Enhancing image captioning with depth information using a Transformer-based framework

Anonymous authors
Paper under double-blind review

Abstract

Captioning images is a challenging scene-understanding task that connects computer vision and natural language processing. Although image captioning models have been capable of producing excellent descriptions, a significant advancement in this area focuses primarily on generating a single sentence for 2D images. In this paper, we investigate whether combining depth information with RGB images for the captioning task can help to generate better captions. For this purpose, we propose a Transformer-based encoder-decoder framework for generating a multi-sentence description of a 3D scene. The RGB image and its corresponding depth map are provided as inputs to our framework, which combines them to produce a better understanding of the input scene. We explore different fusion approaches to fuse RGB and depth images. We first study the NYU-v2 dataset and found inconsistent labeling that prevents the benefit of using the depth information to enhance the captioning task. The results were even worse than when using RGB images only. As a result, we propose a cleaned version of the NYU-v2 dataset that is more consistent and informative. Extensive experiments on the cleaned dataset show that the proposed framework can effectively benefit from depth information and generate better captions. Code, pre-trained models, and the cleaned version of the NYU-v2 dataset will be made publically available.

1 Introduction

In recent years, there has been a significant research effort in the integration of vision and language to bridge the gap between the computer vision and natural language processing domains. Vision-Language tasks include processing a picture and a text that is related to it simultaneously. Several vision-language tasks have achieved significant progress, including image-text matching (Lee et al., 2018; Yu et al., 2014), visual question answering (Antol et al., 2015; Yu et al., 2019b; Cadene et al., 2019), visual grounding (Yu et al., 2018; Hong et al., 2019), and image captioning (Xu et al., 2015; You et al., 2016; Anderson et al., 2018; Lu et al., 2018; Huang et al., 2019).

Image captioning has many uses, including assisting visually impaired people in better perceiving their surroundings, extending human-robot interactions, and many others. In the image captioning task, the model generates a textual caption for a given input image. This task is simple for humans but difficult for computers since it includes detecting the main objects in an image, understanding their relationships to each other, and describing them in natural language.

In the past few years, most image captioning methods (Vinyals et al., 2015; Xu et al., 2015; Anderson et al., 2018; Lu et al., 2017) adopt an encoder-decoder framework, inspired by the sequence-to-sequence model for machine translation (Sutskever et al., 2014). Traditionally in such frameworks, a deep convolutional neural network (CNN) encoder converts the input image into region-based visual features, while a recurrent neural network (RNN) decoder produces caption words based on the extracted visual features. Recent captioning models tend to replace the RNN model in the decoder component with Transformer (Vaswani et al., 2017) because of its potential for parallel training, richer representation, and better performance with long dependencies. The Transformer architecture, which records correlations between areas and words using

self-attention, is an excellent example of such an encoder-decoder architecture (Herdade et al., 2019; Li et al., 2019; Yu et al., 2019a; Cornia et al., 2020; Mishra et al., 2021).

Despite these efforts, these methods and datasets are mostly restricted to 2D images and never utilize 3D information. In this paper, we are interested in exploiting the depth information of RGB-D scenes to improve the description generation task. Towards this goal, we propose a Transformer-based encoder-decoder model to generate multi-sentence descriptions for the 3D scenes. We aim to benefit from the existence of the depth map to enrich the image description and solve the lack of cognition on the spatial relationship in realistic 3D scenes. The depth and RGB images are separate modalities acquired by the sensor, and all of them provide us with information about how the scene appears. Depth images, unlike RGB images, do not offer fine visual details of the objects but rather present more distinct silhouettes of them. The depth channel has already been utilized successfully with CNNs to increase classification accuracy (Schwarz et al., 2015; Eitel et al., 2015) since it has complimentary information to the RGB channels and contains structural information of the image. In addition, depth images show a better description of the relationship between the 2D and 3D spatial positions of the objects in a scene, which enriches the scene description. To demonstrate the effectiveness of our approach, we analyzed the NYU-v2 dataset. From our analysis, we discovered inconsistent labeling, such that the ground truth is not always represented, and the level of information in the descriptions ranges from a one-sentence general overview to a much more thorough one. So, we reviewed and cleaned it to make it consistent and focused. We also show that our model can significantly improve 3D image captioning over the RGB-only baselines. In summary, our contributions are three-fold:

- We propose a cleaned version of the NYU-v2 dataset as a benchmark that is small in size and representative of both 2D and 3D relations.
- We propose an end-to-end trained framework for generating a multi-sentence description of a 3D scene.
- Experimental results show the effectiveness of our method in exploiting depth information and generating better captions.

2 Related work

2.1 Multi-sentence image captioning

The majority of recent research on image captioning concentrates on single-sentence captions. However, providing a single statement is insufficient for expressing a semantically rich image. Krause et al. (2017) introduced the task of image paragraph captioning using an encoder-decoder framework and a hierarchical recurrent neural network. In particular, a sentence RNN was used to produce sentence topics, while several word RNNs are in charge of producing words. Following that, Chatterjee & Schwing (2018) combined coherence vectors with global topic vectors to create various cohesive paragraph topics. Also, Wang et al. (2019) proposed a convolutional auto-encoding method to effectively utilize the image representation throughout the decoding process to extract valuable and representative topics over region-level features. To solve the drawbacks of these RNN-decoder-based approaches, Li et al. (2020) proposed a Dual-CNN language model for generating a paragraph that describes a given image.

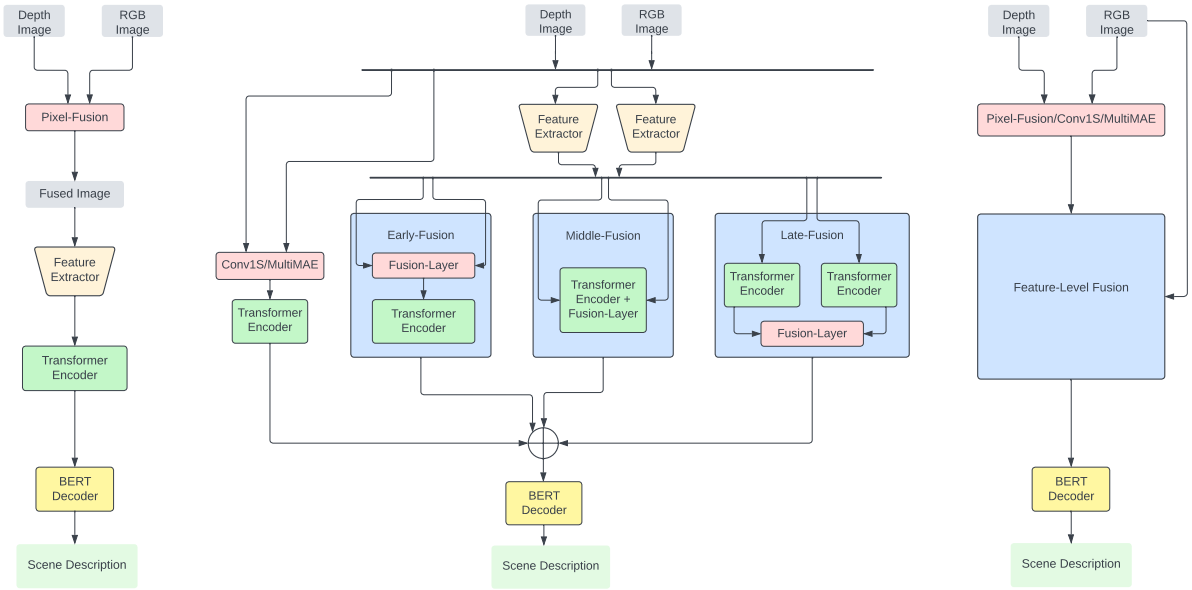
2.2 3D Scene understanding

Several studies have investigated tasks for 3D scene understanding, such as text-to-image alignment (Kong et al., 2014) or generating referring expressions (Chen et al., 2020; Mauceri et al., 2019) for a specific object in an image. Regarding the caption generation task, Lin et al. (2015) proposed a framework for automatically generating multi-sentence linguistic descriptions of complicated indoor scenes. The primary goal of their work was to construct a 3D parsing system to produce a semantic representation of the scene. Because this method is hand-crafted, it rarely generates many sentences from a single image. After that, Moon & Lee (2018) studied the development of natural language descriptions from indoor scenes using graph convolution networks to extract local features from a 3D semantic graph map and long short-term memory (LSTM)

to construct scene descriptions. However, graph convolutional networks typically require auxiliary models (e.g., visual relationship detection and attribute detection models) to generate the visual scene graph in the picture in the first place.

2.3 Self-supervised learning (SSL)

SSL receives supervisory signals directly from the data. Creating an auxiliary pre-text task for the model from the input data is the basic concept behind SSL; as the model completes the auxiliary task, it learns about the underlying structure of the data. For instance, it learns to predict any unseen part of the input from the seen part. Autoencoders (Vincent et al., 2008) instantiate this principle in vision by predicting the missing parts at the pixel or token level. Recently, masked autoencoders (MAE) (He et al., 2022; Bachmann et al., 2022), a class of autoencoders, have attracted attention in computer vision. As the name implies, a masked autoencoder learns representations by reconstructing the input's randomly masked patches. Another approach of SSL that has many successful applications in computer vision is the contrastive learning approach (Hadsell et al., 2006). In contrastive learning, the model learns the image representation by contrasting the input examples. Specifically, the model will try to minimize the representation distance between similar images. Current methods, implemented with Siamese networks (Bromley et al., 1993; Bertinetto et al., 2016), produce two distinct augmentations of samples and feed them into the networks for contrastive learning. SimSiam (Chen & He, 2021) and BYOL (Grill et al., 2020) are two recent, well-known research that used Siamese architectures and were pioneers in removing negative samples from contrastive learning.



(a) Pixel-level fusion (b) Feature-level fusion (c) Hybrid fusion

Figure 1: Overview of our proposed model for a 3D scene description generation task. We studied three distinct approaches to fuse the depth map with the RGB image. (a) Apply fusion to a pixel space (pixel-level fusion). (b) Apply fusion in the feature space (feature-level fusion). (c) Use two feature fusion methods or combine pixel and feature fusion methods (Hybrid fusion)

3 Methodology

This section describes our approach to building a Transformer-based encoder-decoder model for multi-sentence 3D scene captioning. We examined three alternative fusion approaches for fusing depth and RGB images: pixel-level fusion (Figure 1(a)), feature-level fusion (Figure 1(b)), and hybrid fusion (Figure 1(c)).

There are various ways to carry out each approach. We conducted an extensive study with a large number of combinations (30+ architectures) to determine the best one. As Figure 1 shows, the main architectural components of our proposed model are: depth fusion module, feature extractor backbone, Transformer-based encoder block, and BERT decoder block. Our model takes as input both RGB and depth images that represent the 3D scene. The fusion step is responsible for combining the two input images in pixel or feature spaces. The feature extractor backbone generates the feature map for the input image, while the Transformer-based encoder generates a new representation of this feature map. Finally, we use BERT as our language model.

3.1 Depth fusion

Fusing RGB and depth images can be performed in several ways. The different approaches used in our work can be generally divided into three classes: pixel-level fusion methods, feature-level fusion methods, and hybrid methods. Figure 1 represents an overview of these classes which will be explained in the following subsections.

3.1.1 Pixel-level fusion

In pixel-level image fusion, the original data from the source images containing complementary information of the same scene are merged to create a more informative image. We apply this type of fusion in three different ways, as illustrated in Figure 2:

1. Color and depth images are concatenated and mapped into a 3-channel input using the Distance maps fusion module (DMF) (Sofiiuk et al., 2020). It accepts as input a concatenation of a depth image and an RGB image. The DMF module processes the 4-channel input with 1 x 1 convolutions, followed by the ReLU activation function, and outputs a 3-channel tensor that can be sent into the RGB-trained backbone.
2. Inspired by Sofiiuk et al. (2021), we augment the weights of the first convolutional layer to make the pre-trained CNN model take 4-channel input rather than just an RGB image. We refer to this alteration of the network architecture by Conv1E.
3. Fusion at the level of HSV image representation as described by Mechal et al. (2022) by the following steps: First, map the RGB image to HSV representation. Then, replace the V channel with the depth map to get an HSD representation. Finally, use this HSD image as input to the pre-trained CNN model. An additional step conducts the inverse mapping from HSD to RGB before utilizing CNN. We refer to the output of this additional step as an RGBD image.

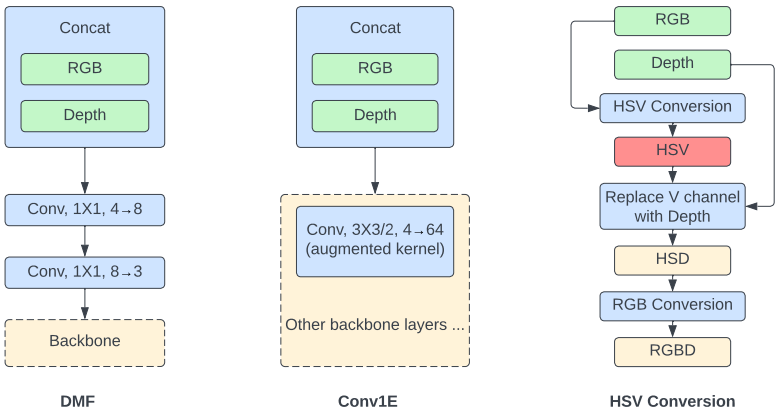


Figure 2: Different pixel-level fusion choices described in section 3.1.1

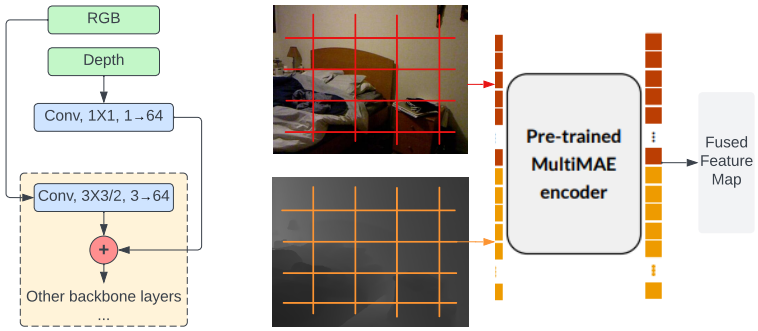
3.1.2 Feature-level fusion

Compared to pixel-level approaches, feature-level fusion methods process data at higher levels. Specifically, we apply feature extraction techniques first to generate a feature map for each image data source. Then, the fusion process deals with these feature maps and outputs a single feature map representing the different modalities. We studied four techniques to apply feature-fusion in our work: concatenation, cross-attention, Conv1S (Sofiiuk et al., 2021), and MultiMAE.

Regarding concatenation and cross-attention methods, we have three position choices to use in our model, illustrated in Figure 1(b). We denoted them by early-fusion, middle-fusion, and late-fusion. The **early-fusion** position is before the encoder block. We perform the fusion over the feature extractor outputs and right before the transformer encoder. The **middle-fusion** choice is to perform the fusion inside the encoder, briefly described in section 3.3. The last choice is **late-fusion**, where we pass each feature map to a separate encoder, then apply the fusion over the encoder outputs.

The idea of Conv1S, as shown in Figure 3(a), is to pass the additional input, e.g., depth image, to a convolutional block that outputs the same shape tensor as the first convolutional block in the backbone. The output of the first backbone convolutional layer is then element-wise summed with this tensor.

Another choice is using a pre-trained MultiMAE. Figure 3(b) shows that we pass the RGB and depth images to the pre-trained MultiMAE encoder and use the output feature map as the fused features to our model. In our work, we denote MultiMAE output features based on RGB and depth inputs by **MAE_CD**.



(a) Conv1S (b) MultiMAE

Figure 3: Conv1S and MultiMAE methods. Example approaches for fusing RGB and depth images at the feature level

3.1.3 Hybrid fusion

Hybrid methods use the advantages of both pixel-level fusion and feature-level fusion methods by combining them (**pixel-feature**). An example of this approach is combining HSV conversion with feature-level fusion. Another way to apply hybrid fusion is by combining two feature-level fusion methods (**feature-feature**), as in applying the concatenation/cross-attention method with the MultiMAE features.

3.2 Feature extractor backbone

As the first step in our model, we need to extract features of input images. To do this, we employed a pre-trained feature extractor model, either a supervised model such as a pre-trained CNN (e.g., EfficientNet (Tan & Le, 2019)) or a pre-trained model in an unsupervised fashion such as a masked autoencoder (e.g., ViTMAE (He et al., 2022) and MultiMAE (Bachmann et al., 2022)). To investigate different methods to extract depth features, we trained the SimSiam (Chen & He, 2021) on the SUN RGB-D depth dataset (Song et al., 2015) and then used it as a depth feature extractor model. We did the same thing with the fused

images (HSD/RGBD) and trained a SimSiam model on the fused images generated from the SUN RGB-D dataset.

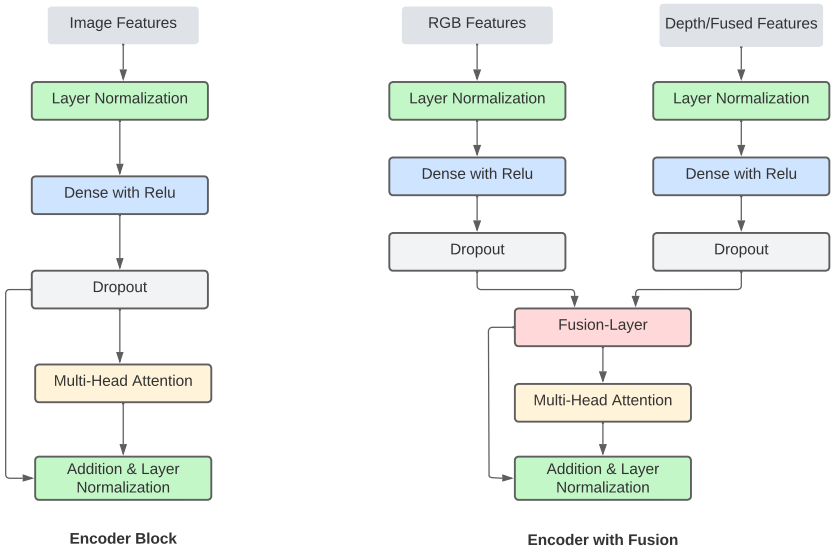


Figure 4: Transformer-based Encoder block architecture. **Left:** the base encoder block used in all experiments except in the case of the middle fusion. **Right:** the encoder block used in the case of the middle-fusion approach

3.3 Transformer-based encoder

Given a set of visual features extracted from the previous step, we passed them to a Transformer-based encoder. As shown in Figure 4, the transformer encoder consists of 5 layers that process the input consecutively: 1) normalization layer; 2) dense layer with ReLU; 3) dropout layer; 4) multi-head self-attention layer; 5) layer normalization with a residual connection. Figure 4 also shows the encoder in the case of the middle-fusion under the feature-level fusion methods. In this case, the encoder has two feature maps as inputs instead of one. Each feature map sequentially passes through a normalization, dense, and dropout layer. Then, the fusion layer, either concatenation or cross-attention, combines these projected features. After that, we apply the multi-head self-attention to the fused output. Finally, apply the layer normalization with a residual connection.

3.4 BERT decoder

The encoder outputs are fed into a language decoder model to generate scene descriptions. We use an uncased pre-trained BERT model (Devlin et al., 2018) as a language decoder. BERT is trained on English Wikipedia and BooksCorpus using a combination of Masked Language Modeling and next-sentence prediction tasks. We fine-tuned the BERT decoder during the training to steer the language model towards the new objective.

4 Dataset

To test the proposed framework, we use the NYU-v2 dataset (Lin et al., 2015), which has 1449 RGB-D images of indoor scenes and one description per image. The number of sentences in each description ranges from one to ten, with an average of three. Following the partition employed in Lin et al. (2013), these images are split into a training set and a testing set. The training set includes 795 scenes, while the testing set includes the remaining 654.

Table 1: Original dataset statistics. Shows how many times each object class is mentioned in the text in relation to all visual occurrences of that class

	mantel	counter	toilet	sink	bathtub	bed	headb.	table	shelf	cabinet	sofa	chair	chest	refrig.	oven	microw.	blinds	curtain	board	monitor	printer
% mentioned	41.4	46.4	90.6	78.2	55.6	79.4	9.2	53.4	32.9	27.7	51.2	31.6	45.7	55.7	31.4	56.0	26.2	23.0	28.6	66.7	44.3

4.1 Dataset cleaning

Because the dataset's annotators had no idea of the classes of interest in the images, the descriptions in the dataset are inconsistent. Moreover, the annotators were asked to describe the scene to someone who does not see it without any additional instructions. As a result, the descriptions vary from a general one-sentence summary to one that is more detailed. The dataset has 21 objects of interest classes, as mentioned in Lin et al. (2013). According to Kong et al. (2014), Table 1 shows the object of interest classes and the percentage of referring times to each object concerning the number of all visual objects of that class. This percentage, as we can see, ranges from 9% for items like a "headboard" to 91% for items such as a "toilet."

Another issue we discovered was the non-uniform point of reference while describing the scene. Sometimes it's from the viewpoint of the viewer, and other times it's from the perspective of the described object.

To address the inconsistency in the dataset descriptions, we manually reviewed the dataset and applied consistent relabeling. We follow the following rules in the relabelling process:

- Concentrate on the 21 objects of interest (listed in table 1).
- Describe the object's appearance, such as shape, color, and material.
- Describe the 2D and 3D spatial relationships between the scene's objects.
- Unite the point of reference to the viewer's point of view.

Figure 5 shows examples from the dataset before and after the relabelling process. In the first two images (5(a), 5(b)), for instance, the scene caption describes objects that are not part of the interest classes (e.g., fan, ironing board, lamp, and walls). In such cases, we remove the description of those objects to keep the focus on the interest classes. The next two images (5(c), 5(d)), on the other hand, represent a case where an object that belongs to the interest classes is not mentioned in the description (e.g., cabinets). In that case, we add a description to this object. The example where we changed the point of reference to apply the final rule and unify the point of view to the viewer is shown in the images (5(e), 5(f), 5(g)). The final case illustrated in the last two images (5(h), 5(i)), where we keep the description as is because it satisfies the conditions.

5 Experiments

5.1 Evaluation metrics

We evaluate the caption quality using the standard automatic evaluation metrics, namely BLEU-1, BLEU-4 (Papineni et al., 2002), ROUGE-1, ROUGE-L (ROUGE, 2004), and METEOR (Denkowski & Lavie, 2014). They are denoted as B-1, B-4, R-1, R-L, and M, respectively. METEOR and ROUGE-L are f-measure scores, ROUGE-N is a recall-based measure, and BLEU is a precision-based measure. METEOR has more flexible matching requirements, therefore it is the first targeted metric, followed by ROUGE-L, ROUGE-1, BLEU-4, and BLEU-1.

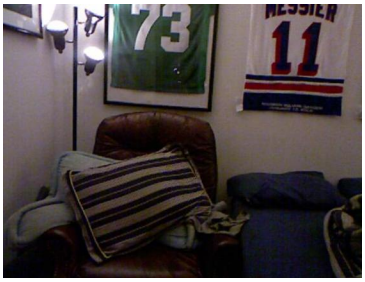
5.2 Baselines

To answer the central question in our study, "*Can we use the depth information to produce better captions?*" we evaluate our proposed model with different depth fusion approaches against previous work and an ablated baseline.

The baseline model takes as input the RGB image, uses an EfficientNet pre-trained backbone to extract the image features, and then feeds the image features to the Transformer-based encoder. After that, the language decoder (BERT) uses the encoder output to generate the caption. In one experiment, we trained the BERT from scratch, whereas in the other, we fine-tuned the pre-trained BERT.

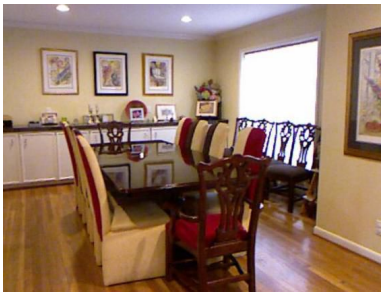
Show and Tell (Xu et al., 2015) use RNN as a textual decoder with attention over the image features to produce the related words.

(a) **Before:** A bedroom with dark walls and a bed against the right wall. A fan is on the ceiling above the bed. There is an ironing board by the left window. Another window is in the back wall.
After: A bed with metal frame is against the right wall in this bedroom. White blinds cover two windows, one behind the bed and the other to its left.



(b) **Before:** A room with a leather chair in the corner, on which two pillows are placed. Behind the chair there is a tall lamp with three light bulbs. There are three sports t-shirts hanging on the wall behind the chair.
After: It is a room with a leather chair in the corner, on which two pillows are placed.

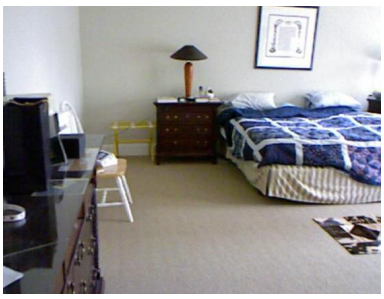
(c) **Before:** A tall man is working in his study room. He is standing next to a table which is covered with paper and notebooks. There are three chairs next to the table.
After: There is a table covered with paper and notebooks in front of us. Three chairs are around the table. Behind the table are a small cabinet and a window curtain to its right.



(d) **Before:** This is a dining room with a long wooden table and white and red chairs next to the table. It could also be a meeting room since there are extra chairs behind the table. The room has four paintings on the walls.
After: This is a dining room with a long wooden table and white and red chairs next to the table. It could also be a meeting room since there are extra chairs behind the table. By the back wall are white cabinets.

(e) **Before:** A living room with a chess table in the middle of the room. To the right of the table are two blue armchairs. Behind the table and armchairs is a fireplace. Along the same wall is a cabinet with books.
After: It's a living room with a chess table surrounded by four small chairs in the middle of the room. Two blue armchairs are to the left of the table. Behind the table and armchairs is a fireplace. Along the same wall is a cabinet with books.

(f) **Before:** A neatly furnished room with a double bed, pillows on top of the bed and a cabinet with six drawers to the right of the bed. On top of the cabinet is a lamp.
After: This is a neatly furnished bedroom. There is a double bed with a dark wooden frame and pillows on top of the bed. A cabinet with drawers is to the left of the bed.



(g) **Before:** This is a bedroom with a bed that has two pillows on top, a blue-white bed sheet, two night lamps on both sides of the bed, and a cabinet directly on our right.
After: This is a large bedroom with a bed that has two pillows on top and a blue-white bed sheet. There is a night-stand on each side of the bed. Near the left corner is a small, yellow table or chair. A cabinet is directly on our left and a chair behind it.

(h) **Before:** This is a picture of a meeting room. There are purple chairs around the meeting table, and a white board with drawings.
After: This is a picture of a meeting room. There are purple chairs around the meeting table, and a white board with drawings.

(i) **Before:** Two printers are on top of a cabinet. There are also a bunch of other things on top of the cabinet. Above the printers there is a mail holder.
After: Two printers are on top of a cabinet. There are also a bunch of other things on top of the cabinet. Above the printers there is a mail holder.

Figure 5: Examples from the NYU-v2 dataset before and after the cleaning step. **(a) and (b)** are examples of removing the description of objects other than the classes of interest. **(c) and (d)** are examples of adding descriptions to objects belonging to interest classes. **(e), (f), and (g)** are examples of change the point of reference. **(h) and (i)** are examples of not changing the scene description

Show and Tell with transformers, a modified version of Xu et al. (2015), uses a 3-layer Transformer-decoder. We use the implementation provided by Tensorflow (2022).

VirTex(Desai & Johnson, 2021) jointly train a convolutional network (ResNet-50) and Transformers (two unidirectional Transformers) from scratch.

5.3 Implementation details

We trained our model end-to-end with the decoder initialized by the pre-trained BERT-BASE model. The input images are resized to 224 $\times$ 224 resolution. We keep the pre-trained feature extractor backbone frozen across all experiments. The output dimension of the token embedding layers in the BERT decoder is 768, and the input visual features are also mapped into 768 with a linear projection inside the transformer encoder. We used a grid search to select the best hyperparameters in our study. We adjusted the number of attention heads, learning rate, BERT dropout, and encoder dropout separately for each experiment. The whole model is trained with cross-entropy loss. We use a batch size of 16 through all experiments unless explicitly stated otherwise. For optimization, we use AdamW optimizer(Kingma & Ba, 2014). We use the Tensorflow (Abadi et al., 2016) framework and run all experiments on a single Nvidia Geforce RTX 2070 Super GPU.

6 Results and discussion

6.1 Main results

2D vs. 3D Captioning. We compare our 3D captioning approach to the baseline models on the official test split of the NYUv2 sentence dataset (Lin et al., 2013) in Table 2. We report BLEU-1, BLEU-4, ROUGE-L, and METEOR results. Regarding our approach, we list the results of the best configuration for each depth fusion method. We fine-tuned the hyperparameters of each method independently to ensure a fair comparison. Table 2 shows that our approach outperforms the baselines with a remarkable margin. The hybrid fusion method achieves the highest score for the METEOR metric (with a **3.59** improvement), while

the feature-level fusion method is better in BLEU-1 and BLEU-4 metrics (with **3.58** and **1.65** improvements, respectively). The ROUGE scores for the two methods are comparable (**1.17** and **1.20** for ROUGE-1 and ROUGE-L, respectively).

Table 2: Comparison of 3D captioning results obtained by our model and the baseline models. **Black** indicates results better than baselines and **Blue** indicates the best results

	Modality	Inputs	Method	B-1	B-4	R-1	R-L	M
Show-and-Tell (Xu et al., 2015)	2D	RGB	-	26.22	6.54	40.41	32.65	23.61
Show-and-Tell(Transformer) (Tensorflow, 2022)	2D	RGB	-	35.81	6.95	35.14	29.3	24.95
VirTex (Desai & Johnson, 2021)	2D	RGB	-	30.56	6.39	39.72	36.08	25.51
Our baseline(BERT from scratch)	2D	RGB	-	35.58	8.67	42.57	38.76	26.24
Our baseline(pre-trained BERT)	2D	RGB	-	42.4	10.99	44.5	41.04	30.25
Ours (Pixel-level)	3D	RGB+Depth	DMF	**44.69**	**11.65**	43.67	40.28	**30.83**
Ours (Feature-level)	3D	RGB+Depth	Late-fusion (Cross-attention)	**45.98**	**12.64**	**45.67**	**42.22**	**32.87**
Ours (Hybrid)	3D	RGB+MAE_CD	Late-fusion (Concat)	41.05	10.62	**45.65**	**42.24**	**33.84**

Pixel-Level fusion. Table 3 compares the performance of the three pixel-level fusion methods. DMF performs the best in all assessed metrics except for METEOR. The highest METEOR score is achieved by HSV conversion with RGBD.

Table 3: Evaluation results of the pixel-level fusion methods described in section 3.1.1

Method	Inputs	B-1	B-4	R-L	M
DMF	RGB+Depth	**44.69**	**11.65**	**40.28**	30.83
Conv1E	RGB+Depth	43.5	10.86	40.23	30.85
HSV Conversion	HSD	40.69	9.44	37.16	28.17
	RGBD	43.64	11.02	39.98	**32.04**

Featur-Level Fusion. We examine the effectiveness of the feature fusion methods presented in Section 3.1.2 in Table 4. This table also shows the positions where fusion can occur. Overall, all of the methods outperform the baseline models. Using a cross-attention layer to perform late fusion RGB and depth images achieves the best results in all assessed metrics.

Table 4: Evaluation results of the feature-level fusion methods described in section 3.1.2

Method	Inputs	B-1	B-4	R-L	M
Conv1S	RGB+Depth	44.47	11.41	40.48	31.04
MultiMAE		43.64	11.02	39.98	32.04
Early-Fusion(Cross-attention)		42.57	11.4	41.76	31.65
Middle-Fusion(Cross-attention)		43.72	11.83	40.73	31.05
Late-Fusion(Cross-attention)		**45.98**	**12.64**	**42.22**	**32.87**

Concatenation vs. Cross-attention Table 5 compares concatenation and cross-attention as a used fusion layer. We first evaluate in case of the early-fusion position, and the results for cross-attention outperform the concatenation method. We then evaluated the results of the middle-fusion position, and concatenation is better than cross-attention. Lastly, we evaluated the results of the late-fusion position with RGB and

depth images as inputs, as well as RGB and MultiMAE features as inputs. Cross-attention outperforms concatenation on RGB and depth images. While for RGB and MultiMAE features, cross-attention results are better in the BLEU score, concatenation results are better in both ROUGE and METEOR scores.

Table 5: Comparison of concatenation and cross-attention methods at three different positions of our model. **Bold** indicates the best results in each position

Inputs	Position	Method	B-1	B-4	R-L	M
RGB+Depth	Early-fusion	Concat	41.27	10.86	41.36	30.8
		Cross-attention	**42.57**	**11.4**	**41.76**	**31.65**
RGB+Depth	Middle-fusion	Concat	**45.02**	**12.31**	**41.22**	**31.79**
		Cross-attention	43.72	11.83	40.73	31.05
RGB+Depth	Late-fusion	Concat	44.36	11.29	40.83	31.05
		Cross-attention	**45.98**	**12.64**	**42.22**	**32.87**
RGB+MAE_CD	Late-fusion	Concat	41.05	10.62	**42.24**	**33.84**
		Cross-attention	**45.86**	**11.63**	40.64	32.25

Hybrid Fusion. In Table 6, we compare the performance of the hybrid fusion approach. The method can be implemented by combining pixel and feature techniques (pixel-feature) or by combining two feature methods (feature-feature) (Section 3.1.3). The best results for BLEU-4 and ROUGE-L (**1.76** and **1.23** improvements, respectively, over the baseline) are obtained with pixel-feature. The best results for BLEU-1 and METEOR (**4.54** and **3.59** improvements, respectively, over the baseline) are obtained with feature-feature.

Table 6: Evaluation results of hybrid fusion approach. **Top:** evaluation results of pixel and feature fusion method combination (pixel-feature). **Bottom:** evaluation results for fusing two feature-based methods (feature-feature). **Black** indicates the best results in each category and **Blue** indicates the best results overall

	Method	Inputs	B-1	B-4	R-L	M
Pixel-Feature	Early-fusion(Cross-attention)	RGB+RGBD	42.59	11.38	42	31.61
	Middle-fusion(Concat)	RGB+RGBD	43.88	11.89	41.45	31.95
	Late-fusion(Cross-attention)	RGB+RGBD	**45.97**	**12.75**	**42.27**	**32.71**
Feature-Feature	Conv1S,Middle-fusion(Concat)	RGB+Depth+MAE_CD	44.31	10.6	39.63	31.27
	Middle-fusion(Concat)	RGB+MAE_CD	**46.94**	12.25	41.68	32.16
	Late-fusion(Concat)	RGB+MAE_CD	41.05	10.62	**42.24**	**33.84**
	Late-fusion(Cross-attention)	RGB+MAE_CD	45.86	**11.63**	40.64	32.25

Fusion Position. We compare the performance of our model with the different fusion position choices as described in section 3.1.2. Table 7 demonstrates that when employing the cross-attention as a fusion approach, the late-fusion position works better than both early and middle fusion. Middle-fusion works equally well or better than late-fusion when employing the concatenation approach. When the inputs are RGB and MultiMAE_CD, middle-fusion performs better for BLEU scores, while late-fusion outperforms in the ROUGE and METEOR scores.

Dataset Cleaning. In Table 8, we study the dataset-cleaning effect. We evaluate the show-and-tell model, our baseline, and proposed models on the dataset before and after the cleaning step. Noticeably, the evaluation results of all models on the cleaned dataset outperform their results on the original dataset by

Table 7: comparing the results of our model performance with the different fusion positions (early-fusion, middle-fusion, and late-fusion). **Black** indicates the best results among the three positions with respect to the input and **Blue** indicates the best results overall

Fusion Position	Inputs	Method	B-1	B-4	R-L	M
Early-Fusion	RGB+Depth	Concat	41.27	10.86	**41.36**	30.8
Middle-Fusion			**45.02**	**12.31**	41.22	**31.79**
Late-Fusion			44.36	11.29	40.83	31.05
Early-Fusion	RGB+Depth	Cross-attention	42.57	11.4	41.76	31.65
Middle-Fusion			43.72	11.83	40.73	31.05
Late-Fusion			**45.98**	**12.64**	**42.22**	**32.87**
Early-Fusion	RGB+RGBD	Concat	37.55	10.26	40.09	28.98
Middle-Fusion			43.88	**11.89**	**41.45**	**31.95**
Late-Fusion			**45.75**	11.57	41.1	31.66
Early-Fusion	RGB+RGBD	Cross-attention	42.59	11.38	42	31.61
Middle-Fusion			38.91	9.43	39.35	28.27
Late-Fusion			**45.97**	**12.75**	**42.27**	**32.71**
Early-Fusion	RGB+HSD	Cross-attention	42.77	11.26	42.01	31.53
Middle-Fusion			40.44	10.47	40.09	29.55
Late-Fusion			**45.85**	**12.65**	**42.11**	**32.66**
Middle-Fusion	RGB+MAE_CD	Concat	**46.94**	**12.25**	41.68	32.16
Late-Fusion			41.05	10.62	**42.24**	**33.84**

a remarkable margin. This shows that the cleaned dataset can benefit not just our model but also other models.

Moreover, in the case of depth fusion, the proposed models perform worse than the baseline model on the original dataset. This indicates that the original dataset's inaccurate labeling has an impact on the depth signal's capturing and prevents the model from learning from the depth data.

6.2 Ablation study and analysis

Early-fusion vs. Late-fusion. The late-fusion approach has more parameters than the early-fusion approach because it uses two separate Transformer-based encoders compared to the early-fusion approach's single Transformer-based encoder. For a fair comparison, we compensate the number of parameters in both approaches by increasing the number of trainable variables in the early-fusion approach. We did that by simply using two encoders instead of one, where the output of the first encoder goes as input to the second.

Table 9 shows that, even after increasing the number of trainable variables in the early-fusion approach, the late-fusion approach still outperforms it. This implies the improvement comes from the fusion position, not the number of trainable variables.

MultiMAE: RGB, Depth, or RGB+Depth. Because MultiMAE was pre-trained on RGB, depth, and semantic segmentation, it may accept any of those modalities as input. In our work, we have RGB and depth images. So we can pass RGB, depth, or RGB+Depth to the pre-trained MultiMAE. As shown in the results of the previous experiments, using MultiMAE features based on RGB and depth modalities outperforms the

Table 8: Results of our model and the show-and-tell model before and after the relabeling process

Model	Dataset	Inputs	B-1	B-4	R-L	M
Show-and-Tell(Transformer)	Original	RGB	33.37	5.14	26.38	22.85
Ours	Original	RGB	41.95	9.78	38.28	28.33
		RGB+Depth	41.73	9.22	37.11	27.68
		RGB+MAE_CD	39.8	7.44	35.18	26.07
Show-and-Tell(Transformer)	Cleaned	RGB	35.81	6.95	29.3	24.95
Ours	Cleaned	RGB	42.4	10.99	41.04	30.25
		RGB+Depth	**45.98**	**12.64**	42.22	32.87
		RGB+MAE_CD	41.05	10.62	**42.24**	**33.84**

Table 9: Evaluation results of early-fusion and late-fusion position after equalizing the number of learnable parameters

Method	Params.	B-1	B-4	R-L	M
Early-Fusion(Cross-attention)	344	42.57	11.4	41.76	31.65
Early-Fusion(Cross-attention)	358	**46.07**	12.43	41.49	31.88
Late-Fusion(Cross-attention)	358	45.98	**12.64**	**42.22**	**32.87**

baseline. *Is this improvement because of fusing depth modality with RGB or a better RGB representation by the MultiMAE?* In this ablation, we aim to answer this question by comparing the performance of our model with RGB, depth, and RGB+Depth MultiMAE features. We did all experiments in Table 10 by applying late-fusion using the concatenation method between the RGB image and the MultiMAE output features.

The results show that our model performs better with RGB+Depth MultiMAE features. Moreover, interestingly, using RGB-only MultiMAE features has comparable results as our baseline model. In conclusion, the findings in this section demonstrate that the MultiMAE features' ability to enhance performance is not due to a richer RGB representation but rather to the use of depth information.

Table 10: Ablation results of different MultiMAE input modalities. All experiments were carried out using late-fusion with the concatenation method between the RGB image and the MultiMAE output features

Method	MultiMAE Inputs	B-1	B-4	R-L	M
Our baseline	-	42.4	10.99	41.04	30.25
Late-Fusion(Concat)	RGB	**44.5**	**10.83**	39.5	30.2
	Depth	41.51	10.52	41.09	32.27
	RGB+Depth	41.05	10.62	**42.24**	**33.84**

Feature extractor backbone. Our experiments thus far are based on a frozen feature extractor backbone. In this ablation, we examine whether fine-tuning the feature extractor backbone will improve our results. To do this, we trained the entire model across several epochs with the frozen feature extractor backbone.

Then, unfreeze a few of the feature extractor backbone's top layers and jointly train these last layers with the rest of our model.

Table 11 shows the results. We attempted to fine-tune a different number of top layers in the backbone. We did all experiments in Table 11 by applying early-fusion using the concatenation method between RGB and depth images. The results clearly suggest that our model performs better with a frozen backbone.

Table 11: Ablation results between frozen and fine-tuned feature extractor backbone. All experiments were carried out using early-fusion with the concatenation method between the RGB image and the depth images

Backbone	Unfreezed-Layers	Inputs	B-1	B-4	R-L	M
Frozen	-	RGB+Depth	**41.27**	**10.86**	**41.36**	**30.8**
Fine-tuned	last 40		38.18	9.93	40.35	28.72
Fine-tuned	last 90		38.17	9.83	40.14	28.59
Fine-tuned	last 140		38.5	10.13	40.59	29.11

7 Conclusion

In our work, we proposed a Transformer-based encoder-decoder model that can efficiently exploit depth information, resulting in improved image captioning. Our model takes the RGB and depth images representing the scene and fuses them to get richer information about the input scene. Moreover, we explore different fusion approaches to fuse the depth map with the RGB image. Extensive experiments on a cleaned version of the challenging NYU-v2 sentence dataset validate the effectiveness and superiority of our approach. Cleaning the dataset has been shown to impact the model's performance significantly. Overall, we are hopeful that our cleaned version of the NYU-v2 dataset and framework can support further study in the area of 3D visual language.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 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*, pp. 6077–6086, 2018.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. *arXiv preprint arXiv:2204.01678*, 2022.

Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pp. 850–865. Springer, 2016.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

Remi Cadene, Hedi Ben-Younes, Matthieu Cord, and Nicolas Thome. Murel: Multimodal relational reasoning for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1989–1998, 2019.

Moitreya Chatterjee and Alexander G Schwing. Diverse and coherent paragraph generation from images. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 729–744, 2018.

Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *European Conference on Computer Vision*, pp. 202–221. Springer, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10578–10587, 2020.

Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pp. 376–380, 2014.

Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11162–11173, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 681–687. IEEE, 2015.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. *Advances in Neural Information Processing Systems*, 32, 2019.

Richang Hong, Daqing Liu, Xiaoyu Mo, Xiangnan He, and Hanwang Zhang. Learning to compose and reason with language tree structures for visual grounding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4634–4643, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Chen Kong, Dahua Lin, Mohit Bansal, Raquel Urtasun, and Sanja Fidler. What are you talking about? text-to-image coreference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3558–3565, 2014.

Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 317–325, 2017.

Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 201–216, 2018.

Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8928–8937, 2019.

Ruifan Li, Haoyu Liang, Yihui Shi, Fangxiang Feng, and Xiaojie Wang. Dual-cnn: A convolutional language decoder for paragraph image captioning. *Neurocomputing*, 396:92–101, 2020.

Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *Proceedings of the IEEE international conference on computer vision*, pp. 1417–1424, 2013.

Dahua Lin, Chen Kong, Sanja Fidler, and Raquel Urtasun. Generating multi-sentence lingual descriptions of indoor scenes. *arXiv preprint arXiv:1503.00064*, 2015.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 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*, pp. 375–383, 2017.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7219–7228, 2018.

Cecilia Mauceri, Martha Palmer, and Christoffer Heckman. Sun-spot: an rgb-d dataset with spatial referring expressions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

Chaymae El Mechal, Najiba El Amrani El Idrissi, and Mostefa Mesbah. Cnn-based obstacle avoidance using rgb-depth image fusion. In *WITS 2020*, pp. 867–876. Springer, 2022.

Santosh Kumar Mishra, Rijul Dhir, Sriparna Saha, Pushpak Bhattacharyya, and Amit Kumar Singh. Image captioning in hindi language using transformer networks. *Computers & Electrical Engineering*, 92:107114, 2021.

Jiyoun Moon and Beomhee Lee. Scene understanding using natural language description based on 3d semantic graph map. *Intelligent Service Robotics*, 11(4):347–354, 2018.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Lin CY ROUGE. A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization of ACL, Spain*, 2004.

Max Schwarz, Hannes Schulz, and Sven Behnke. Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features. In *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 1329–1335. IEEE, 2015.

Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8623–8632, 2020.

Konstantin Sofiiuk, Ilia A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. *arXiv preprint arXiv:2102.06583*, 2021.

Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 567–576, 2015.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Tensorflow. Image captioning with visual attention, 2022. URL `https://www.tensorflow.org/tutorials/text/image_captioning`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

Jing Wang, Yingwei Pan, Ting Yao, Jinhui Tang, and Tao Mei. Convolutional auto-encoding of sentence topics for image paragraph generation. *arXiv preprint arXiv:1908.00249*, 2019.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057. PMLR, 2015.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4651–4659, 2016.

Jun Yu, Jing Li, Zhou Yu, and Qingming Huang. Multimodal transformer with multi-view visual representation for image captioning. *IEEE transactions on circuits and systems for video technology*, 30(12): 4467–4480, 2019a.

Zhou Yu, Fei Wu, Yi Yang, Qi Tian, Jiebo Luo, and Yueting Zhuang. Discriminative coupled dictionary hashing for fast cross-media retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pp. 395–404, 2014.

Zhou Yu, Jun Yu, Chenchao Xiang, Zhou Zhao, Qi Tian, and Dacheng Tao. Rethinking diversified and discriminative proposal generation for visual grounding. *arXiv preprint arXiv:1805.03508*, 2018.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6281–6290, 2019b.