applied sciences

MDPI

Article

Sequential Brain CT Image Captioning Based on the Pre-Trained Classifiers and a Language Model

Jin-Woo Kong [1], Byoung-Doo Oh [2], Chulho Kim [3] and Yu-Seop Kim [1,*]

[1] Department of Convergence Software, Hallym University, Chuncheon-si 24252, Gangwon-do, Republic of Korea; kongjw0110@gmail.com
[2] Cerebrovascular Disease Research Center, Hallym University, Chuncheon-si 24252, Gangwon-do, Republic of Korea; iambd822@gmail.com
[3] Department of Neurology, Chuncheon Sacred Heart Hospital, Chuncheon-si 24253, Gangwon-do, Republic of Korea; gumdol52@hallym.or.kr
* Correspondence: yskim01@hallym.ac.kr

Abstract: Intracerebral hemorrhage (ICH) is a severe cerebrovascular disorder that poses a life-threatening risk, necessitating swift diagnosis and treatment. While CT scans are the most effective diagnostic tool for detecting cerebral hemorrhage, their interpretation typically requires the expertise of skilled professionals. However, in regions with a shortage of such experts or situations with time constraints, delays in diagnosis may occur. In this paper, we propose a method that combines a pre-trained CNN classifier and GPT-2 to generate text for sequentially acquired ICH CT images. Initially, CNN undergoes fine-tuning by learning the presence of ICH in publicly available single CT images, and subsequently, it extracts feature vectors (i.e., matrix) from 3D ICH CT images. These vectors are input along with text into GPT-2, which is trained to generate text for consecutive CT images. In experiments, we evaluated the performance of four models to determine the most suitable image captioning model: (1) In the N-gram-based method, ReseNet50V2 and DenseNet121 showed relatively high scores. (2) In the embedding-based method, DenseNet121 exhibited the best performance. (3) Overall, the models showed good performance in BERT score. Our proposed method presents an automatic and valuable approach for analyzing 3D ICH CT images, contributing to the efficiency of ICH diagnosis and treatment.

Keywords: intracerebral hmorrhage; medical image captioning; deep learning; convolutional neural network; GPT-2

Citation: Kong, J.-W.; Oh, B.-D.; Kim, C.; Kim, Y.-S. Sequential Brain CT Image Captioning Based on the Pre-Trained Classifiers and a Language Model. *Appl. Sci.* **2024**, *14*, 1193. https://doi.org/10.3390/app14031193

Academic Editors: Charles Tijus, Kuei-Shu Hsu, Teen-Hang Meen, Po-Lei Lee and Chun-Yen Chang

Received: 15 December 2023
Revised: 28 January 2024
Accepted: 30 January 2024
Published: 31 January 2024

1. Introduction

Intracerebral hemorrhage (ICH) is a severe cerebrovascular disorder where blood vessels within the brain rupture, leading to bleeding in the brain tissue. It accounts for 10–30% of strokes and exhibits high incidence and mortality rates. Additionally, the brain tissue is highly sensitive, so when bleeding occurs, the tissue can easily be damaged, potentially impairing or even halting brain function, posing a direct threat to the patient's life [1–5]. ICH is primarily diagnosed by analyzing computed tomography (CT) images. This is because CT images sequentially capture images from the beginning to the end of the target, enabling precise confirmation of the presence and location of ICH and facilitating rapid examinations. However, if the diagnosis is delayed, increased pressure within the brain can lead to a higher likelihood of the bleeding area expanding, exacerbating neurological damage. Missing the opportune time for treatment may result in complications due to bleeding, increasing the probability of additional medical issues [4–7].

Furthermore, the analysis of CT images requires technical expertise, experience, and knowledge. Recent studies worldwide have reported additional challenges, including (1) a fourfold increase in the workload of radiologists from 2006 to 2020 [8]; (2) the potential decrease in the accuracy of CT image analysis due to the increased workload [9]; (3) a

minimum of 30 min required by radiologists to write an interpretation report after analyzing CT images [10]. Therefore, there is a growing effort in research for the image captioning-based automation of medical image analysis to assist physicians by reducing the time required for radiologists to write interpretation reports and streamlining the diagnostic and treatment processes [11–17].

Studies have used MIMIC-CXR [18], Open-I [19], MS-COCO [20], and ImageCLEF [21] datasets, which are publicly available training datasets for various types of medical images, as well as chest X-ray images. Reference [11] designs three encoders to extract the following feature vectors: (1) visual feature vector (using VGG-16 [22]): vector representation for medical images, (2) semantic feature vector (using VGG-16): vector representation for the classification results and information about the imaging method of medical images, (3) caption vector (using NLTK [23]): vector representation for captions about medical images. These encoder vectors are concatenated and fed into a decoder, which is LSTM (long short-term memory [24]), to generate texts through a beam search method. Reference [12] developed an encoder–decoder architecture using SAT (Show Attend and Tell [25]), a caption generation model for medical images, and GPT-3 [26] as the encoder and decoder to perform text generation for chest X-ray images. Firstly, the SAT encoder generates text for medical images. Then, the GPT-3 decoder is pre-trained with the MIMIC-CXR dataset and fine-tuned with the text of the SAT. Reference [13] modified the CNN encoder of SAT to ResNet-101 and generated texts for medical images through this change. Reference [14] used a pre-trained ResNet34 [27] to represent feature vectors for medical images and applied the MLC (multi-label classification) method to predict the most relevant words from the text. This involves selecting the top-ranked words from the classified words and generating the final caption by arranging them according to statistical rules.

In Reference [15], the encoder is composed of an ensemble learning model using various pre-trained CNN-based models and k-NN (k = 1) [28], while the decoder employs GPT-2 [29]. The authors experimented with various combinations of pre-trained CNN-based models to design a combined encoder as follows: DenseNet [30], InceptionV3 [31], InceptionResNetV2 [32], and Xception [33] (each model is ensembled with 1-NN). At this time, they generated text for medical images using the Voting ensemble learning method. Reference [16] generated texts for fetal ultrasound videos. In this case, the encoder uses the VGG-16 CNN model to represent feature vectors for each frame, and a gaze-assisted model is employed to extract gaze maps for each frame. Then, residual connections are performed for the extracted feature vectors and gaze maps, which are then passed to the decoder, a convolutional LSTM, to generate text.

Reference [17] proposes a 3D CT scan captioning model with an encoder–decoder structure, where the encoder is a 3D CNN model and the decoder is a distilGPT-2 language model. The model is trained end-to-end using an encoder–decoder strategy to generate medical interpretation texts for 3D ICH CT images. Among the proposed models, the one that utilizes an EfficientNet-B5 encoder converted to 3D CNN and employs the beam search text generation strategy achieves an average BLEU score of 0.35. But, the 3D CNN typically used for video captioning considers both spatial and temporal dimensions, which leads to high computational costs and an increased number of model parameters, potentially increasing the risk of overfitting [34]. On the other hand, 2D CNN considers only the spatial features of the image, resulting in lower computational costs and a reduced number of parameters, decreasing the risk of overfitting. Moreover, it excels at extracting structural features from CT images.

Therefore, most image captioning research for medical images has been carried out targeting single medical images (2D images), and it is challenging to find studies focusing on sequentially appearing medical images (3D images) like videos or CT scans (3D CT images). Therefore, there is a need for automated text generation technology capable of processing 3D ICH CT images to improve the diagnostic efficiency and accuracy of ICH. However, specialized models for handling image and text data have the following limitations. CNN models excel in image classification and feature extraction but lack the ability to consider context and sequence when generating sentences. On the other hand,

sequence models like GPT-2 are suitable for sequential data like text or time series but struggle to handle spatial information found in images. Therefore, integrating CNN models which excel in feature extraction with sequence models which are suitable for sequential data may be effective in CT image captioning task.

In this paper, we propose a method to alleviate the burden of analyzing and interpreting CT images for medical professionals by utilizing pre-trained CNN-based classifiers and the language model GPT-2 on 2D images. The goal is to reduce the time required for analysis and report writing, focusing on automatically generating text for sequential brain CT images. First, we make CNN-based classifiers using 2D CNN which has strengths in extracting visual features. The CNN-based classifier undergoes fine-tuning using the Kaggle dataset [35] for ICH multi-classification based on brain CT images. In this case, the utilized pre-trained 2D CNN models were ResNet-50V2 [27], DenseNet-121 [30], VGG-16, and VGG-19 [22]. ResNet and DenseNet can be expected to achieve accurate classification performance on CT images by learning fine features and detailed information through residual connections and dense connections [36]. Additionally, VGG, with its simple yet effective structure, has the advantage of learning various features in CT image classification [37].

Then, the fine-tuned CNN-based classifier serves as the encoder for feature extraction and passes the feature matrix to GPT-2 as follows: it extracts feature vectors from frame-level images of 3D ICH CT images and integrates these vectors into a single matrix (e.g., token embedding). This matrix is then transmitted along with the corresponding text to train GPT-2 to generate text for 3D ICH CT images. By this, the proposed model can perform like an integrated model with strengths of CNN models and sequence models. Finally, we performed a performance evaluation on the proposed model using N-gram-based metrics (BLEU [38], METEOR [39], ROUGE [40], and CIDEr [41]) and embedding-based metrics (skip-thought [42], embedding average, vector extrema [43], and greedy matching [44]) to compare the generated text from test images with reference text. Also, we assessed the drawbacks of the two metrics through the secured BERT score [45].

This paper is structured as follows: Section 2 presents the data collection and preprocessing methods for training the fine-tuned CNN classifier and for collecting and preprocessing data for CT image captioning. In Section 3, a detailed explanation of the structures of the CNN classifier and GPT-2, along with hyperparameters, is provided. Section 4 describes the evaluation metrics used and discusses the performance of the CNN classifier along with the results of the evaluation metrics. Section 5 concludes the work by detailing the results and findings of this paper.

2. Dataset

2.1. Fine-Training for Classifier

While we have sequential brain CT images and corresponding text data, there is a shortage of data to pre-train the CNN classification model from brain CT images. Therefore, we performed fine-tuning using the ICH CT images and the corresponding ICH multi-classification dataset available on Kaggle.

The brain CT images are in DICOM format with meta-information, as shown in Figure 1, and the labels indicating the presence of ICH subtypes for each given image are recorded in CSV format. The subtypes of ICH include the following: intraparenchymal, intraventricular, subarachnoid, subdural, and epidural. The ICH CT images are in DICOM format, as illustrated in Figure 1, with metadata included. Labels indicating the presence of ICH subtypes for a given image are recorded in CSV format. The subtypes of ICH include: intraparenchymal, intraventricular, subarachnoid, subdural, and epidural.

Using the metadata from Figure 1, pixel-formatted images are adjusted for brightness and contrast by windowing [46] with window center, window width, intercept, and slope data, emphasizing ICH more clearly. Subsequently, the images are resized to 224 $\times$ 224 (width $\times$ height) dimensions, and then formatted into PNG. The labels are modified, with normal brain CT images assigned 0, and ICH subtypes of brain CT images assigned 1. Due to a significant difference in the number of normal brain CT images (640k) and images

with the presence of ICH (107k), there is a potential for bias in the training data. Therefore, we randomly selected 150k normal brain CT images and trained the model alongside images containing ICH. Out of the total 257,932 images, 145,807 were used for training, 48,362 for validation, and finally, 64,483 for assessing the model's accuracy. This approach allows us to train the model without bias in the training data and validate and evaluate its performance effectively.

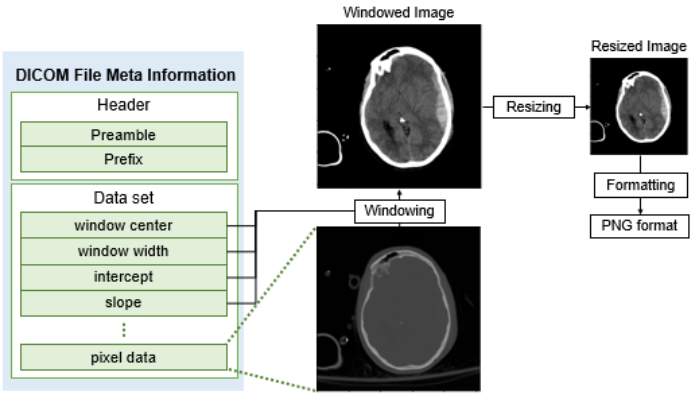

Figure 1. The preprocessing process for DICOM images containing meta-information. The size of CT images is 512 × 512, with a window center of 47, window width of 80, intercept of −1024, and slope of 1.0 applied for windowing. The images were resized to 224 × 224.

2.2. Image Captioning

We used CT scans and corresponding text for a total of 10,368 ICH patients from Chuncheon Sacred Heart Hospital (https://chuncheon.hallym.or.kr/, accessed on 14 December 2023) and Hallym University Sacred Heart Hospital (https://hallym.hallym.or.kr/, accessed on 14 December 2023) in South Korea. These data are also in DICOM format and processed in the same way as Kaggle data.

Text data, when generated, have the potential risk of violating an individual's privacy and breaching medical confidentiality, leading to the possibility of leakage of patient information. Therefore, in this study, all sensitive information, including personal details, was removed in advance, and the following preprocessing steps were undertaken: removal of special characters, conversion to lowercase, elimination of non-English characters, etc. Out of the 10,368 CT scans, 9330 were used for training, 519 for testing, and 518 for validation during the training process to assess the model's performance.

3. Methodology

3.1. Pre-Trained CNN Based Classifier

The automatic text generation model for ICH CT images proposed in this study is depicted in Figure 2. Firstly, pre-trained CNN models are fine-tuned for binary classification tasks, as shown in Figure 3.

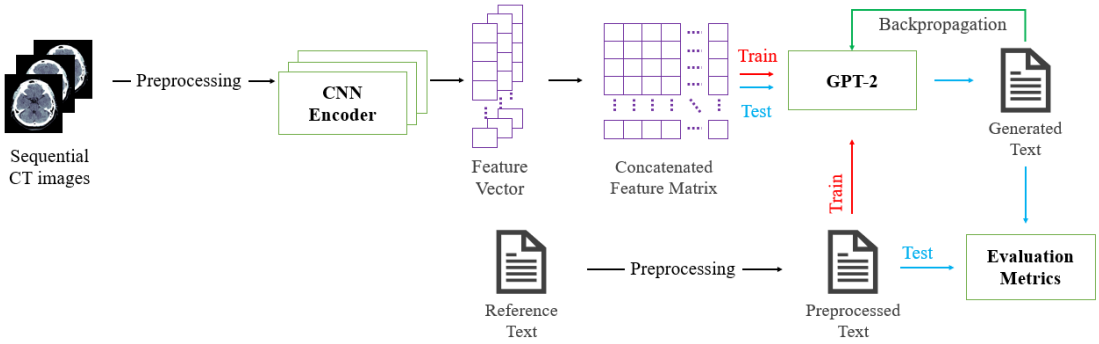

Figure 2. The overall flowchart of the interpretation text generation model using CNN encoder and GPT-2.

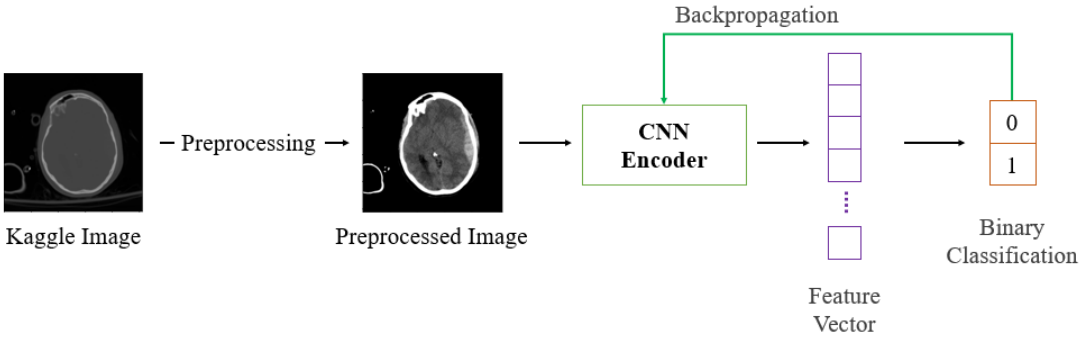

Figure 3. Fine-tuning the pre-trained CNN.

The pre-trained CNN has an input layer of size 224 × 224, and a fully-connected layer with a size of 1024 is added for fine-tuning, just before the output layer, to ensure the extraction of features of the same size before the output layer. Following this, it learns the presence of ICH from preprocessed Kaggle image data. During training for text generation, we use the architecture excluding the last layer responsible for classification, referred to as the CNN encoder. In this context, we compare and analyze the performance using the following four fine-tuned CNNs to determine a suitable encoder architecture:

- ResNet-50V2 is a lightweight and efficient model compared to its predecessor, ResNet-50. It utilizes residual connections to improve the learning process by adding skip connections, which add the feature maps extracted from the previous layer to the input of the next layer. This increases the depth of the network, showcasing improved performance during the training process. The architecture of ResNet-50V2, depicted in Figure 4, incorporates pretrained weights that enhance the performance in training with low-resource data, making it adept at feature extraction for untrained data such as medical images. The hyperparameters used in ResNet-50V2 are as follows: the initial layer consists of a 2D convolution layer with a 7 × 7 kernel size and 64 filters, followed by batch normalization and ReLU activation functions. Subsequently, a 3 × 3 max-pooling layer with a stride of 2 is added. The following layers include four residual blocks. The first block has 64 filters and a stride of 2, the second block has 128 filters and a stride of 2, the third block has 256 filters and a stride of 2, and the fourth block has 512 filters with a stride of 1.

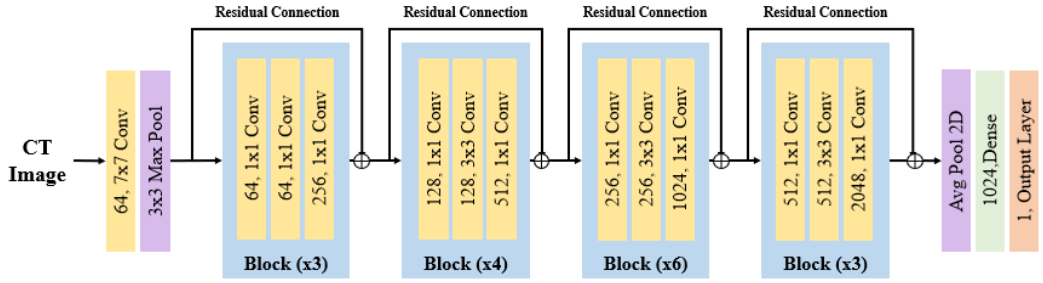

Figure 4. ResNet-50V2 architecture, the value before each block is connected to the value after the block through a simple addition in residual connections.

- DenseNet-121 is structured with dense blocks and transition layers, utilizing a sequence of convolution layers and skip connections. While ResNet forms a pathway by connecting the immediate layer with an element-wise addition, DenseNet densely connects layers as it goes deeper, employing channel-wise concatenation. The dense block forms dense connections between internal layers, enhancing feature extraction and the ability to reuse information. The transition layer adjusts the size of feature maps, maintaining the efficiency of the model. In addition, through the dense connection structure, features between layers accumulate, enabling the extraction of optimized

features for subtle changes or patterns related to ICH. The architecture of DenseNet-121 is depicted in Figure 5, and the hyperparameters used are as follows: the first layer uses a 7×7 kernel size with 64 filters, along with batch normalization and ReLU activation functions. Furthermore, the transition layer consists of a 1×1 convolution layer and a 2×2 average pooling layer.

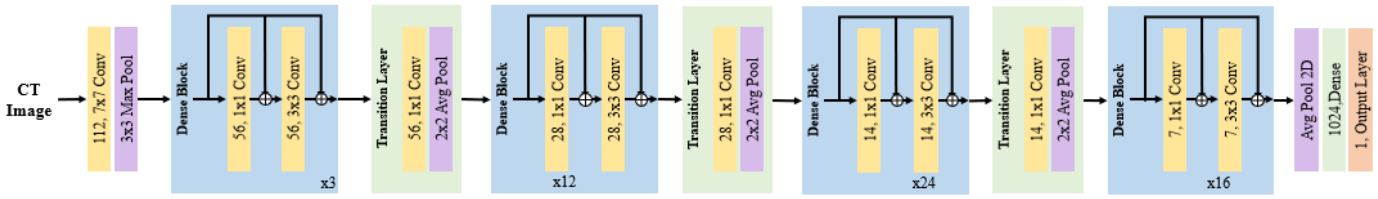

Figure 5. DenseNet-121 architecture, all convolutional layers within the dense block are densely connected by concatenation until they input into the transition layer.

- VGG-16 consists of 16 layers, comprising 13 convolution layers and 3 fully connected layers. The distinctive feature of VGG-16 is its deep structure and the use of small filter sizes. VGG-16 is a simple yet powerful model primarily employed in computer vision tasks, capable of extracting rich features due to its very deep network architecture. This feature extraction ability enables the detection and extraction of various features of ICH, deriving relevant information. The architecture of VGG-16 is depicted in Figure 6, and the hyperparameters used are as follows: all convolution layers have a 3×3 kernel size with ReLU activation functions applied. Max pooling layers reduce the size of feature maps using a 2×2 kernel with a stride of 2. The fully connected layer consists of three dense layers with ReLU activation functions.

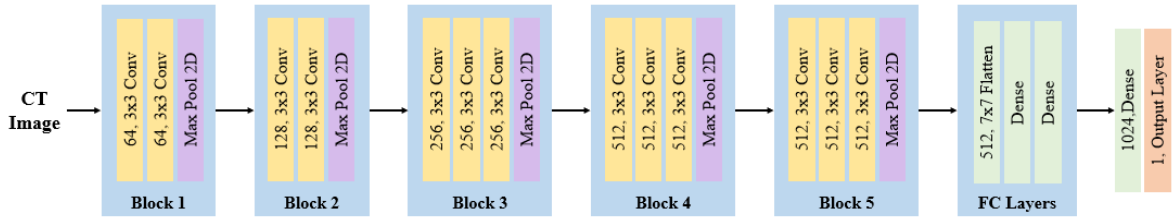

Figure 6. VGG-16 architecture.

- VGG-19 is a model with a structure similar to VGG-16, but it has a more complex architecture with additional layers, allowing it to learn more intricate features. It consists of 19 layers, with an additional convolution layer in each of the third, fourth, and fifth blocks compared to VGG-16. The inclusion of these three extra convolution layers in VGG-19 enables it to learn more complex features of ICH and recognize a greater variety of detailed patterns. The architecture of VGG-19 is illustrated in Figure 7, and the hyperparameters used are as follows: it comprises 16 convolution layers with 3×3 filter sizes and 3 fully connected layers.

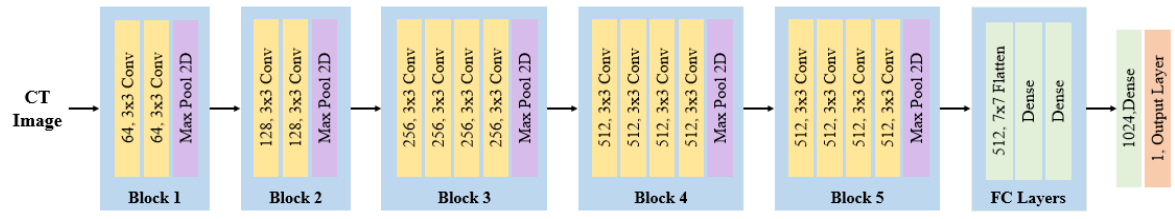

Figure 7. VGG-19 architecture.

Following this, frame-level feature vectors of 3D ICH CT images are extracted from the CNN encoder with the output layer excluded for feature extraction. These vectors are then

merged into a feature matrix. The merged feature matrix is adjusted to a size of 74 × 1024, aligning with the largest number of frames among the 3D ICH CT images. This structure is consistent with token embeddings generated from text. GPT-2 is trained to generate text for consecutive CT images using the feature matrix and corresponding text. Finally, the trained GPT-2 is used to predict and generate text for test 3D ICH CT images. The generated text is evaluated by comparing it with the reference text for the test CT images.

3.2. GPT-2

In this study, GPT-2 was utilized due to computer resource constraints. GPT-2 is the second model in the widely used GPT series in the field of natural language processing. This model is based on the transformer architecture, using the attention mechanism to capture relationships between words in a sentence. This enables it to effectively learn correlations among words that are farther apart in a sentence, leveraging the advantages of RNN and LSTM-based models while enhancing computational efficiency through parallel processing. By training GPT-2 on the features of 3D ICH CT images and language information, it could enhance the understanding of ICH for medical professionals and patients. Also, it might assist in predicting the likelihood of ICH, proposing hypotheses considering symptoms and related factors, thus aiding in further examinations or evaluations.

Figure 8 illustrates the process of training GPT-2 using the extracted and merged feature matrix from the CNN encoder and the corresponding text. The input to GPT-2 consists of embedding vectors for tokens that include positional encoding. This vector sequence is transformed into Query (Q) vectors representing current positional information, Key (K) vectors measuring relationships between different positions, and Value (V) vectors containing actual information. These vectors generate attention scores through masked multi-head attention, applying masking with a Lookahead mask to conceal information beyond the current position. The generated attention scores pass through the SoftMax function to calculate attention weights. After passing through the input embedding and residual connection, layer normalization is performed. Subsequently, the generated vectors are utilized as Q vectors in multi-head attention, and the merged feature matrix created through the CNN encoder serves as K and V vectors (i.e., cross-attention). Q, K, and V vectors are transformed into vectors with the same dimension, which is the embedding dimension divided by the number of transformer heads. The input vectors then undergo the calculation of attention weights, followed by residual connection and layer normalization. Finally, they pass through the position-wise feed-forward neural network, and residual connection and layer normalization are applied. This entire process is repeated for the number of decoders (n = 16).

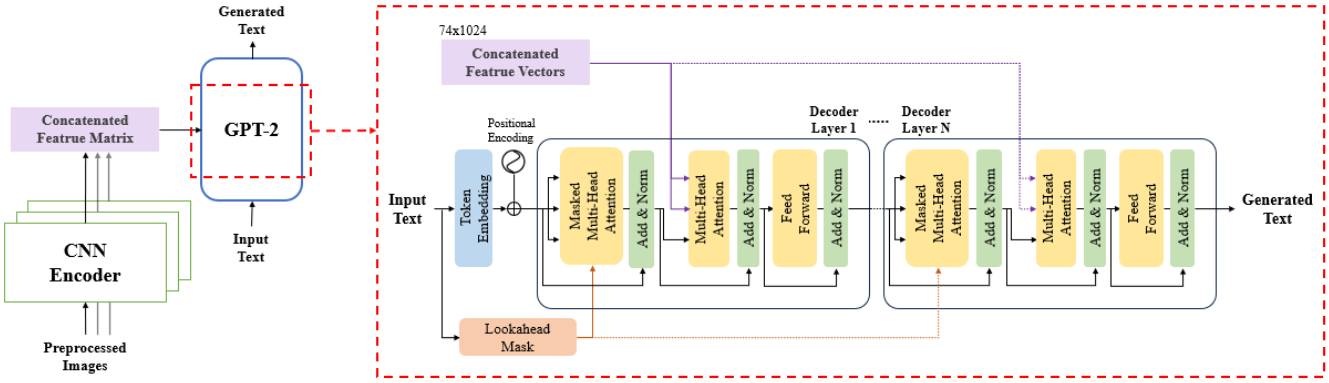

Figure 8. The training process involves using feature matrices extracted from sequential ICH CT images and their corresponding texts to train GPT-2. The number of decoders (N) is 16.

The GPT-2 architecture utilized in this study consists of 6 transformer blocks, with an additional 4 layers added to the existing 12 decoder layers. Each decoder layer employs 6 multi-head attention modules. Finally, the test data undergo automatic text generation

using GPT-2 based on the corresponding feature vectors. The generated text is then evaluated for performance by comparing it with the reference text from the test data, utilizing 8 evaluation metrics and BERT scores. Text generation employs beam search and greedy search methods. Beam search is a sequence-decoding strategy that explores the global optimum by maintaining multiple candidates simultaneously, considering various options to enhance the results. In contrast, greedy search is a sequential decoding strategy that is computationally faster than beam search. It considers only the choice with the highest score at each step, sequentially determining the sequence.

4. Experiments

4.1. Experimental Setup

All models were implemented with TensorFlow in Python3 and were run in an environment with two NVIDIA A5000 GPUs (Nvidia Corporation, Santa Clara, CA, USA). The training took approximately 17.5 h. Inference using test data took approximately 4.5 h.

To ensure that all fine-tuned CNNs undergo training under the same conditions, the following parameters were considered: the loss function is binary cross-entropy and the optimization algorithm is Adam optimizer [47]. The batch size was set to 64, and the learning rate was set to 1×10^{-5}. Training proceeded for 300 epochs, with early stopping configured to halt training if the validation data loss did not decrease for 15 consecutive epochs. The GPT-2 transformer block employed 768 embedding dimensions, and the vocabulary size was 5000, indicating that the model was trained on a dataset containing 5000 unique tokens.

The total number of parameters in GPT-2 was 168M, with a batch size of 16 and a learning rate set to 1×10^{-5}. The loss function was the SparseCategoricalCrossentropy function from Keras, and the optimization algorithm was the Adam optimizer. Training spanned 300 epochs, with early stopping configured to halt training if the validation data loss did not decrease for 20 consecutive epochs. The text generated by GPT-2 was limited to a maximum caption length of 200 words.

4.2. Evaluation Metrics

4.2.1. N-Gram-Based Evaluation Metrics

N-gram is a method of dividing text or sentences into consecutive N tokens to analyze the frequency and order of each unit. Therefore, N-gram-based evaluation metrics assess how well N-gram units in the reference text match those in the generated text. In this paper, the following four metrics were used: BLEU, METEOR, ROUGE_L, and CIDEr.

BLEU [38] is a metric for evaluating the quality of machine-generated sentences. It calculates N-gram precision by comparing the output of a machine translation system with reference translation sentences, combining them with harmonic mean to compute a score. The calculated score measures how similar the predicted sentences are to the reference sentences, providing a method to assess the performance of the translation. METEOR [39] operates similarly to BLEU but maps words in generated sentences to words in reference sentences, evaluating how accurately they follow the order and structure. It considers the quality of sentences by penalizing for incorrect order, considering the meaning and structure of the sentences. ROUGE_L [40] finds the structurally similar longest common subsequence (LCS) between the generated sentence and the reference sentence to measure the similarity between sentences. CIDEr [41] introduces weights for each N-gram to evaluate the match between the generated text and reference sentences. Initially, it sums the weights for all reference sentences, then calculates the average by dividing the total by the number of reference sentences to determine the average matching degree between reference sentences. Subsequently, it computes the average for the matching degree between the generated text and reference sentences, resulting in the final score.

4.2.2. Embedding-Based Evaluation Metrics

Embedding is a technique that maps words or tokens to high-dimensional vectors. It aims to capture semantic relationships between words, ensuring that words with similar meanings are positioned closer in the vector space. Consequently, embedding-based metrics assess the semantic similarity between reference and generated texts. In this paper, the following four metrics were utilized: skip-thought, embedding average, vector extrema, and greedy matching. These metrics evaluate the semantic similarity between the reference and generated texts based on their meanings.

Skip-thought [42] is an LSTM-based language model that generates sentence-level embeddings and is structured as an encoder–decoder architecture. The encoder encodes the given input sentence, and the decoder predicts the next sentence that follows the input sentence. The similarity between the input sentence and the predicted next sentence is measured. Embedding average is a method of representing a sentence's embedding vector by taking the average of the embedding vectors for each word in the sentence. This approach retains information about the words within the sentence and can encapsulate the meaning and grammatical structure of the sentence. Thus, embedding average measures the similarity of embedding vectors between two sentences. Equation (1) illustrates the evaluation method using embedding average, where $\bar{e}_S$ represents the average of word embeddings for each token in sentence S.

$$\begin{gathered} \bar{e}_S = \frac{\sum_{w \in S} e_w}{\left|\sum_{w' \in S} e_{w'}\right|} \\ \text{Embedding average} := cos\_sim(\bar{e}_S, \bar{e}_{S'}) \end{gathered} \tag{1}$$

Vector extrema [43] selects the value that is furthest from 0 among the maximum and minimum values in the embedding vector to generate a representative value for a sentence. Using cosine similarity, this generated value for the reference sentence and the predicted sentence is used to measure their similarity. In greedy matching [44], given a predicted sentence and a reference sentence, the word embeddings of each word in the predicted sentence calculate the maximum similarity score among all word embeddings in the reference sentence. Conversely, the word embeddings of each word in each reference sentence calculate the maximum similarity score among all word embeddings in the predicted sentence. The two values are then added, averaged, and used to measure the similarity between the two sentences.

4.2.3. BERT Score

BERT score [45] is a sentence similarity metric that effectively combines the advantages of N-gram-based metrics and embedding-based metrics using BERT [48]. The BERT score involves inputting both the reference and predicted sentences into the BERT model to obtain contextual embeddings. These embeddings are then used to create a similarity matrix using cosine similarity for each token pair. The generated matrix calculates precision through column-wise max pooling and recall through row-wise max pooling. F1 is computed from the calculated recall and precision.

4.3. Experiment Results

Table 1 presents the performance evaluation results of the fine-tuned CNN classifier on the publicly available Kaggle data. VGG-19 achieved the highest precision at 0.94, while VGG-16 showed the highest recall at 0.89. All fine-tuned CNN classifiers demonstrated a high F1 score and accuracy of around 90%, indicating successful training. Next, we evaluated the performance of the four fine-tuned CNN classifiers and the GPT-2-based image captioning model on generated text. We assessed performance using N-gram-based evaluation metrics and embedding-based evaluation metrics, as shown in Tables 2 and 3.

Table 1. Performance comparison with four fine-tuned CNN classifiers.

Classifiers	Precision	Recall	F1-Score	Acc
ResNet-50V2	0.93	0.87	**0.90**	**0.92**
DenseNet-121	0.93	0.86	0.89	0.91
VGG-16	0.92	**0.89**	**0.90**	**0.92**
VGG-19	**0.94**	0.86	**0.90**	**0.92**

Table 2 shows the performance evaluation results for N-gram-based metrics (BLEU, METEOR, ROUGE_L, CIDEr). DenseNet-121 exhibited relatively high scores in BLEU and METEOR. While ResNet-50V2 had lower BLEU scores compared to DenseNet-121, the difference was not substantial, and it showed high scores in ROUGE_L and CIDEr. Despite VGG-16 and VGG-19 demonstrating high classification accuracy, as seen in Table 1, they exhibited lower performance in N-gram-based evaluations.

Table 2. Evaluation scores based on the N-gram metrics for the final model. (B1 to B4: BLEU, B@4: average of B1 to B4, M: METEOR, R_L: ROUGE_L, C: CIDEr, B: beam search (n = 3), G: greedy search).

Models (With GPT-2)		B1	B2	B3	B4	B@4	M	R_L	C
ResNet-50V2	B	0.27	0.19	0.16	0.13	0.18	**0.14**	**0.30**	**0.38**
	G	0.25	0.19	0.15	0.13	0.18	0.13	**0.30**	**0.36**
DenseNet-121	B	**0.28**	**0.21**	**0.17**	**0.14**	**0.20**	**0.14**	0.28	0.25
	G	**0.28**	**0.21**	**0.17**	**0.14**	**0.20**	**0.14**	0.29	0.27
VGG-16	B	0.20	0.14	0.12	0.10	0.14	0.10	0.21	0.18
	G	0.20	0.15	0.12	0.10	0.13	0.09	0.20	0.16
VGG-19	B	0.21	0.16	0.13	0.11	0.12	0.10	0.23	0.16
	G	0.21	0.16	0.13	0.10	0.12	0.10	0.23	0.17

Table 3 presents another set of metrics, including skip-thought, embedding average, vector extrema, and greedy matching score, which calculate the cosine similarity between the embeddings of predicted and reference sentences. The model utilizing DenseNet-121 encoder achieved the highest scores in each embedding metric: 0.54 in skip-thought, 0.71 in embedding average, 0.46 in vector extrema, and 0.63 in greedy matching. It consistently displayed high scores in N-gram-based metrics as well. This indicates that DenseNet-121's feature representation benefits from the densely connected nature of its layers, enabling it to capture complex patterns more effectively. The deeper layers contribute to the accumulation and reuse of more features, resulting in the observed high-performance outcomes [49].

Table 3. The embedding-based metric evaluation scores of the final model. (ST: skip-thought, EA: embedding average, VE: vector extrema, GM: greedy matching, B: beam search (n = 3), G: greedy search).

Models (+GPT-2)		ST	EA	VE	GM
ResNet-50V2	B	0.51	0.69	0.44	**0.63**
	G	0.51	0.69	0.44	**0.63**
DenseNet-121	B	**0.54**	**0.71**	**0.46**	**0.63**
	G	**0.54**	**0.71**	**0.45**	**0.63**
VGG-16	B	0.51	0.66	0.42	0.59
	G	0.51	0.66	0.42	0.60
VGG-19	B	0.50	0.66	0.41	0.59
	G	0.51	0.67	0.44	0.59

N-gram-based evaluation metrics are effective in capturing local patterns but have limitations in handling long sentences or diverse vocabularies, posing challenges in terms of vocabulary diversity and context representation. Moreover, due to a fixed number of previous tokens, they might struggle with capturing long-term dependencies, considering only partial sequence information, and potentially limiting the model's understanding and prediction of context.

Embedding-based evaluation metrics measure performance on specific tasks but do not provide insights into how well a model performs in different tasks or domains. This limitation hinders the evaluation of a model's generalization ability. Embeddings are learned automatically based on the training data, and if the data are biased or contain limited information, the learned embeddings may reflect this bias, leading to a decrease in model performance when applied to new data or domains.

BERT score is an evaluation metric designed to leverage the strengths of both metrics while addressing their shortcomings. Table 4 presents the evaluation scores using BERT score. The employed BERT model is PubMedBERT [50], and the results show high performance across precision, recall, and F1 scores, all reaching 80%, in evaluating sentence similarity based on contextual information.

Table 4. BERT score of the final model utilizing PubMedBERT.

PubMedBERT	Precision	Recall	F1-Score
ResNet50V2 + GPT2	**0.83**	**0.81**	**0.82**
DenseNet121 + GPT2	0.80	0.80	0.80
VGG16 + GPT2	0.82	0.80	0.81
VGG19 + GPT2	0.81	0.80	0.80

Table 5 displays the text generated from 3D ICH CT images by the final model, combining each CNN encoder and GPT-2. In the generated sentences, words unrelated to the reference sentences are marked in red, words semantically similar to ICH are marked in blue, and accurately generated words are marked in purple. When comparing the reference sentences with the generated ones, ResNet-50V2 failed to produce sentences mentioning the incorrect location and ICH, but accurately generated key information about subdual hematoma (SDH) and brain herniation. Both DenseNet-121 and VGG-16 did not generate sentences related to ICH but produced text related to lacunar infarctions and SAH, resembling ICH in the CT images. VGG-19 generated sentences related to ICH but appeared to confuse SDH with ICH and generated ICH as lacunar infarctions.

Table 5. Text generated from the final model (red: words generated differently from the ground truth, blue: words semantically similar to ICH, purple: words generated identical to the ground truth).

Ground Truth	ResNet50V2 + GPT2	DenseNet121 + GPT2	VGG16 + GPT2	VGG19 + GPT2
SDH right fronto temporo parietal ICH right temporo parietal brain herniation, otherwise no demonstrable abnormal finding.	**SDH left** fronto parietal with **brain herniation**, otherwise no demonstrable abnormal finding.	**SDH right fronto temporo parietal** and right **tentorium small vessel disease with lacunar infarctions**, otherwise no demonstrable abnormal finding.	**SDH right fronto temporo parietal** and falx **SDH**, otherwise no demonstrable abnormal finding.	**SDH in left basal ganglia small vessel disease with lacunar infarctions**, otherwise no demonstrable abnormal finding.

4.4. Discussion

Unlike the classification performance shown in Table 1, VGG-16 and VGG-19 exhibit lower scores in the N-gram-based metric evaluations of Table 2. On the other hand, ResNet-50V2 and DenseNet-121 generate text with higher scores compared to VGG-16 and VGG-19. This difference highlights the correlation between the model's architecture

and the complexity of the task it aims to perform. ResNet-50V2 and DenseNet-121 can effectively capture complex patterns and features through residual and dense connections. Moreover, they can efficiently learn and generalize complex features with fewer parameters. In text generation tasks, the importance of contextual coherence and semantic consistency is highly significant. In this context, it can be observed that ResNet-50V2 and DenseNet-121 possess architectures that consider features while simultaneously extracting and utilizing consistent features. In contrast, the lower performance of VGG in N-gram-based metric evaluations suggests a lack of contextual understanding in the model [51,52]. However, it can be observed that these scores do not differ significantly from the results of generating text through a single medical image [53].

Similar to our study, Reference [17] trained the encoder and decoder together, requiring both normal and ICH patients to go through the text generation process. In contrast, our study allows for the pre-confirmation of normal patients in the encoder section, enabling faster patient classification. Furthermore, when generating text using only data from ICH patients, the performance in terms of B@4, METEOR, and ROGUE-L is very similar to the performance when using DenseNet-121 as the encoder. This suggests that comparable performance can be expected with fewer resources, indicating better efficiency.

The fine-tuned CNN learned only the presence or absence of ICH. However, as seen in Table 4, it can be observed that the model generates words related to diseases other than ICH. This may be attributed to the fact that the fine-tuned CNN has learned both normal and abnormal brains and has discriminated features beyond ICH. And the scores obtained through the BERT score showed a consistently high accuracy of 80%, unlike other evaluation metrics. This suggests that PubMedBERT, pre-trained on extensive textual data in the medical field, exhibited outstanding performance in generating text.

The approach we proposed has the advantage of leveraging features from sequential images and textual information, but it also has the following limitations: (1) training and execution are more complex; (2) insufficient data may compromise the model's performance; (3) GPT-2 relies on the feature vectors from the CNN encoder during training, making the representational capability of the CNN encoder crucial. However, ongoing data collection and technological advancements can enhance the performance of both the CNN encoder and GPT-2. Moreover, due to the absence of a gold standard dataset, it is challenging to objectively evaluate superiority. Given the medical context, the most significant concern in text generation is the Hallucination problem, which should be evaluated from a clinical perspective.

5. Conclusions

In this paper, we propose a method for automatically generating text from sequential brain CT images using a pre-trained CNN and the language model GPT-2, focusing on 2D images. We fine-tuned four types of pre-trained CNN classifiers, combined the CNN encoder up to the layer extracting features from the fine-tuned CNN classifier with GPT-2, and explored suitable models for the task.

Firstly, the fine-tuned CNN classifiers, trained and tested on publicly available datasets, all exhibited high accuracy, surpassing 90%. Furthermore, regarding the generation of text for continuous brain CT images combined with GPT-2, ResNet and DenseNet exhibited excellent performance in terms of both similarity to the actual answers (N-gram-based evaluation metric) and semantic aspects (embedding-based evaluation metric). Additionally, utilizing PubMedBERT for BERT score, the models achieved outstanding results in generating sentences relevant to the medical field.

In the future, we will augment the objective evaluation of our proposed method by conducting a direct assessment from radiologists, evaluating aspects such as the model's stability, consistency, and clinical relevance. Additionally, we will explore the following methods to improve model performance: (1) the multimodal method that integrates data other than CT images; (2) utilization of advanced models such as medical imaging-specific CNN and GPT-3 to determine the structural characteristics of ICH and the linguistic

characteristics of the text more accurately. Through these efforts, we anticipate that our proposed method could assist radiologists in swiftly identifying various conditions, including intracranial hemorrhage (ICH), and making informed treatment decisions.

Author Contributions: Conceptualization, Y.-S.K. and C.K.; methodology, J.-W.K. and B.-D.O.; formal analysis, J.-W.K. and B.-D.O.; resources, Y.-S.K.; data curation, C.K.; writing—original draft preparation, J.-W.K.; writing—review and editing, Y.-S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2022R1A5A8019303), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) [No. 2021-0-02068, Artificial Intelligence Innovation Hub (Artificial Intelligence Institute, Seoul National University)], and Korea Health Technology R&D Project through the Korean Health Industry Development Institute (KHIDI) grant funded by the Ministry of Health and Welfare, Republic of Korea (grant number: HR21C0198).

Institutional Review Board Statement: This study was performed in accordance with the Declaration of Helsinki, and it was approved by the Institutional Review Board at Chuncheon Sacred Heart Hospital (IRB No. 2021-10-012).

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [https://www.kaggle.com/competitions/rsna-intracranial-hemorrhage-detection]. The data from Chuncheon Sacred Heart Hospital and Hallym University Sacred Heart Hospital in this study are available on request from the corresponding author due to privacy and legal restrictions.

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

References

1. Rindler, R.S.; Allen, J.W.; Barrow, J.W.; Pradilla, G.; Barrow, D.L. Neuroimaging of Intracerebral Hemorrhage. *Neurosurgery* **2020**, *86*, E414–E423. [CrossRef] [PubMed]
2. Ginat, D.T. Analysis of head CT scans flagged by deep learning software for acute intracranial hemorrhage. *Neuroradiology* **2020**, *62*, 335–340. [CrossRef] [PubMed]
3. Ibrahim, A.; Arifianto, M.R.; Al Fauzi, A. Minimally Invasive Neuroendoscopic Surgery for Spontaneous Intracerebral Hemorrhage: A Review of the Rationale and Associated Complications. *Complic. Neurosurg.* **2023**, *130*, 103–108.
4. Ovenden, C.D.; Hewitt, J.; Kovoor, J.; Gupta, A.; Edwards, S.; Abou-Hamden, A.; Kleinig, T. Time to hospital presentation following intracerebral haemorrhage: Proportion of patients presenting within eight hours and factors associated with delayed presentation. *J. Stroke Cerebrovasc. Dis.* **2022**, *31*, 106758. [CrossRef]
5. Mohammed, B.A.; Senan, E.M.; Al-Mekhlafi, Z.G.; Rassem, T.H.; Makbol, N.M.; Alanazi, A.A.; Almurayziq, T.S.; Ghaleb, F.A.; Sallam, A.A. Multi-Method Diagnosis of CT Images for Rapid Detection of Intracranial Hemorrhages Based on Deep and Hybrid Learning. *Electronics* **2022**, *11*, 2460. [CrossRef]
6. Chandrabhatla, A.S.; Kuo, E.A.; Sokolowski, J.D.; Kellogg, R.T.; Park, M.; Mastorakos, P. Artificial Intelligence and Machine Learning in the Diagnosis and Management of Stroke: A Narrative Review of United States Food and Drug Administration-Approved Technologies. *J. Clin. Med.* **2023**, *12*, 3755. [CrossRef] [PubMed]
7. Cordonnier, C.; Demchuk, A.; Ziai, W.; Anderson, C.S. Intracerebral haemorrhage: Current approaches to acute management. *Lancet* **2018**, *392*, 1257–1268. [CrossRef] [PubMed]
8. Bruls, R.; Kwee, R. Workload for radiologists during on-call hours: Dramatic increase in the past 15 years. *Insights Imaging* **2020**, *11*, 121. [CrossRef]
9. Alexander, R.; Waite, S.; Bruno, M.A.; Krupinski, E.A.; Berlin, L.; Macknik, S.; Martinez-Conde, S. Mandating limits on workload, duty, and speed in radiology. *Radiology* **2022**, *304*, 274–282. [CrossRef]
10. Ayesha, H.; Iqbal, S.; Tariq, M.; Abrar, M.; Sanaullah, M.; Abbas, I.; Rehman, A.; Niazi, M.F.K.; Hussain, S. Automatic medical image interpretation: State of the art and future directions. *Pattern Recognit.* **2021**, *114*, 107856. [CrossRef]
11. Beddiar, D.R.; Oussalah, M.; Seppänen, T.; Jennane, R. ACapMed: Automatic Captioning for Medical Imaging. *Appl. Sci.* **2022**, *12*, 11092. [CrossRef]
12. Selivanov, A.; Rogov, O.Y.; Chesakov, D.; Shelmanov, A.; Fedulova, I.; Dylov, D.V. Medical image captioning via generative pretrained transformers. *Sci. Rep.* **2023**, *13*, 4171. [CrossRef]
13. Tsuneda, R.; Asakawa, T.; Aono, M. Kdelab at ImageCLEF 2021: Medical Caption Prediction with Effective Data Pre-processing and Deep Learning. In Proceedings of the CLEF (Working Notes), Bucharest, Romania, 21–24 September 2021; pp. 1365–1374.

14. Castro, V.; Pino, P.; Parra, D.; Lobel, H. PUC Chile team at Caption Prediction: ResNet visual encoding and caption classification with Parametric ReLU. In Proceedings of the CLEF (Working Notes), Bucharest, Romania, 21–24 September 2021; pp. 1174–1183.
15. Charalampakos, F.; Karatzas, V.; Kougia, V.; Pavlopoulos, J.; Androutsopoulos, I. AUEB NLP Group at ImageCLEFmed Caption Tasks 2021. In Proceedings of the CLEF (Working Notes), Bucharest, Romania, 21–24 September 2021; pp. 1184–1200.
16. Alsharid, M.; Cai, Y.; Sharma, H.; Drukker, L.; Papageorghiou, A.T.; Noble, J.A. Gaze-assisted automatic captioning of fetal ultrasound videos using three-way multi-modal deep neural networks. *Med. Image Anal.* **2022**, *82*, 102630. [CrossRef] [PubMed]
17. Kim, G.-Y.; Oh, B.-D.; Kim, C.; Kim, Y.-S. Convolutional Neural Network and Language Model-Based Sequential CT Image Captioning for Intracerebral Hemorrhage. *Appl. Sci.* **2023**, *13*, 9665. [CrossRef]
18. Johnson, A.E.; Pollard, T.J.; Berkowitz, S.J.; Greenbaum, N.R.; Lungren, M.P.; Deng, C.-y.; Mark, R.G.; Horng, S. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. *Sci. Data* **2019**, *6*, 317. [CrossRef] [PubMed]
19. 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]
20. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13 2014, pp. 740–755.
21. Ionescu, B.; Müller, H.; Péteri, R.; Abacha, A.B.; Sarrouti, M.; Demner-Fushman, D.; Hasan, S.A.; Kozlovski, S.; Liauchuk, V.; Cid, Y.D.; et al. Overview of the ImageCLEF 2021: Multimedia Retrieval in Medical, Nature, Internet and Social Media Applications. In Proceedings of the Experimental IR Meets Multilinguality, Multimodality, and Interaction: 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, 21–24 September 2021; Proceedings, 2021, pp. 345–370.
22. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
23. Loper, E.; Bird, S. Nltk: The natural language toolkit. *arXiv* **2002**, arXiv:cs/0205028.
24. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
25. 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, 7–9 July 2015; pp. 2048–2057.
26. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
28. Peterson, L. K-nearest neighbor. *Scholarpedia* **2009**, *4*, 1883. [CrossRef]
29. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
30. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
31. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
32. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
33. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
34. Yu, J.; Yang, B.; Wang, J.; Leader, J.; Wilson, D.; Pu, J. 2D CNN versus 3D CNN for false-positive reduction in lung cancer screening. *J. Med. Imaging* **2020**, *7*, 051202. [CrossRef] [PubMed]
35. Kaggle. Kaggle Competitions: RSNA Intracranial Hemorrhage Detection. Available online: https://www.kaggle.com/competitions/rsna-intracranial-hemorrhage-detection (accessed on 5 December 2023).
36. Zhou, Q.; Zhu, W.; Li, F.; Yuan, M.; Zheng, L.; Liu, X. Transfer learning of the ResNet-18 and DenseNet-121 model used to diagnose intracranial hemorrhage in CT scanning. *Curr. Pharm. Des.* **2022**, *28*, 287–295. [CrossRef] [PubMed]
37. Mahmoud, A.; Awad, N.A.; Alsubaie, N.; Ansarullah, S.I.; Alqahtani, M.S.; Abbas, M.; Usman, M.; Soufiene, B.O.; Saber, A. Advanced Deep Learning Approaches for Accurate Brain Tumor Classification in Medical Imaging. *Symmetry* **2023**, *15*, 571. [CrossRef]
38. 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, 7–12 July 2002; pp. 311–318.
39. Banerjee, S.; Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; pp. 65–72.
40. Lin, C.-Y. Rouge: A package for automatic evaluation of summaries. In Proceedings of the Text Summarization Branches Out, Barcelona, Spain, 25–26 July 2004; pp. 74–81.
41. Vedantam, R.; Lawrence Zitnick, C.; Parikh, D. Cider: Consensus-based image description evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4566–4575.

42. Kiros, R.; Zhu, Y.; Salakhutdinov, R.R.; Zemel, R.; Urtasun, R.; Torralba, A.; Fidler, S. Skip-thought vectors. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28.
43. Forgues, G.; Pineau, J.; Larchevêque, J.-M.; Tremblay, R. Bootstrapping dialog systems with word embeddings. In Proceedings of the Nips, Modern Machine Learning and Natural Language Processing Workshop, Montreal, QC, Canada, 9–11 December 2014; p. 168.
44. Rus, V.; Lintean, M. An optimal assessment of natural language student input using word-to-word similarity metrics. In Proceedings of the Intelligent Tutoring Systems: 11th International Conference, ITS 2012, Chania, Crete, Greece, 14–18 June 2012; Proceedings 11 2012, pp. 675–676.
45. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating text generation with BERT. *arXiv* **2019**, arXiv:1904.09675.
46. Tidwell, A.S. Advanced imaging concepts: A pictorial glossary of CT and MRI technology. *Clin. Tech. Small Anim. Pract.* **1999**, *14*, 65–111. [CrossRef] [PubMed]
47. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
48. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
49. Dai, Y.; Song, Y.; Liu, W.; Bai, W.; Gao, Y.; Dong, X.; Lv, W. Multi-focus image fusion based on convolution neural network for Parkinson's Disease image classification. *Diagnostics* **2021**, *11*, 2379. [CrossRef]
50. Gu, Y.; Tinn, R.; Cheng, H.; Lucas, M.; Usuyama, N.; Liu, X.; Naumann, T.; Gao, J.; Poon, H. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthc.* **2021**, *3*, 1–23. [CrossRef]
51. Al-Malla, M.A.; Jafar, A.; Ghneim, N. Pre-trained CNNs as Feature-Extraction Modules for Image Captioning: An Experimental Study. *ELCVIA Electron. Lett. Comput. Vis. Image Anal.* **2022**, *21*, 1–16. [CrossRef]
52. Staniūtė, R.; Šešok, D. A systematic literature review on image captioning. *Appl. Sci.* **2019**, *9*, 2024. [CrossRef]
53. Park, H.; Kim, K.; Park, S.; Choi, J. Medical image captioning model to convey more details: Methodological comparison of feature difference generation. *IEEE Access* **2021**, *9*, 150560–150568. [CrossRef]