the training process more stable [10]. Meanwhile, the periods of trigonometric functions in different dimensions are different, which helps in obtaining the relative position information among sequence elements in the same dimension.

In our baseline, the visual feature outputted from the backbone is fused with the hard encoding to provide position information for the visual features. The overall framework model is shown in Figure 2.

3.2. Coordinate-Based Spatial Position Encoding

As outlined in Section 1, RNNs (LSTM) inherently take word order into account. In the context of NLP, this means that they parse a sentence word by word in a sequential manner. In the image captioning task, the spatial-based position encoding that we build into the Transformer-based model will help the model to build better relationship between visual pixels and words in the resulting descriptive sentence, as discussed in Section 2. Therefore, motivated by the vectoring of the center coordinate (x, y) of the anchor query [11], we encode the position information of a visual feature pixel using two axes, row and column.

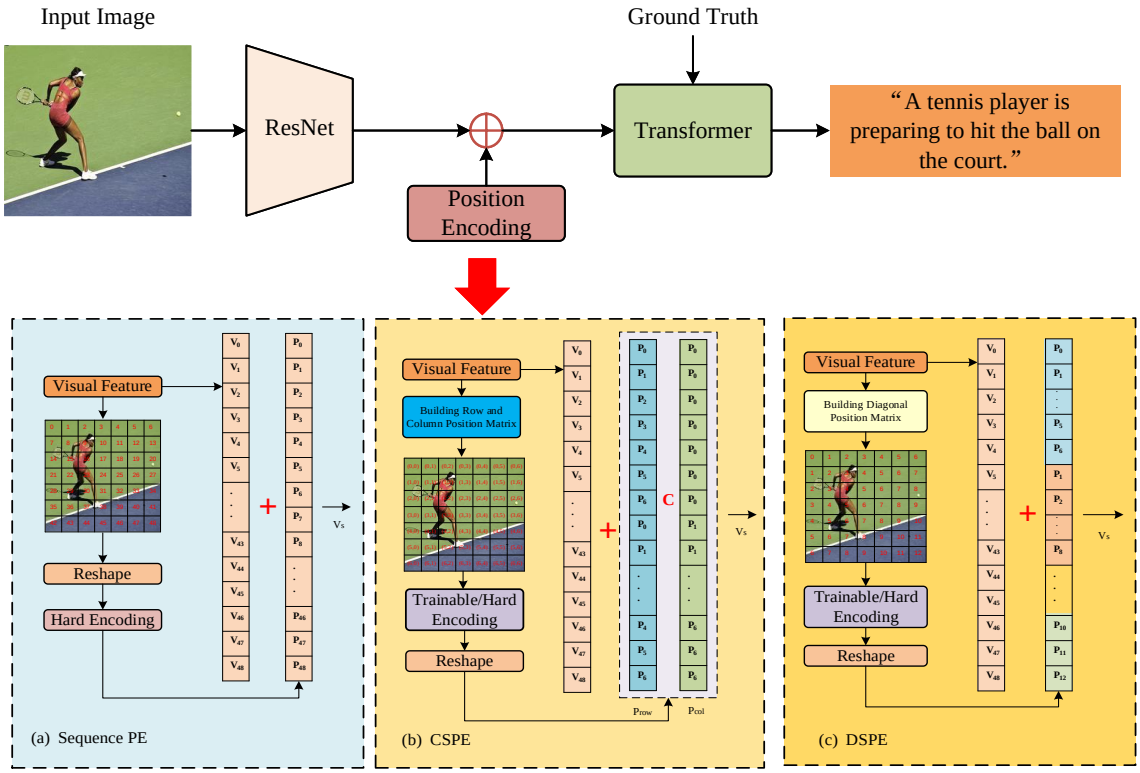


Figure 2. General architecture of the model and the three position encoding methods adopted in each version of the model. The image is first input into ResNet to extract features, then combined with the position encoding, and then input into Transformer for decoding to obtain the description sentences. (**a**) Sequence position encoding (SPE): the position labels of visual features are provided row by row, and their feature dimensions are expanded by hard coding, and finally they are added to the visual features. (**b**) Coordinate-based spatial position encoding (CSPE): First, 2D coordinates are provided for each visual pixel, and then the row and column position encoding are encoded separately. Finally, the row and column position encoding are concatenated and added to the visual feature. (**c**) Diagonal-based spatial positional encoding (DSPE): this provides location markers for visual features along the diagonal direction, then encodes them, before, finally adding them to the visual features.

Given the output visual feature map F = (B, C, W, H) from the backbone, in which B denotes the batch size and C represents the channel number, set to 1 and 2048, respectively, and W and H are the width and height of the visual feature map, both of which are set 7 for convenient display, first, we predefine the spatial position matrix of a row P_{row}, which has the same shape as F, as $P_{i,j}$, the coordinate of the position element at the $i-th$ row and $j-th$ column. Then, we assign $(0 \sim (W-1))$ to all pixels from left to right in each row, $(P_0, P_1, :::; P_i, :::, P_{W-1})$, and then, perform the reshape flattening operation to obtain the row position sequence $P_{row} = (P_0, P_1; P_2; ::: P_{W-1}; :::; P_0, P_1; P_2; :::; P_{W-1})$. We can obtain the sequence of column positions $P_{col} = (P_0, P_1, P_2, :::, P_{H-1}; :::; P_0, P_1, P_2, :::, P_{H-1})$ by the same procedure, which is to say, assign the $(P_0 \sim P_{H-1})$ to the visual feature pixel from top to bottom in each column. Subsequently, the row position encoding and column position encoding are vectorized and fused by Equation (1) and Equation (2), respectively. This gives us the final 2D spatial position vector representation $P_{CSPE} = (P_{0,0}, P_{0,1}, :::, P_{0,W-1}; : P_{i,j} :; P_{H-1,0}, ::: P_{H-1,W-2}, P_{H-1,W-1})$, $i \in [0, W-1]$, $j \in [0, H-1]$.

$$P_{CSPE} = M(PE(P_{row}), PE(P_{col})). \quad (2)$$

Here, M denotes the fusing operation of concatenation or addition, PE is the embedding function that vectorizes our generated position codes, and P_{CSPE} has the same dimension as the visual information sequence, whose dimension is (B, L, C), $L = W * H$.

Next, we perform the reshape operation on the feature map F to get $V_F = (V_{0,0}, V_{0,1}, ..., V_{i,j}, ..., V_{H-1,W-1})$, $V_{i,j} \in R^{2048}$, and then fuse this with our 2D spatial position coding P_{CSPE} by Equation (3), which is the same as that in [7] and defined as:

$$V_{spatial} = V_F + P_{CSPE}. \tag{3}$$

Finally, the sequence visual feature with spatial information is forwarded to the feature enhancement modular or Transformer-based decoder, to further enhance the visual feature or directly generate a description sentence. The detailed procedure is shown in Algorithm 1.

Algorithm 1 Coordinate-based Spatial Position Encoding (CSPE)

Require: F is the feature map, its shape is (B, W, H, C), where B is Batch, W and H are Width and Height, and C is channel.

Ensure: $V_{spatial}$ is the sequence features with spatial position information, its shape is (B, L, C), where L is the Length. $L = W * H$.

1: Predefine the row position matrix, as $\begin{bmatrix} 0 & 1 & 2 & 3 & \cdots & W-1 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 1 & 2 & 3 & \cdots & W-1 \end{bmatrix}$, the shape is $(1, W, H)$.

2: Reshape it to a sequence, denoted as P_{row}, the shape is $(1, L)$.

3: Predefine the column position matrix, as $\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 2 & 2 & \cdots & 2 \\ \vdots & \vdots & \cdots & \vdots \\ H-1 & H-1 & \cdots & H-1 \end{bmatrix}$, the shape is $(1, W, H)$.

4: Reshape it to a sequence, denoted as P_{col}, the shape is $(1, L)$.

5: Encode and concat P_{row} and P_{col} by Equation (3), and repeat this in the Batch dimension, the shape is (B, L, C), denoted as P_{CSPE}.

6: Reshape F to a sequence, denoted as $V_{original}$, the shape is (B, L, C).

7: $V_{spatial} = V_{original} + P_{CSPE}$

8: **return** $V_{spatial}$

3.3. Diagonal-Based Spatial Position Encoding

The CSPE described in the previous section effectively prevents the loss of spatial position information by encoding the position of the feature map in rows and columns separately. However, it needs to encode twice for the two axes, row and column, which increases the time complexity. In fact, it is possible to encode the feature map only once while ensuring that the 2D spatial position information is retained. In addition, we consider that the spatial location information of the visual feature has a triangular symmetry and that the column encoding can be represented, for example, using the transposition of the row encoding of the first method described above. Thus, we explore another, diagonal-based, position encoding approach. First, the original visual information of the image is extracted by the encoder and reshaped into a sequence, denoted as $V_{original} = (V_{0,0}, V_{0,1}, V_{0,2}, ..., V_{H-1,W-2}, V_{H-1,W-1})$, $V_{i,j} \in R^{2048}$, with dimensions (B, L, C). Next, we use $F_{(i,j)}$ $(i \in [0, W-1], j \in [0, H-1])$ to represent any pixel information on the feature map, and encode the position along the diagonal direction of the feature map (the direction of $F_{0,0}, F_{1,1} \sim F_{H-1,W-1}$), as shown in Figure 2c. The pixel points at position $F_{0,0}$ are assigned code mark 0, those at $F_{0,1}, F_{1,0}$, code mark 1, those at $F_{0,2}, F_{1,1}, F_{2,0}$, code mark 2, and so on. Then, we reshape the position encoding to obtain the sequence of position information, as $P_{diagonal} = (P_0, P_1, ..., P_{W-1}, ..., P_{H-1}, ..., P_{(H-1)+(W-1)})$. Next, we

vectorize $P_{diagonal}$ based on Equation (1). Finally, the visual information sequence and the position information sequence are fused to obtain the brand new spatial visual information sequence $V_{spatial}$ as:

$$V_{spatial} = V_{original} + PE(P_{diagonal}). \tag{4}$$

The new $V_{spatial}$ already has 2D spatial position information, and it is then decoded to generate the descriptive sentence of the corresponding image. The specific implementation is shown in Figure 2c and Algorithm 2. Ablation experiments prove that DSPE is slightly better than CSPE in terms of time cost, but its performance is slightly inferior. The results of the ablation experiments are shown in Table 1.

Algorithm 2 Diagonal-based Spatial Position Encoding (DSPE)

Require: F is the feature map, its shape is (B, W, H, C), where B is Batch, W and H are Width and Height, and C is channel.

Ensure: $V_{spatial}$ is the sequence features with spatial position information, its shape is (B, L, C), where L is the Length. $L = W * H$.

1: Reshape F to a sequence, noted as $V_{original}$, the shape is (B, L, C).

2: Predefine a diagonal marker matrix, as $\begin{bmatrix} 0 & 1 & 2 & 3 & \cdots & W-1 \\ 1 & 2 & 3 & \cdots & \cdots & W \\ 2 & 3 & \ddots & \cdots & \cdots & W+1 \\ 3 & \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ H-1 & H & H+1 & \cdots & \cdots & H+W-2 \end{bmatrix}$, the shape is $(1, W, H)$.

3: Reshape it to a sequence, and repeat on the Batch dimension, denoted as $P_{diagonal}$, the shape is (B, L).

4: Encode $P_{diagonal}$ and add it to $V_{original}$ by Equation (4), denoted as $V_{spatial}$.

5: **return** $V_{spatial}$

Table 1. Results of ablation experiments. All scores are shown as percentages.

Model	B1	B2	B3	B4	M	R	C	S	W
Baseline	72.4	55.6	42.0	31.8	25.8	53.4	98.5	18.7	54.3
Baseline + SPE (hard)	73.6	56.9	43.4	33.2	26.5	54.6	103.3	19.5	55.2
Baseline + SPE (trainable)	73.6	57.0	43.4	33.2	26.5	54.6	103.7	19.5	55.1
Baseline + CSPE (hard)	74.0	57.5	43.9	33.6	26.8	54.9	104.2	19.7	55.5
Baseline + CSPE (trainable)	74.0	57.5	43.9	33.6	26.6	54.8	103.9	19.6	55.3
Baseline + DSPE (hard)	73.6	56.9	43.5	33.4	26.6	54.7	103.4	19.6	55.2
Baseline + DSPE (trainable)	74.0	57.5	43.9	33.5	26.6	54.8	103.9	19.7	55.3

4. Experiments

4.1. Datasets

In the experiments, we use the MS COCO 2014 [12] dataset for model training and validation. This dataset is the most commonly used dataset in image captioning and contains 82,783 images in the training set, 40,504 images in the validation set, and 40,775 images in the test set, each containing five descriptive sentences. We also use MS COCO 2014 after "Karpathy splits" [29], which consists of 113,287, 5000, and 5000 images for training, validation, and testing, respectively. Moreover, to validate the robustness of our methods in automatically generating radiology reports applications, we chose the IU X-RAY [13] dataset, which is a publicly available dataset collected and collated by Indiana University and includes 7470 chest radiographs and corresponding 3955 pathology report annotations. We present a summary of the properties of the datasets we used in Table 2.

Table 2. Summary of the properties of the datasets we used, including the number of images contained in each subset and the number of annotations.

Dataset	Training	Validation	Test	Annotation
MS COCO 2014 [12]	82,783	40,504	40,775	5 per image
Karpathy splits [29]	113,287	5000	5000	5 per image
IU X-RAY [13]	7470	-	-	3955 in total

4.2. Evaluation Metrics

In our work, we use the full range of image captioning evaluation metrics, including BLEU [30], METEOR [31], ROUGE- [32], CIDEr [33], SPICE [34], and the latest WMD [35]. BLEU measures the quality of predicted sentences by matching n-gram correlations between predicted sentences and manually labeled sentences. METEOR considers the accuracy, recall, and arrangement of matching marks. ROUGE focuses on the fluency and sufficiency of sentences generated. CIDEr is a combination of BLEU and a vector space model. SPICE is used to measure how image titles effectively restore objects, attributes, and their relationships. WMD is a proposed metric based on word2vec [36] and text travel distance, which is used to measure the similarity between the generated sentences and the annotated headings.

4.3. Settings

In our ablation study, we use the ResNet101 network as the encoder of the model. We remove the classification head from ResNet101 and add a new linear layer to obtain 2048-dimensional feature vectors for each image. We use Transformer as the decoder and replace the original SPE in the encoding layer with CSPE and DSPE. We use the Adam optimizer to train all models with the input image of 384 * 384, where the betas parameter is set to (0.9, 0.999), eps is set to 1×10^{-8}, and the weight decay is set to 0. In the first stage, we train 7 epochs with a 3×10^{-5} learning rate, and use a 7.5×10^{-6} learning rate for a further 4 epochs. The batch size is set to 20. We also adopt beam search for better sentences. Our models are built and run in Python3.7, Pytorch1.8.1, and Cuda11.4. The hardware employs a single GeForce RTX 3090 GPU (made by Nvidia in Santa Clara, CA, USA) for model training.

4.4. Ablation Study

4.4.1. Performance Comparison of Different Methods

To verify the effectiveness of CSPE and DSPE under appropriate conditions, we conduct extensive ablation experiments, consisting of (1) without adding any position encoding to the baseline; (2) with sequence-based position encoding SPE; and (3) with our two methods of CSPE and DSPE. In addition, we validate the performance of our encoding methods with two main modes of current mainstream position encoding approaches, i.e., hard coding (also called absolute computing in [10]) and trainable coding. All experiments are optimally trained using cross-entropy loss with the same hyper-parameters on MS COCO 2014.

Table 1 shows the results of our experiments. The model with SPE achieves better results than the baseline, obtaining a 1.2% improvement (from 72.4% to 73.6%) in BLEU-1, and a 1.4% improvement from 31.8% to 33.2% in BLEU-4, which verifies that position encoding is an important part for an encoder-Transformer model. Compared to the baseline without any position encoding, the model with CSPE and DSPE obtained higher scores on various metrics, such as BLEU-4 (from 31.8% to 33.6% and 33.8%) and CIDEr (from 98.5% to 104.2% and 103.9%). Compared with the model with SPE, both of our methods obtain about a 0.4% improvement in BLEU-1 (from 73.6% to 74.0%), and 0.9% and 0.6% improvements in CIDEr (from 103.3% to 104.2% and 103.9%, respectively), which proves that our methods can effectively protect the 2D spatial position information of each pixel on the feature map. In addition, we found that the trainable and hard coding methods

yield nearly the same results within the error tolerance. For example, they obtain the same scores in BLEU-1, BLEU-4, METEO, ROUGE, and SPICE (73.6%, 33.2%, 26.5%, 54.6%, and 19.5%, respectively).

4.4.2. Performance Analysis of M Fusion Methods

As discussed in Section 3, CSPE performs row position encoding and column position encoding, respectively, and then fuses them. The M fusion operation has two methods, addition and concatenation. The former reduces the feature dimension of pixel row and column position encoding by half, and then the rows and columns are stitched together by the concatenation operation, as shown in Figure 2. The new feature dimension is the same as the original one and will not affect the next process. For example, with the CSPE approach, in a $H = 7, W = 7$ feature map, its row position is coded as P_{row}, and its column position is coded as P_{col}, and these are then encoded and combined by Equations (2) and (3). The latter adds the row position features and column position features directly. Using this approach, the spatial position codes of some pixels are symmetrical about the diagonal, for example, F_{12} and F_{21} (F_{ij} denotes the element of the i-th row and j-th column of the feature map), and the 2D position encoding are calculated as $P_{12} = P_1 + P_0$ and $P_{21} = P_0 + P_1$. Obviously, symmetrical pixels have the same encoding markers. Because addition does not disrupt the feature information of the corresponding dimension, F_{12} and F_{21} obtain the same spatial position information. This unreasonable defect leads to ambiguity in the 2D spatial position information and affects the performance of the model. For the concatenation operation, the row and column encoding need to be spliced in the feature dimension. In this way, the first half of the feature dimension represents the row position information, and the second half represents the column position information. Using this approach, they are not confused and more clearly represent the spatial position information augmented by feature position. As shown in Table 3, compared with the addition operation, the concatenation operation achieves higher scores in most metrics except for SPICE, in which the two operations obtain the same score, and even obtains a relative improvement of 1.0% in CIDEr. The results clearly confirm our view that the concatenation operation (that is feature augmentation) is more stable and better than the addition operation (information fusion) in representing 2D position information.

Table 3. Comparison of different feature fusion methods of CSPE and DSPE. All scores are shown as percentages. h indicates hard-coding, t indicates self-learning (trainable), PF indicates position feature, A indicates addition, and C indicates concatenation.

ResNet101 + Transformer	B1	B4	M	R	C	S	W
+CSPE(h)PF(C)	74.0	33.6	26.8	54.9	104.2	19.7	55.5
+CSPE(h)PF(A)	73.5	33.3	26.6	54.7	103.2	19.7	55.2
+DSPE(h)	73.6	33.4	26.6	54.7	103.4	19.6	55.2
+CSPE(t)PF(C)	74.0	33.6	26.6	54.8	103.9	19.6	55.3
+CSPE(t)PF(A)	74.0	33.5	26.5	54.7	103.5	19.5	55.2
+DSPE(t)	74.0	33.5	26.6	54.8	103.9	19.7	55.3

4.4.3. Hard Coding and Trainable

For DSPE, the pixels whose positions are symmetrical about the diagonal will obtain the same feature encoding. In theory, DSPE is a fast version of the addition operation of the CSPE approach. As shown in Table 3, both methods achieve almost the same results in the ablation experiment. For example, both DSPE (Hard) and CSPE (Hard) with PF (ADD) achieve 26.6% in METEOR and 54.7% in ROUGE. Almost identical results were also achieved with the trainable approach. However, since DSPE only needs to perform position encoding once, there is a reduction in computational effort and time cost in training and inference. Table 4 shows the comparison of the speed and parameter count between different approaches.

Table 4. Comparison of Time Cost and Computational Effort between different Approaches. C/s indicates the time consumption for one iteration through the module, in seconds. GS/FPS indicates the speed of sentence generation by the generator, in FPS. Note that since the hard-coded approach does not involve trainable parameters, we denote it by "-" in the table.

Module	C/s	GS/FPS	Parameter
+CSPE(h)PF(A)	0.0632	4.6	-
+DSPE(h)	0.0220	4.9	-
+CSPE(t)PF(A)	0.0082	8.5	49,152
+DSPE(t)	0.0068	10.2	47,104

4.4.4. Qualitative Analysis

In this section, using MS COCO 2014, we employ the base model without/with SPE, CSPE, and DSPE, respectively, to obtain descriptive sentences and compare their differences, as shown in Figure 3. The descriptive sentences generated by the base model have some relevance to the image contents, but some important content is ignored. For example, in the first image, the base model fails to recognize the horse, so the "riding horses" is described as "standing", and in the third image, it ignores the scene in the image. Compared with the base model, the model with SPE captures more visual information, and it can describe the parts that the base model cannot describe. However, it has the problem of misrepresenting some objects; for example, it describes "gloves" as "bat" in the second image. Our proposed methods produce more detailed sentences due to the spatial position information guidance. In the third image, our proposed methods focus not only on the background of the image "grass field", but also on the number of zebras in the image, and use the corresponding descriptive terms "two zebras" or "a couple of zebras". The descriptive sentences obtained by the model with our methods are closer to ground truth. The above analysis proves that position encoding is important for the image captioning task, and spatial-based position encoding can make a model pay more attention to details of the image and achieve more consistency with human description habits than SPE.

4.5. Discussion

There has been a lot of work on position encoding in different vision models or vision tasks, such as DETR [11], DAB [37], and MViT2 [38]. Compared with these models, our CSPE has significant differences. Firstly, the other approaches aim to obtain the center coordinates of the predefined object boxes (anchors), while our methods aim to obtain the 2D coordinates of each pixel on the feature map. Secondly, the other approaches use the position encoding in the query component at each layer of the decoder, while ours is in the self-attention module in the feature enhancement module at each layer of the decoder, which is mainly used to preserve the pixels' spatial position information in the image captioning task. Finally, in terms of operation implementation, the other approaches generally use the addition operation to fuse information, while our approach mainly adopts concatenation to augment feature position in feature maps. The above analysis and experiment results have proved that concatenation has a good effect and rationality. In addition, the Transformer-based classification network ViT [27], swin [28], and others use position encoding to the patches of the image before the feature extractor. This approach belongs to the sequence position encoding category, that is SPE. In contrast, our approach is mainly used after the feature extractor, to solve the 2D spatial position information loss problem when the visual feature map is flattened into a sequence.



Base: a group of people standing on a sandy beach.
Sequence PE: a group of people riding horses on a beach.
GT: two men are riding horses along the beach.
CSPE: a couple of people riding horses on a beach.
DSPE:two people riding horses on a beach.



Base: a baseball player on top of a field .
Sequence PE: a baseball player holding a bat on top of a field .
GT: a catcher squatting at a base with his gloved hand extended.
CSPE: a baseball player holding baseball gloves on a field.
DSPE: a baseball player holding a catchers mitt on top of a field.



Base: a number of zebras in a field near one another.
Sequence PE: a herd of zebra standing on top of a dry grass field.
GT: two zebras standing next to each other in a dry grass field.
CSPE: a couple of zebras standing next to each other in a field of tall grass.
DSPE: two zebras standing next to each other on top of a dry grass field.

Figure 3. Comparison of sentences generated by different models, baseline, SPE, GT, and our CSPE and DSPE, in which GT represents Ground Truth. Where the bold items represent the ground truth given by the dataset and the red parts represent the statements generated by our method.

4.6. Robustness on Medical Application

Image captioning has a wide range of applications in medicine, such as medical imaging. One of the important research directions in medical imaging is automatically generating radiology reports. In [39], the authors extended the image to text model to medical datasets. In order to verify the effectiveness and robustness of our proposed methods in application, we select the memory-driven Transformer [2] model (MD-Transformer), one of the recent advanced models in automatically generating radiology reports applications, as a base model, and change the position encoding in the model with our approaches to validate that our approaches offer an improvement in performance.

MD-Transformer is an image captioning model for automatic generation of radiology reports, which, in common with our proposed method, also employs a ResNet-Transformer and uses sequence position encoding. To ensure a fair comparison, based on the IU X-RAY dataset, we set various hyperparameters to be the same as those used in the original model during training. In order to demonstrate the best improvement in performance, we choose two methods with the best performance as shown by the ablation experiments, CSPE (hard) and DSPE (trainable), to replace the position encoding in the original model.

As shown in Table 5, our methods improve the performance of MD-Transformer. CSPE (hard) demonstrates an improvement of 1.3% in BLEU-3 and 1.7% in BLEU-4, but shows a reduction by 0.1% in ROUGE. Although the DSPE (trainable) method performs poorly on BLEU-1 and BLEU-2 (reduced by 0.5% and 0.1%, respectively), it demonstrates some improvement in other metrics, especially in ROUGE, which is improved by 3.8%. From the above analysis, we can conclude that CSPE (hard) and DSPE (trainable) have improved the performance of the model within the error tolerance, making the reports generated by MD-Transformer [2] more accurate. Our methods have proven to be equally effective for practical medical applications, and have strong portability.

Table 5. Performance improvement with Memory-Driven Transformer. All models are optimized with cross-entropy loss.

Memory-Driven Transformer	B1	B2	B3	B4	M	R
Baseline	47.0	30.4	21.9	16.5	18.7	37.1
+DSPE(trainable)	46.5	30.3	22.8	17.5	19.0	**40.9**
+CSPE(hard)	**47.8**	**31.2**	**23.2**	**18.2**	**19.6**	37.0

4.7. Comparing with the State-of-the-Art

In order to fairly compare our proposed approach with those of the current state-of-the-art models, such as LSTM-A [40], RFNet [41], Up-Down [42], GCN-L [18], LBPF [43], SGAE [19], AoANe [44], X-LAN [45], X-T [45], CBTIC [46], we replace ResNet101 with swin Transformer, remove the last pooling and classification layers, and add a linear layer, a GELU activation function layer, a layer normalization and a dropout layer, as shown in the following formula:

$$x_{out} = Dropout(Layernorm(GELU(Linear(x_{in})))). \quad (5)$$

where x_{in} denotes the input features and x_{out} denotes the output features, both of which have dimensions of (B, L, C).

We add CSPE and DSPE, respectively, to the models. We also use beam search to select the best sentence and the beam-size parameter is 3. The results are reported on the Karpathy test split for offline evaluation, and are shown in Table 6.

Table 6. Comparison of our proposed methods with state-of-the-art models. All models are optimized with cross-entropy loss.

Model	B1	B2	B3	B4	M	R	C	S
LSTM-A [40]	75.4	-	-	35.2	26.9	55.8	108.8	20.0
RFNet [41]	76.4	60.4	46.6	35.8	27.4	56.5	112.5	20.5
UP-Down [42]	77.2	-	-	36.2	27.0	56.4	113.5	20.3
GCN-L [18]	77.3	-	-	36.8	27.9	57.0	116.3	20.9
LBPF [43]	77.8	-	-	37.4	28.1	57.5	116.4	21.2
SGAE [19]	77.6	-	-	36.9	27.7	57.2	116.7	20.9
AoANet [44]	77.4	-	-	37.2	28.4	57.5	119.8	21.3
X-LAN [45]	**78.0**	62.3	48.9	38.2	28.8	58.0	122.0	21.9
X-T [45]	77.3	61.5	47.8	37.0	28.7	57.5	120.0	21.8
CBTIC [46]	**78.0**	62.2	48.5	37.5	**29.1**	58.2	122.6	**22.3**
Ours(DSPE)	77.7	62.1	48.6	38.1	28.9	57.9	122.6	22.0
Ours(CSPE)	77.9	**62.4**	**49.0**	**38.5**	29.0	**58.3**	**123.6**	22.1

Our CSPE achieves better results, and its overall performance is better than most of the current peer models, especially in terms of BLEU-4 and CIDEr scores (38.5% and 123.6%, respectively). Compared with CBTIC's SOTA model, our model achieves competitive results for BLEU-1 (77.9% vs. 78.0%) and ROUGE (58.3% vs. 58.2%), and even outperforms CBTIC in some metrics, such as BLEU-4 and CIDEr, both of which show a 1% improvement. Although DSPE is slightly inferior in performance, it outperforms X-Transform on most evaluation metrics. If we use more advanced backbone networks such as EfficientNet-(v2) [47] and ConNeXt [48], our model will achieve even better performance.

5. Conclusions

In this paper, we analyze why spatial position information is lost in traditional image captioning models, clarify the necessity of spatial position encoding, and propose a coordinate-based spatial position coding (CSPE) approach. CSPE encodes the positional information of feature pixels from both row and column, which effectively enhances the positional representation of visual features. To reduce the time cost, we also explore a

diagonal-based spatial position encoding (DSPE) approach. Compared to CSPE, DSPE is slightly worse in performance but is computationally faster. On MS COCO 2014, CSPE and DSPE boost CIDEr by 5.7% and 5.4%, respectively, over a baseline model without sequence-based spatial encoding, which effectively improves the quality of the generated sentences. In addition, in medical captioning, the proposed methods improve the accuracy of the generated reports, which verifies their plug-and-play ability and robustness. However, the proposed methods cannot be extended to larger-resolution images due to the fixed encoding mode. In future work, we will further explore more optimized positional representations to address this limitation and continue to investigate the impact of visual positional encoding on visual and textual alignment.

Author Contributions: X.Y. contribution: methods, manuscript preparation, equipment resource support, verification, data management. S.H. contribution: experimentation, review, editing, supervision. J.W. contribution: discussion, review. Y.Y. contribution: review; Z.H. contribution: discussion, polish; S.M. contribution: discussion, resource equipment support. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 61741216) and the Shaanxi Province Qinchuangyuan "Scientist + Engineer" Team Construction Project (Grant No. 2023KXJ-241).

Data Availability Statement: Our data and code can be obtained from the following link: https://github.com/bayi233/Spatial-based-PE.

Acknowledgments: We also thank the open-source community for their contributions.

Conflicts of Interest: The authors declare no conflict of interest..

References

1. Datta, R.; Joshi, D.; Li, J.; Wang, J.Z. Image retrieval: Ideas, influences, and trends of the new age. *ACM Comput. Surv.* **2008**, *40*, 1–60. [CrossRef]
2. Chen, Z.; Song, Y.; Chang, T.H.; Wan, X. Generating Radiology Reports via Memory-driven Transformer. *arXiv* **2020**, arXiv:2010.16056.
3. Xu, M.; Islam, M.; Ren, H. Rethinking surgical captioning: End-to-end window-based mlp transformer using patches. In Proceedings of the Medical Image Computing and Computer Assisted Intervention—MICCAI 2022: 25th International Conference, Singapore, 18–22 September 2022; Proceedings, Part VII; Springer Nature: Cham, Switzerland, 2022.
4. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3156–3164.
5. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2048–2057.
6. Lu, J.; Xiong, C.; Parikh, D.; Socher, R. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 375–383.
7. Zhang, W.; Nie, W.; Li, X.; Yu, Y. Image Caption Generation With Adaptive Transformer. In Proceedings of the 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Jinzhou, China, 6–8 June 2019; pp. 521–526.
8. Li, G.; Zhu, L.; Liu, P.; Yang, Y. Entangled transformer for image captioning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, South Korea, 27 October–2 November 2019; pp. 8928–8937.
9. Liu, W.; Chen, S.; Guo, L.; Zhu, X.; Liu, J. CPTR: Full Transformer Network for Image Captioning. *arXiv* **2021**, arXiv:2101.10804.
10. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, ÅĄ.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. Available online: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (accessed on 1 June 2017).
11. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
12. Lin, T.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; DollÃąr, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
13. Demner-Fushman, D.; Kohli, M.D.; Rosenman, M.B.; Shooshan, S.E.; Rodriguez, L.; Antani, S.; Thoma, G.R.; McDonald, C.J. Preparing a collection of radiology examinations for distribution and retrieval. *J. Am. Med. Inform. Assoc.* **2016**, *23*, 304–310. [CrossRef] [PubMed]

14. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
15. Medsker; Larry, R.; Jain, L.C. Recurrent neural networks. *Des. Appl.* **2001**, *5*, 64–67.
16. Zhao, X.; Li, W.; Zhang, Y.; Gulliver, T.A.; Chang, S.; Feng, Z. A faster RCNN-based pedestrian detection system. In Proceedings of the 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), Montreal, QC, Canada, 18–21 September 2016.
17. 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, Las Vegas, NV, USA, 27–30 June 2016.
18. Yao, T.; Pan, Y.; Li, Y.; Mei, T. Exploring visual relationship for image captioning. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
19. Xu, Y.; Tang, K.; Zhang, H.; Cai, J. Auto-encoding scene graphs for image captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10685–10694.
20. Wei, H.; Li, Z.; Zhang, C.; Ma, H. The synergy of double attention: Combine sentence-level and word-level attention for image captioning. *Comput. Vis. Image Underst.* **2020**, *201*, 103068. [CrossRef]
21. Liang, Y.; Hu, H. Visual Skeleton and Reparative Attention for Part-of-Speech image captioning system. *Comput. Vis. Image Underst.* **2019**, *189*, 102819.
22. Lorenzo, J.; Alonso, I.P.; Izquierdo, R.; Ballardini, A.L.; Saz, ÃĄH.; Llorca, D.F.; Sotelo, M.ÃĄ. Capformer: Pedestrian crossing action prediction using transformer. *Sensors* **2021**, *21*, 5694. [CrossRef] [PubMed]
23. Ilya, S.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, arXiv:1409.3215.
24. Zhang, J.; Tian, J.; Alcaide, A.M.; Leon, J.I.; Vazquez, S.; Franquelo, L.G.; Luo, H.; Yin, S. Lifetime Extension Approach Based on Levenberg–Marquardt Neural Network and Power Routing of DC-DC Converters. *IEEE Trans. Power Electron.* **2023**, *38*, 10280–10291. [CrossRef]
25. Zhang, J.; Huang, C.; Chow, M.Y.; Li, X.; Tian, J.; Luo, H.; Yin, S. A Data-model Interactive Remaining Useful Life Prediction Approach of Lithium-ion Batteries Based on PF-BiGRU-TSAM. *IEEE Trans. Ind. Inform.* **2023**. [CrossRef]
26. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-Attention with Relative Position Representations. *arXiv* **2018**, arXiv:1803.02155.
27. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
28. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
29. Karpathy, A.; Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3128–3137.
30. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002.
31. Denkowski, M.; Lavie, A. Meteor universal: Language specific translation evaluation for any target language. In Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, MD, USA, 26–27 June 2014.
32. Lin, C. Rouge: A package for automatic evaluation of summaries. Text summarization branches out 2004, pp. 74–81. Available online: https://aclanthology.org/W04-1013/ (accessed on 1 July 2004).
33. Ramakrishna, V.; Lawrence, C.; Parikh, D. Cider: Consensus-based image description evaluation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 15 October 2015; pp. 4566–4575.
34. Anderson, P.; Fernando, B.; Johnson, M.; Gould, S. Spice: Semantic propositional image caption evaluation. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 382–398.
35. Kusner, J.; Sun, Y.; Kolkin, N.; Weinberger, K. From word embeddings to document distances. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 957–966.
36. Mikolov, T.; Sutskever, I.; Kai, C.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. *Adv. Neural Inf. Process. Syst.* **2013**. [CrossRef]
37. Liu, S.; Li, F.; Zhang, H.; Yang, X.B.; Qi, X.; Su, H.; Zhu, J.; Zhang, L. DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR. *arXiv* **2022**, arXiv:2201.12329.
38. Li, Y.; Wu, C.; Fan, H.; Mangalam, K.; Xiong, B.; Malik, J.; Feichtenhofer, C. Improved Multiscale Vision Transformers for Classification and Detection. *arXiv* **2021**, arXiv:2112.01526.
39. Daniela, O.; Adriana, B.; Dinu, L.P. Towards Mapping Images to Text Using Deep-Learning Architectures. *Mathematics* **2020**, *8*, 1606.
40. 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. 4894–4902.
41. Jiang, W.; Ma, L.; Jiang, Y.; Liu, W.; Zhang, T. Recurrent fusion network for image captioning. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 499–515.
42. Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; Zhang, L. Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6077–6086.

43. Qin, Y.; Du, J.; Zhang, Y.; Lu, H. Look back and predict forward in image captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8367–8375.
44. Huang, L.; Wang, W.; Chen, J.; Wei, X. Attention on attention for image captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10971–10980.
45. Pan, Y.; Yao, T.; Li, Y.; Mei, T. X-linear attention networks for image captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2020; pp. 10971–10980.
46. Zhou, Y.; Hu, Z.; Liu, D.; Ben, H.; Wang, M. Compact Bidirectional Transformer for Image Captioning. *arXiv* **2022**, arXiv:2201.01984.
47. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
48. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.