remote sensing

Article

CapERA: Captioning Events in Aerial Videos

Laila Bashmal [1], Yakoub Bazi [1,*], Mohamad Mahmoud Al Rahhal [2], Mansour Zuair [1] and Farid Melgani [3]

[1] Computer Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[2] Applied Computer Science Department, College of Applied Computer Science, King Saud University, Riyadh 11543, Saudi Arabia
[3] Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy
* Correspondence: ybazi@ksu.edu.sa

Abstract: In this paper, we introduce the CapERA dataset, which upgrades the Event Recognition in Aerial Videos (ERA) dataset to aerial video captioning. The newly proposed dataset aims to advance visual–language-understanding tasks for UAV videos by providing each video with diverse textual descriptions. To build the dataset, 2864 aerial videos are manually annotated with a caption that includes information such as the main event, object, place, action, numbers, and time. More captions are automatically generated from the manual annotation to take into account as much as possible the variation in describing the same video. Furthermore, we propose a captioning model for the CapERA dataset to provide benchmark results for UAV video captioning. The proposed model is based on the encoder–decoder paradigm with two configurations to encode the video. The first configuration encodes the video frames independently by an image encoder. Then, a temporal attention module is added on the top to consider the temporal dynamics between features derived from the video frames. In the second configuration, we directly encode the input video using a video encoder that employs factorized space–time attention to capture the dependencies within and between the frames. For generating captions, a language decoder is utilized to autoregressively produce the captions from the visual tokens. The experimental results under different evaluation criteria show the challenges of generating captions from aerial videos. We expect that the introduction of CapERA will open interesting new research avenues for integrating natural language processing (NLP) with UAV video understandings.

Keywords: unmanned aerial vehicles (UAVs); aerial videos; video captioning; transformers

check for updates

Citation: Bashmal, L.; Bazi, Y.; Al Rahhal, M.M.; Zuair, M.; Melgani, F. CapERA: Captioning Events in Aerial Videos. *Remote Sens.* **2023**, *15*, 2139. https://doi.org/10.3390/rs15082139

Academic Editor: Qi Wang

Received: 19 December 2022
Revised: 30 March 2023
Accepted: 17 April 2023
Published: 18 April 2023

1. Introduction

Unmanned aerial vehicles (UAVs) have witnessed a dramatic advancement over the last decade. The technological improvements in acquisition sensors and avionics systems have made these platforms a popular tool for mapping and monitoring environments. Compared with satellites and manned aerial vehicles, UAVs offer an inexpensive and easy-to-deploy platform for aerial data acquisition with high-resolution and real-time streaming capabilities. In addition, UAVs can fly at various altitudes, speeds, and over different locations, offering higher flexibility and mobility to complete missions over inaccessible areas. As a result of these peculiarities and features, the use of UAVs has proliferated over recent years, and aerial data have been utilized in various applications such as traffic management, reconnaissance operations, and environmental monitoring.

With the increasing number of videos generated by UAVs, there is a growing demand for automating the interpretation of these videos to understand their content. Deep learning is currently the mainstream tool of the most successful automatic scene-understanding approaches because it can extract high-level semantic knowledge from the given data. However, the success of many deep learning methods is highly dependent on the availability of large-scale annotated datasets, and it is a key factor for these models to learn useful information from the data. Datasets such as Microsoft COCO [1] and ImageNet-21K [2] have greatly contributed to the development of the current deep-learning models in computer vision. Unfortunately, annotating large datasets is an expensive, time-consuming, and tedious process, and for the same reasons, constructing aerial datasets remains a major challenge.

Over the past years, researchers have started to pay attention to applications that involve analyzing images and videos collected from UAVs. Consequently, UAV-captured datasets have been created to serve different applications such as object detection [3], semantic segmentation [4], and anomaly detection [5]. However, existing datasets have limitations in multiple aspects. First, most of the UAV datasets are primarily focusing on the tasks of object detection and tracking [6–8], which both aim to locate a specific object in the image or the video at hand, such as a vehicle [9] or plant [10]. Secondly, the available UAV datasets are annotated with a single semantic label about the main subject in the image or video [11], with some efforts to concurrently detecting the presence of multiple objects [12–16]. However, it would be preferable if the recognition system is able to recognize not only the main subject appearing in the UAV scene, but also to provide a thorough and complete scene description that reflects the human cognitive skill of giving a concise description. However, there already exist UAV caption datasets that provide human-like interpretation for the whole scene. These datasets have been created for applications within restricted environments such as generating descriptions from construction sites [17]. These limitations in terms of the tasks, the subjects, and the environments bring urgent demand to a benchmark that serves general surveillance and monitoring applications.

Among the existing datasets, the Event Recognition in Aerial Videos (ERA) dataset [11] has been built for the overall understanding of aerial scenes and videos in more open and unconstrained environments. This dataset contains UAV images and videos that represent diverse scenarios in our daily life such as traffic, security, disasters, sports, social activity, and normal nonevent videos. Different benchmark models based on 3D-CNN have been evaluated for event recognition by using this dataset [11]. Other works have utilized the ERA dataset and proposed different methods for recognizing the main event in the video. In [18], the authors proposed the formulation of the problem of recognizing UAV videos in unconstrained environments as an open-set problem in which a non-event video is classified as an unknown. Jin et al. [19] proposed a two-path model, one path for generating a holistic representation and the other for modeling the temporal relations between frames. The ERA dataset was built for describing the dominant event in the aerial video with the goal of assigning each video with a coarse label such as ‘fire’ or ‘traffic collision’. However, a UAV video contains a large quantity of information, which cannot be fully described with a single label. Giving a caption to the UAV video can provide a further level of understanding of its content that includes different objects, their attributes, as well as the ongoing dynamics therein. Hence, integrating NLP with video understanding is critical to completely capturing the high-level content of the video.

Given a UAV video, the caption-generation task aims to automatically describe its visual content with a text in natural language. The caption is expected to comprehensively interpret what is going on in the video from a high-level perspective, including the scene, events, objects, their attributes, their evolution over time, and the interrelationships between them. Providing a description for a video recorded by a UAV helps to gain a deeper insight for its visual content, which could be the cornerstone for more advanced applications such as intelligent visual surveillance, fine-grained event recognition, and urban and rural scene understanding. The task of caption generation has been previously investigated for static UAV imagery [20]. Nevertheless, the development of UAV video captioning methodologies is still limited due to the lack of publicly available video–language datasets. Consequently, building a UAV video captioning benchmark is pivotal to advance visual–language integration research in remote sensing.

Generating a description for a video recorded by a UAV is particularly challenging due to several factors. Generally, these videos have varying spatial resolutions of the subjects and noticeable motion blur and present varied and wide viewing angles. This is mainly due to the flexibility of the platform that results from fast motion and constantly changing attitudes during flight. In contrast to image captioning, which aims to describe a static scene captured by UAV, a video consists of a sequence of frames that typically record interactions between people or objects that evolve over time. Hence, video captioning is more difficult in that the dynamics between consecutive frames must be understood in addition to spatial information from each frame to generate a descriptive sentence. Thus, the development of captioning methods should take into consideration these unique characteristics.

To address these issues, we present CapERA, which is a new dataset (publicly available at: http://www.github.com/yakoubbazi/CapEra (accessed on 23 November 2022)) that extends the ERA dataset to the context of video captioning [11]. The dataset consists of 2864 videos and 14,320 captions, where each video is paired with five unique captions. This dataset is constructed for UAV video captioning, but it can possibly empower other video–language integration tasks (e.g., content-based text-videos retrieval, video question answering). The ground-truth captions are carefully annotated by a human expert and augmented by automatic caption generation methods to alleviate the heavy workload of annotation. We also propose two caption generation models for this task to provide benchmarks for UAV video captioning. In summary, our contributions are mainly two-fold:

(1) We introduce a UAV video-caption dataset named CapERA, which is an extension to the ERA dataset with diverse textual descriptions. To the best of the author's knowledge, no labeled dataset is available for captioning UAV videos. The proposed dataset is comprehensive and challenging that will serve as a new benchmark for the community and can promote vision–language research for UAV videos, including text-video retrieval and Visual Question Answering (VQA).

(2) We design an encoder–decoder architecture for UAV video captions with two configurations for encoding the video. The first models the spatiotemporal features directly from the input video, while the second is based on learning spatial-frame-based features and modeling the temporal features with a time attention module. The features generated by either configuration are fed directly to a GPT-like language decoder for caption generation.

The remainder of this paper is organized as follows. In Section 2, we describe the CapERA dataset including the annotation process. Next, in Section 3, the benchmark models are described in detail. Section 4 describes the experimental setup and presents the results. Section 5 presents a discussion of the results with several ablation studies. Finally, Section 6 concludes the paper.

2. CapERA Dataset

In this section, we present the CapERA dataset. We first introduce the pipeline for providing the captions in detail and present the overall statistics of the dataset. The CapERA dataset is an upgrade of the Event Recognition in Aerial Videos (ERA) dataset [11], which was released in 2020 for the task of event recognition. The original ERA dataset contains 2864 videos captured with UAV platforms related to 25 events, namely, post-earthquake, flood, fire, landslide, mudslide, traffic collision, traffic congestion, harvesting, ploughing, constructing, police chase, conflict, baseball, basketball, boating, cycling, running, soccer, swimming, car racing, party, concert, parade/protest, religious activity, and nonevent. Each video is five seconds long and recorded at 30 frames per second (fps) with a resolution of 640 × 640 pixels. The dataset contains 2864 videos split into 1473 and 1391 video samples for training and testing, respectively.

The ERA dataset has been collected from YouTube videos captured by various UAV platforms from various locations around the globe, hence featuring extensive diversity in terms of the subjects, environment, resolution, illumination, weather, background, camera motion, and flying attitude. Such unconstrained scenarios represent the complexity of the real world more closely, which is crucial for developing robust surveillance and analyzing models for videos.

To address the lack of available datasets for captioning aerial videos, we propose the CapERA dataset, which is built by providing captions for each video in the ERA dataset. To construct the annotation, we ask a human annotator to provide a concise description for each video. The annotation is written to include all information present in the clip such as the main event, object, place, action, numbers, and time. The total time of the manual annotation process is about 36 h. In the CapERA dataset, we provide five captions for each video to take into account as much as possible the variation in describing the same video. However, collecting five captions is resourcefully expensive. Therefore, caption augmentation is employed to alleviate the heavy workload of annotation. We specifically employ two automatic augmentation approaches: back-translation and paraphrasing. In both approaches, the manually given caption is used to guide the automatic generation of the augmented captions. Figure 1 presents a flowchart of the caption generation process.

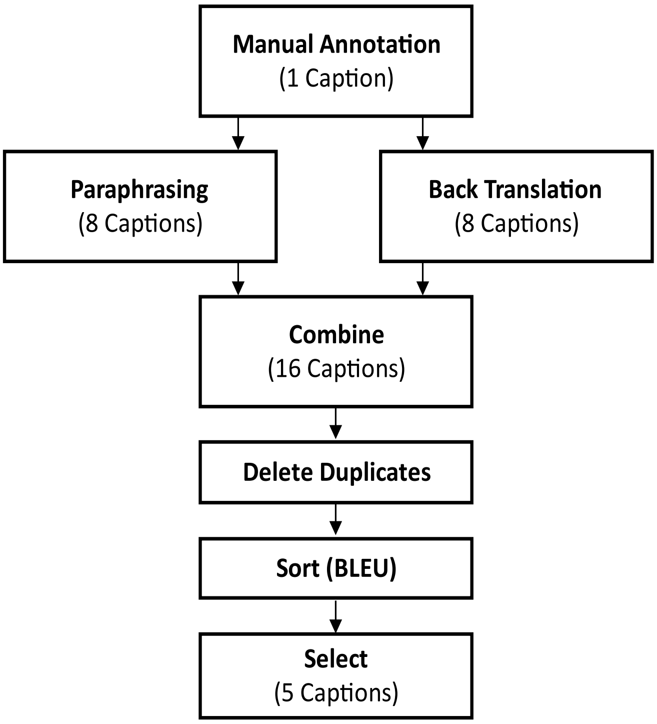
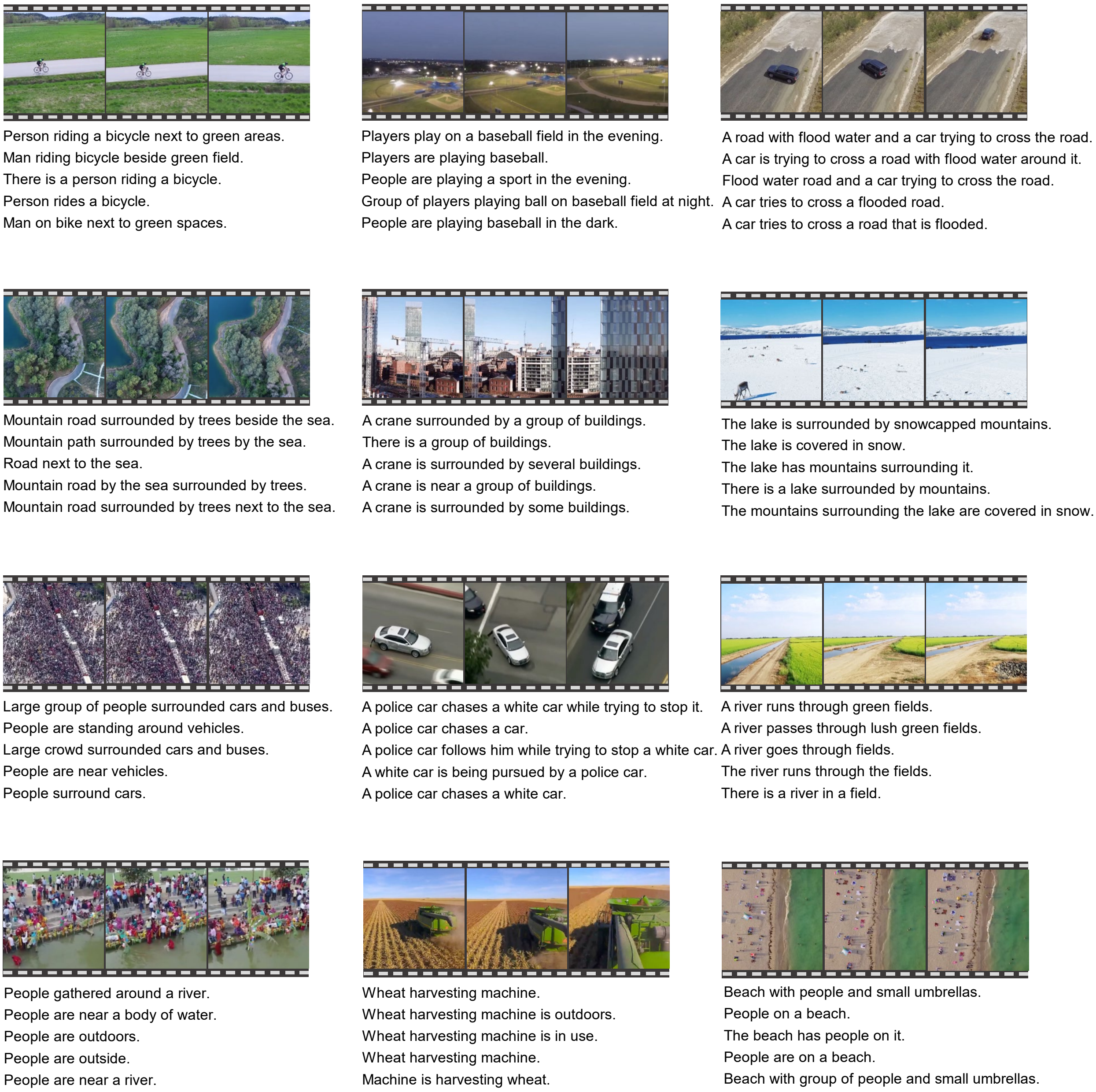


Figure 1. Pipeline for constructing CapERA dataset.

In the backtranslation, the manual caption is first translated into eight different common languages, namely, Arabic, French, Turkish, Russian, German, Spanish, Simplified Chinese, and Hindi. Then, all the translated captions are translated back into English. This step generates eight captions with similar meaning but expressed differently. In the paraphrasing approach, a deep paraphrasing model [21] is used to generate eight captions that express the manual caption in different words without altering the meaning. Then, the captions generated by the two augmentation methods are put together and the redundant captions are discarded. Next, all captions are sorted in ascending order based on the BiLingual Evaluation Understudy (BLEU) score [22]. Specifically, we take each caption one by one and compare it against all the other captions including the manual one. This will order all captions based on the N-gram similarity and retain the most different captions compared to the other captions on the top. Finally, the top four captions are taken, in addition to the original human-annotated caption for constructing the final dataset. Ultimately, we obtain a total of 14,320 captions, where each video is associated with five captions.

Figure 2 shows three frames of sample videos from the dataset along with their corresponding captions. The first caption is human-annotated, while the rest are automatically generated. In addition, Table 1 shows some statistics of the dataset. Similar to the original ERA dataset, CapERA contains 2864 videos, 1473 in the training set, and 1319 in the test set. The total number of captions in the dataset is 14,320 with an average length of 9.67 words. The size of the vocabulary is 1890, which means that the captions contain a diverse variety of words.



Person riding a bicycle next to green areas.
Man riding bicycle beside green field.
There is a person riding a bicycle.
Person rides a bicycle.
Man on bike next to green spaces.

Players play on a baseball field in the evening.
Players are playing baseball.
People are playing a sport in the evening.
Group of players playing ball on baseball field at night.
People are playing baseball in the dark.

A road with flood water and a car trying to cross the road.
A car is trying to cross a road with flood water around it.
Flood water road and a car trying to cross the road.
A car tries to cross a flooded road.
A car tries to cross a road that is flooded.

Mountain road surrounded by trees beside the sea.
Mountain path surrounded by trees by the sea.
Road next to the sea.
Mountain road by the sea surrounded by trees.
Mountain road surrounded by trees next to the sea.

A crane surrounded by a group of buildings.
There is a group of buildings.
A crane is surrounded by several buildings.
A crane is near a group of buildings.
A crane is surrounded by some buildings.

The lake is surrounded by snowcapped mountains.
The lake is covered in snow.
The lake has mountains surrounding it.
There is a lake surrounded by mountains.
The mountains surrounding the lake are covered in snow.

Large group of people surrounded cars and buses.
People are standing around vehicles.
Large crowd surrounded cars and buses.
People are near vehicles.
People surround cars.

A police car chases a white car while trying to stop it.
A police car chases a car.
A police car follows him while trying to stop a white car.
A white car is being pursued by a police car.
A police car chases a white car.

A river runs through green fields.
A river passes through lush green fields.
A river goes through fields.
The river runs through the fields.
There is a river in a field.

People gathered around a river.
People are near a body of water.
People are outdoors.
People are outside.
People are near a river.

Wheat harvesting machine.
Wheat harvesting machine is outdoors.
Wheat harvesting machine is in use.
Wheat harvesting machine.
Machine is harvesting wheat.

Beach with people and small umbrellas.
People on a beach.
The beach has people on it.
People are on a beach.
Beach with group of people and small umbrellas.

Figure 2. Samples from CapERA dataset. Each video is annotated with five captions. The first caption is manually annotated by a human, while the rest are automatically generated.

Table 1. Statistics from CapERA dataset.

	Video	Caption	Avg. Caption Length	Vocabulary Size
CapERA	2864	14,320	9.67	1890
Train	1473	7365	9.93	1283
Test	1319	6955	9.39	1471

3. Description of the Proposed Captioning Models

Let $X \in \mathbb{R}^{T \times H \times W \times 3}$ be a video from the CapERA dataset with T sampled frames each of dimension $H \times W \times 3$, and y is the corresponding caption of the video. We propose two configurations for mapping the input video into the sequence of words of the caption y. In the first configuration, which is described in Section 3.1, frames are evenly extracted from the input video and encoded independently into features using an image encoder. The frame-level features are fused using concatenation. Then, a temporal attention module is added to consider the temporal dynamics between features derived from the video frames. In Section 3.2, we describe the second configuration in which the video is encoded directly using a video encoder that employs spatial and temporal attention alternatively to model the dependencies within the video. Section 3.3 describes the decoding stage of both configurations, in which a language decoder plays the role of the captioner, which uses the visual feature representation from the whole video to generate the caption y.

3.1. Frames Encoder with Temporal Attention

Figure 3 shows the architecture of the first configuration of the captioning model. In this configuration, T frames are uniformly extracted from the input video $X = \{X_1, X_2, \ldots, X_T\}$, where each frame $X_t \in \mathbb{R}^{H \times W \times 3}$ is considered as an independent RGB image, and $t = 1, \ldots, T$ denotes the index of the frame in the video.

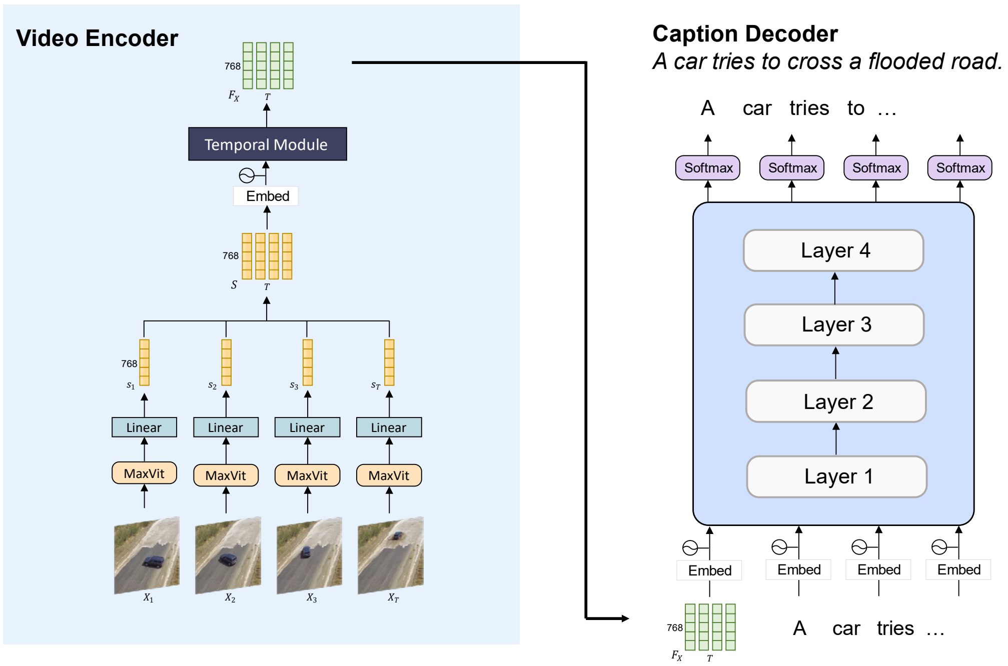


Figure 3. An illustration of the first proposed captioning model. The figure illustrates an example of sampling four frames from a video (T = 4) and extracting frame-level features from each frame. The frames' features are concatenated together and passed to an attention module to consider the temporal information.

We use MaxViT [23] as a backbone model to extract features from the frames. However, any other vision-model can be used as a backbone. The main motivation for choosing MaxViT is that it is a hybrid CNN-Transformer architecture that benefits from both the inductive bias of the convolution and the global feature learning of the attention mechanism. MaxViT applies a combination of convolution, local window attention, and global sparse attention to capture both local and global features from the frame. Specifically, the MaxViT backbone starts with a few convolutional layers, followed by multiple multi-axis self-attention (Max-SA) blocks. This block has a MBConv layer [24], a block local attention layer, and a grid global attention layer. These layers work together to allow global–local spatial interactions with a linear complexity.

For a frame X_t, the spatial feature is obtained by propagating the frame into a MaxViT encoder. We specifically utilize the output of the global average pooling layer. Next, we use a fully connected layer with a GELU activation function [25] to obtain the feature representation $s_t \in \mathbb{R}^{1\times 768}$:

$$s_t = GELU(Linear(MaxViT(X_t))) \tag{1}$$

For a video $X = \{X_1, X_2, \ldots, X_T\}$ with T frames, $S \in \mathbb{R}^{1\times 768\times T}$ represents the global representation of the whole video obtained by concatenating the spatial visual tokens for each frame in the video, where T represents the length of the visual sequences and 768 represents the feature dimension:

$$S = Concat(s_1, s_2, \ldots, s_T) \tag{2}$$

In addition to modeling spatial features within each frame, the temporal information is essential to devise the contextual relationship between frames and, hence, to describe the dynamics in the video when the caption is generated. To model the temporal aspect, a transformer layer with a self-attention module is added on top of the frame backbone. First, following the process used in [26,27], the obtained frame-level representations are embedded by a linear layer $E \in \mathbb{R}^{768\times 768}$. Then, a global learnable token $s_{class} \in \mathbb{R}^{1\times 768}$ is concatenated with the output representation. The positional information of each frame in the video $E_{pos} \in \mathbb{R}^{(T+1)\times 768}$ is added to it to capture the sequential information. The resulting sequence z is given by this equation:

$$z = [s_{class};\ s_1E; s_2E; \ldots;\ s_TE] + E_{pos} \tag{3}$$

This sequence is passed to a one-layer transformer encoder with multi-head self-attention (MHSA) and multi-layer perceptron (MLP) blocks. The latter block consists of two linear layers with a GeLU activation between them. Each of the two blocks of the encoder employs residual skip connections and is preceded by a normalization layer (LN) [25], as expressed by these equations:

$$z' = MHSA(LN(z)) + z \tag{4}$$

$$F_X = MLP(LN(z')) + z' \tag{5}$$

The temporal dependencies between frames are extracted by the MHSA module, which consists of multiple attention heads. First, three matrices—the query Q, key K, and value V—are generated from the representation z using three learnable layers W_Q, W_K, and $W_V \in \mathbb{R}^{768\times 768}$, such that $Q = W_Q z$, $K = W_K z$, and $V = W_V z$. Then, the attention weights for a single head are computed by the following equation:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{768}}\right)V \tag{6}$$

The results of all the attention heads are concatenated together and then projected through a feed-forward layer. Then, the output is passed to the MLP block (Equation (5)). Finally, the output of the MLP is considered as the feature representation $F_X \in \mathbb{R}^{1\times 768\times T}$ of the entire video clip X. This representation is supplied as an input to the language decoder, which decodes it into a caption.

3.2. Video Encoder with Spatial and Temporal Attention

The architecture of the second configuration is illustrated in Figure 4. In this model, the video is encoded directly using a pre-trained video encoder known as TimeSformer, which is a vision model proposed for video understanding [28]. TimeSformer utilizes the self-attention mechanism for capturing long-term dependencies. This idea was first proposed for sequence-to-sequence machine translation [27], and recently adopted for image and video understanding tasks.

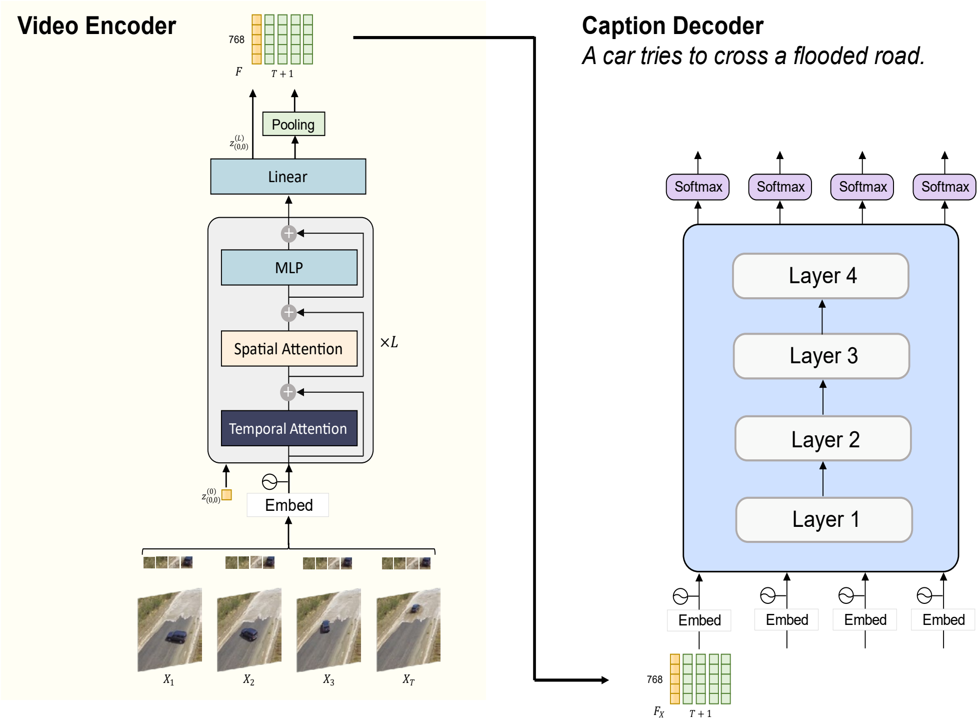


Figure 4. An illustration of the second configuration of the proposed captioning framework for CapERA dataset. The figure illustrates the example of sampling four frames from a video (T = 4). These frames are embedded and augmented with positional information and then encoded using a video encoder that employs spatial and temporal attention.

Given a video $X = \{X_1, X_2, \ldots, X_T\}$, each frame X_t is represented by N non-intersecting patches of size $P \times P$, where $N = H \times \frac{W}{P2}$. Then, the patches are flattened and reshaped as a vector $x_{(p,\,t)} \in \mathbb{R}^{3P^2}$, where $p = 1, \ldots, N$ denotes the position of the patch in frame t, and $t = 1, \ldots, T$ denotes the index of the frame in the video. Given the sequence of patch representation, TimeSformer applies divided self-attention mechanism to the time and the space dimension separately to capture the content of the video. First, each patch $x_{(p,\,t)}$ is transformed into visual tokens of the model dimension D by a linear embedding layer $E \in \mathbb{R}^{D \times 3P^2}$:

$$z_{(p,\,t)}^{(0)} = E\left(x_{(p,\,t)}\right) + E_{(p,\,t)}^{pos} \tag{7}$$

where $z_{(p,\,t)}^{(0)} \in \mathbb{R}^D$, and $E_{(p,\,t)}^{pos} \in \mathbb{R}^D$ is a learnable spatial–temporal positional embedding added to each patch representation to preserve its positional information within the space and time dimensions. In addition, a special learnable token $z_{(0,\,0)}^{(0)} \in \mathbb{R}^D$ is prepended to the sequence. This token will serve as the global representation of the input video, which establishes the interaction between the spatial and temporal features.

The sequence is then passed to the video encoder with L consecutive layers. In each layer $l = 1, \ldots, L$, given the sequence $z_{(p,\,t)}^{(l-1)}$, TimeSformer utilizes factorized space–time attention, which employs a separate MHSA block for time and space to simplify the computation. Specifically, each layer l has a series of temporal MHSA, spatial MHSA, and MLP blocks. Each block uses a normalization layer (LN) [25] and skip connections. Formally, the factorized space–time attention is described as follows:

$$z_{(p,\,t)}^{(l)time} = MHSA_{time}\left(LN\left(z_{(p,\,t)}^{(l-1)}\right)\right) + z_{(p,\,t)}^{(l-1)} \tag{8}$$

$$z_{(p,\,t)}^{(l)space} = MHSA_{space}\left(LN\left(z_{(p,\,t)}^{(l)time}\right)\right) + z_{(p,\,t)}^{(l)time} \tag{9}$$

$$z_{(p,\,t)}^{(l)} = MLP\left(LN\left(z_{(p,\,t)}^{(l)space}\right)\right) + z_{(p,\,t)}^{(l)space} \tag{10}$$

Analogs to the MHSA block are used in the first configuration. Both the temporal attention and the spatial attention are formulated as the scaled dot-product attention defined in Equation (6). First, the $MHSA_{time}$ takes three sets of input elements, the query Q, key K, and value V computed from $z_{(p,\,t)}^{(l-1)}$ using learnable embedding matrices $W_Q^{(l)}$, $W_K^{(l)}$, and $W_V^{(l)} \in \mathbb{R}^{D\times D}$, respectively, such that:

$$Q = W_Q^{(l)} z_{(p,\,t)}^{(l-1)},\; K = W_K^{(l)} z_{(p,\,t)}^{(l-1)},\; V = W_V^{(l)} z_{(p,\,t)}^{(l-1)},\, l = 1,\, \ldots,\, L \tag{11}$$

$MHSA_{time}$ is used to detect the relationship between one patch and patches on the same position across different time frames. Thus, K and V are sampled across the different frames of the video. The next $MHSA_{space}$ block uses spatial attention to identify relationships between different patches located within the same frame. For this, Q, K, and V vectors are computed from $z_{(p,\,t)}^{(l)time}$ using embedding matrices $W_Q'^{(l)}$, $W_K'^{(l)}$, and $W_V'^{(l)}$, respectively, where K and V are sampled from the same frame.

The output representation $z_{(p,\,t)}^{(l)space}$ is then passed to the MLP block, which consists of two linear layers separated by a GELU non-linearity activation. We use the representation $z_{(0,\,0)}^{(L)} \in \mathbb{R}^{1\times 768}$ obtained from the output of the final layer L. This representation is considered as the global token of the video that jointly encodes the spatial and the temporal information. In addition, to fully utilize the rich information of the video, extra tokens are obtained by applying global average pooling on the spatial feature vector of each frame obtained by the last layer of the TimeSformer:

$$s_t = \frac{1}{N}\sum\nolimits_{p=1}^{N} z_{(p,t)}^{L} \tag{12}$$

where $s_t \in \mathbb{R}^{1\times 768}$. Thus, T tokens are obtained from the video X. These tokens are concatenated together with the global token $z_{(0,\,0)}^{(L)}$ generated by the TimeSformer encoder. The final representation $F_X \in \mathbb{R}^{1\times 768\times (T+1)}$ is considered as the spatiotemporal feature representation of the input video X, which is passed to the language decoder to generate the caption:

$$F_X = Concat\left(z_{(0,\,0)}^{(L)},\, s_1,\, \ldots, s_T\right) \tag{13}$$

3.3. Language Decoder

The goal of the language model is to generate a natural language sentence that best describes the given UAV video based on the video feature representation F_X. The decoder is implemented as a masked transformer decoder with a GPT-like architecture [29]. The decoder produces the caption word by word in an auto-regressive manner where the prediction of the next word is conditioned on a sequence of the prior words. To learn the decoder to generate captions from the UAV video, we take the spatiotemporal representation of the video from one of the two video configurations and use it as a context for generating the caption.

To train the decoder, the caption y is first split into a sequence of tokens $(w_1,\ w_2,\ \ldots\ w_m)$, where m is the maximum length of the caption tokens. The tokens are concatenated with two special tokens [CLS] and [SEP] to mark the beginning and the end of the sequence. Then, each token is embedded with a word embedding layer to embed it into a vector in a high-dimensional space. In addition, a positional embedding is added to supply the sequence with information about the position of each token.

As shown in Figure 5, the language decoder consists of four layers, and each layer has MHSA and MLP blocks in addition to two normalization layers and skip connections. The MHSA block is leveraged to capture the dependencies between the textual tokens of the captions as well as the given spatiotemporal representation of the video. The decoder specifically employs a masked MSA block that learns the dependencies within the caption tokens without taking into account future tokens. This helps the model predict the next word based on the sequence of previous tokens only. The decoder is followed by a softmax layer that turns the output of the decoder into probabilities, where the word with the highest probability is chosen to be the next word in the caption.

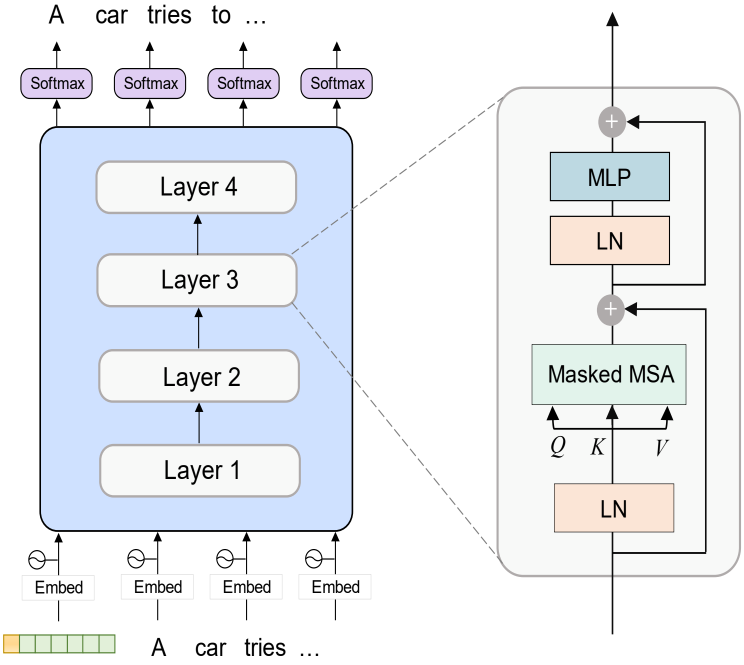


Figure 5. The architecture of the language decoder.

The full model in both configurations is trained by finetuning the parameters of the language decoder by optimizing it on the auto-regressive language modeling objective function conditioned on the spatiotemporal representation of the video from one of the two video modeling configurations. Formally, the objective function is defined as:

$$\mathcal{L} = -\sum_{i=1}^{m} logP(w_i|w_1 \ldots w_{m-1},\ \ F_X) \tag{14}$$

4. Results

In this section, we present the experimental results of the video captioning models on the CapERA dataset. In Section 4.1, we describe the training details and the experimental settings. Evaluation metrics are introduced in Section 4.2, and we report the results in Section 4.3.

4.1. Experimental Settings

For the frame encoder in the first configuration, we use a MaxViT-S model pretrained on ImageNet-21K and finetuned on ImageNet-1K. The model has a size of 69 M and uses an input frame of size 224 × 224 and a window size and grid size of seven. The temporal module closely follows the architecture of a one-layer vision transformer model [26] with 8 attention heads, a dimension of 768, and an MLP of size 512.

In the second configuration, we employ a TimeSformer as a backbone. Specifically, we use a TimeSformer model with $l = 12$ encoder layers, each with 12 attention heads. The embedding dimension in the model is $D = 768$, and the MLP is 3072. We use a TimeSformer model pretrained on the howTo100M dataset with divided space–time attention. For preprocessing, each sampled frame is resized to 224×224, and the frame-level patch P has a size of 16×16 pixels with a sequence length of 196.

The language decoder for both models uses the Bert-base-uncased tokenizer to tokenize the caption. The sequence can take up to 120 token lengths with a vocabulary size of 50,257. The number of decoder layers is four, and it has 768-hidden-feature representation and 12 attention heads. We train the model by finetuning the language decoder only. The model is trained for 50 epochs on the CapERA dataset with a mini batch of size 10. We optimize the model with Adam optimizer [30] and set the learning rate to 3×10^{-4}. We implement all the experiments with the PyTorch framework using a PC workstation with a Core(TM) i9-7920X CPU, GeForce RTX 1080 Ti 11 GB, and 64-GB RAM.

4.2. Evaluation Metrics

To quantitatively evaluate the benchmark models in generating captions, several evaluation metrics are utilized. The first metric is the BLEU [22] score, which is an automatic evaluation metric that measures the matching n-grams (i.e., consecutive words) between the predicted caption and the ground-truth caption. The value of n is selected to be between 1 and 4. The second is the Metric for Evaluation of Translation with Explicit Ordering (METEOR) [31], which is computed by building an alignment between the words of the generated caption and the ground-truth caption. METEOR computes the F1 score on the matching results, which combines the precision and recall. In addition, we compute the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [32], which measures the F-measure of the longest common subsequence between the generated caption and ground truth captions. Finally, we provide the results in terms of the Consensus-based Image Description Evaluation (CIDEr) score [33] which sums the weighted cosine similarity between n-grams found in the generated caption and ground truth captions. CIDEr applies term frequency-inverse document frequency (TF-IDF) weights for every n-gram so that the frequently used words are given less weight.

4.3. Results

The experimental results of the first configuration are presented in Table 2. In this table, we show the results of the model with a variable number of frames. In addition, we examine the effectiveness of the temporal attention module placed on top of the encoded per-frame spatial features and compare it with simply concatenating the features without considering the temporal information. We also measure the actual training and testing time for all experiments.

Table 2. Results of the first proposed captioning model.

		BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGE_L	CIDEr	Train Time	Test Time
Without Temporal Module	4 Frames	47.76	34.61	26.75	20.50	19.45	41.34	49.17	15.94	0.63
	8 Frames	47.53	34.44	26.79	20.82	19.50	41.50	48.42	19.02	0.67
	16 Frames	46.34	33.43	25.90	19.87	18.53	41.29	44.92	25.44	0.74
	32 Frames	46.94	34.11	26.52	20.43	18.63	41.91	48.09	38.10	0.86
With Temporal Module	4 Frames	48.31	35.20	27.20	20.99	20.09	41.93	53.09	18.81	0.67
	8 Frames	49.13	35.92	27.95	21.74	19.82	42.79	53.43	21.67	0.71
	16 Frames	48.03	34.90	27.08	21.07	19.48	42.04	48.36	27.87	0.77
	32 Frames	47.92	34.82	27.01	20.91	19.36	41.88	50.29	41.11	0.89

The results show that using the temporal attention module yields an improvement of the performance in all metrics regardless of the number of sampled frames. This proves that capturing the temporal information is a key factor for understanding the content of the video, hence improving the quality of the generated description. However, adding the temporal information increases the time required to train the model and the time to generate the caption. As shown in the table, adding the temporal attention increases the training time by an average of 2.74 h and the testing time by 0.03 h.

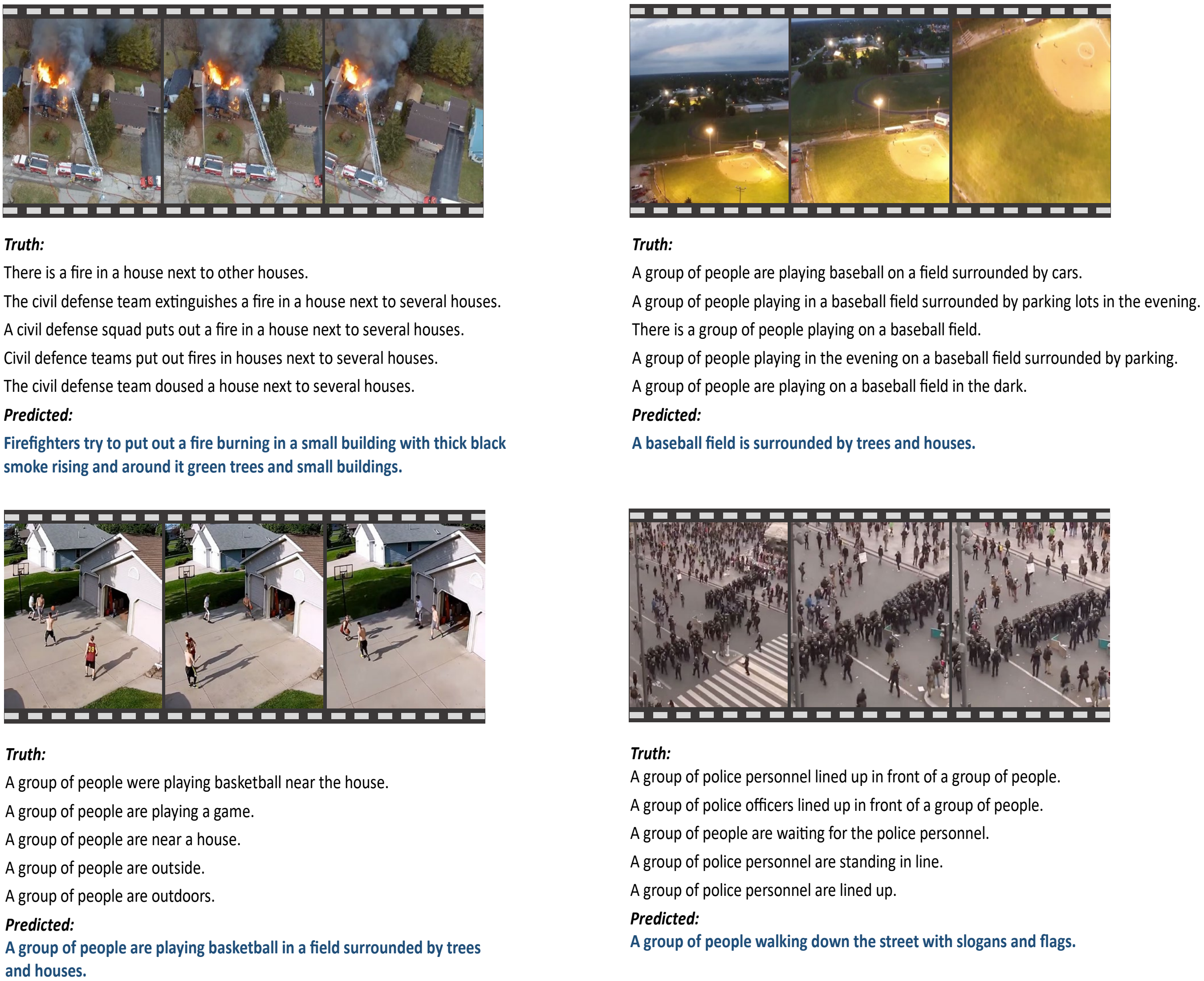
We further investigate how the number of input frames affects the performance of the model by evaluating the model with a variant number of sampled frames $T = \{4, 8, 16, 32\}$. The results show that for the model without the temporal module, fewer frames are generally better. Specifically, the best BLEU1, BLEU2, and CIDEr scores are achieved with only four frames, and sampling eight frames shows the best scores in BLEU3, BLEU4, and METEOR. For the model with temporal attention, the performance of the model gradually improves as the number of the sampled frames increases from four to eight in almost all metrics, except the METEOR where the best score is achieved with only four frames. However, the performance degrades with more input frames. This observation demonstrates that four frames are not sufficient to fully capture the temporal context, but more than eight input frames degrade the performance, which can be caused by temporal information redundancy.

Table 3 shows the results of the second proposed CapERA captioning model with varying numbers of sampled frames. The results of this model also show that sampling eight frames demonstrates the best results in all metrics, except the METEOR where four frames achieve slightly better results. By comparing the results of Tables 2 and 3, we see that for the same number of frames, the second model, which uses TimeSformer to jointly capture temporal and spatial information, provides better results in all metrics compared to the model that fuses the temporal information in a later stage. However, the second model takes a longer time for training and generating the description for the videos. We can see that the second model increases the training time by an average of 3.01 h and the testing time by three minutes. We also see that for more frames sampled from the video, the model takes a longer training and testing time to learn and generate results.

Table 3. Results of the second proposed model.

	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGE_L	CIDEr	Train Time	Test Time
4 Frames	49.82	36.66	28.64	22.20	21.41	43.79	59.40	21.44	0.72
8 Frames	50.43	37.26	29.24	22.90	21.16	43.90	60.42	25.12	0.76
16 Frames	50.10	36.75	28.48	22.01	20.97	43.47	58.55	30.89	0.82
32 Frames	50.13	36.92	28.78	22.30	21.17	43.93	59.31	44.35	0.94

In order to qualitatively evaluate the captioning model, Figure 6 shows examples of the generated caption of random videos from the test set of the CapERA dataset along with the five ground-truth captions. These examples are generated by the second configuration. From the first example of a fire event, we can notice that the caption generated by our model contains all the semantic information present in the video. We can also see that although the generated caption does not perfectly match the ground-truth captions, it expresses objects, attributes, and details in the video that are not present in the ground-truth captions such as ‘thick black smoke rising’ and ‘green trees’. The second example shows the predicted caption for a baseball field video. The caption perfectly describes the main event and the environment shown in the video, despite the model failing to detect the people playing in the baseball field, which is a quite challenging due to the small region they occupy in the frames. The third example shows an accurate caption that describes a video of a group of people playing basketball. The caption perfectly matches the ground-truth captions with more details such as ‘in a field surrounded by trees’. The last example shows the caption for a video from a conflict event. The generated caption provides an accurate and realistic description of the video. These results confirm that the model is effective in describing a complex UAV video due to its ability capture the spatiotemporal information.



Truth:

There is a fire in a house next to other houses.

The civil defense team extinguishes a fire in a house next to several houses.

A civil defense squad puts out a fire in a house next to several houses.

Civil defence teams put out fires in houses next to several houses.

The civil defense team doused a house next to several houses.

Predicted:

Firefighters try to put out a fire burning in a small building with thick black smoke rising and around it green trees and small buildings.

Truth:

A group of people are playing baseball on a field surrounded by cars.

A group of people playing in a baseball field surrounded by parking lots in the evening.

There is a group of people playing on a baseball field.

A group of people playing in the evening on a baseball field surrounded by parking.

A group of people are playing on a baseball field in the dark.

Predicted:

A baseball field is surrounded by trees and houses.

Truth:

A group of people were playing basketball near the house.

A group of people are playing a game.

A group of people are near a house.

A group of people are outside.

A group of people are outdoors.

Predicted:

A group of people are playing basketball in a field surrounded by trees and houses.

Truth:

A group of police personnel lined up in front of a group of people.

A group of police officers lined up in front of a group of people.

A group of people are waiting for the police personnel.

A group of police personnel are standing in line.

A group of police personnel are lined up.

Predicted:

A group of people walking down the street with slogans and flags.

Figure 6. Visualization of some generated captions by the second proposed model. The black-color texts indicate the five ground-truth captions, while the blue-color texts indicate the generated caption.

5. Discussion

In this section, we conduct a thorough ablation analysis of the model architecture and investigate different factors affecting the performance of the captioning model. Specifically, we study three main elements: the effect of changing the number of decoder layers, the improvements resulting from changing the video backbone model, and the effect of fixing and finetuning the TimeSformer backbone. In all experiments, unless otherwise mentioned, we use the second model pre-trained on howTo100M with fixed weights, with eight uniformly sampled frames per input video, and a language decoder with four layers.

5.1. The Effect of Changing the Number of Decoder Layers

In this experiment, we investigate the optimal number of layers in the language decoder that yields the best performance. Table 4 summarizes the experimental results of using a variant number of language decoder layers ranging from two to four on the CapERA dataset. From Table 4 it can be observed that the decoder with four layers achieves the best performance in all metrics except the METEOR and the ROUGE_L, where using two layers only shows a slightly better performance. However, increasing the number of decoder layers increases the complexity of the model and, hence, consumes more time for training and testing the model. As shown in the table, adding one layer increases the training time by around one hour, and the testing time by 1.2 min.

Table 4. Results of changing the number of decoder layers.

Number of Layers	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGE_L	CIDEr	Train Time	Test Time
2	49.93	36.60	28.43	21.93	21.39	44.06	59.88	22.92	0.70
3	49.62	36.67	28.61	22.20	21.20	43.86	58.92	24.02	0.72
4	50.43	37.26	29.24	22.90	21.16	43.90	60.42	25.12	0.74

5.2. The Effect of Changing the Video Backbone Model

The proposed CapERA dataset is composed of 2864 UAV videos only. Therefore, training the captioning model from scratch is difficult and more likely leads to overfitting problems. A common practice for learning visual representations for such relatively small datasets is to employ transfer learning by using a vision model pre-trained on a large-scale dataset. In this experiment, we aim to explore the effect of changing the dataset used for pre-training the video backbone model on the captioning results. We compare between a model pre-trained on howTo100M [34] and Kinetics600 [35,36] datasets. HowTo100M is a large-scale action recognition dataset composed of 136 million narrated web videos of humans performing different tasks, while the Kinetics600 dataset is a video human action classification dataset with 600 action classes. The results in Table 5 demonstrate that extracting features from a model pre-trained on either dataset is effective as the results do not show a significant difference. However, using a TimeSformer backbone pre-trained on the Kinetics600 dataset performs slightly better than the one pre-trained on HowTo100M in all metrics except the BLEU1. In addition, it requires less time for training and generating the captions.

Table 5. Results of changing the video backbone model.

Pretrained Model	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGE_L	CIDEr	Train Time	Test Time
HowTo100M	50.43	37.26	29.24	22.90	21.16	43.90	60.42	25.12	0.74
Kinetics600	50.37	37.35	29.41	23.08	21.65	44.16	61.05	23.13	0.72

5.3. The Effect of Fixing and Finetuning the TimeSformer Model

As mentioned in Section 5.2, it is often advantageous to employ transfer learning when the available dataset is small. Transfer learning can be implemented either by fixing the model's weights and extracting features from the fixed model or by finetuning the model on the specific dataset at hand. In this experiment, we compare the performance of the captioning model when the weights of the TimeSformer are fixed or finetuned. In Table 6, we observe that compared to the finetuned model, extracting features from the fixed model improves the performance of all metrics by 0.8–2% except CIDEr, which is improved by around 6%. We argue that due to the small size of the CapERA dataset, this dataset is not suitable for the model finetuning, and that the features learned by related datasets are sufficient to be applicable for the UAV captioning task.

Table 6. Results of fixing and finetuning the TimeSformer weights.

	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGE_L	CIDEr
Fixed model	50.43	37.26	29.24	22.90	21.16	43.90	60.42
Finetuned model	48.55	35.27	27.38	21.19	20.35	42.68	54.44

6. Conclusions

In this paper, we introduce a new caption dataset, termed CapERA, for generating textual descriptions from aerial videos. This dataset provides five captions for 2864 UAV videos. The CapERA dataset represents diverse scenarios in our daily life and thus supports different vision–language applications in the remote sensing video domain. Based on this dataset, we present two benchmarks for aerial video captioning. First, the visual features from the video are obtained by a video encoder or a frame-based encoder to generate rich features from the video. Afterward, these features are used as a context to guide a language decoder to generate the caption from the UAV video. We experimentally evaluate the performance of the proposed models over our CapERA dataset. We also conduct extensive ablation studies to evaluate the effectiveness of the proposed models. The experimental results demonstrate the challenge of generating captions for aerial videos.

Author Contributions: Formal analysis: L.B.; Methodology: Y.B., M.M.A.R. and F.M., writing—original draft preparation: Y.B., writing—review and editing: L.B., M.Z., software: Y.B., Investigation: M.M.A.R., Project administration: M.Z. and Funding acquisition: M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project no. (IFKSURG-2-585).

Acknowledgments: The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project no. (IFKSURG-2-585).

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

References

1. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Springer: Cham, Switzerland, 2015.
2. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
3. Bazi, Y.; Melgani, F. Convolutional SVM Networks for Object Detection in UAV Imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 3107–3118. [CrossRef]
4. Lyu, Y.; Vosselman, G.; Xia, G.; Yilmaz, A.; Yang, M.Y. UAVid: A Semantic Segmentation Dataset for UAV Imagery. *ISPRS J. Photogramm. Remote Sens.* **2020**, *165*, 108–119. [CrossRef]
5. Jin, P.; Mou, L.; Xia, G.-S.; Zhu, X.X. Anomaly Detection in Aerial Videos with Transformers. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–13. [CrossRef]
6. Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA; pp. 3974–3983.
7. Wu, X.; Li, W.; Hong, D.; Tao, R.; Du, Q. Deep Learning for UAV-Based Object Detection and Tracking: A Survey. *IEEE Geosci. Remote Sens. Mag.* **2022**, *10*, 91–124. [CrossRef]
8. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; Volume 11214, pp. 375–391. ISBN 978-3-030-01248-9.
9. Wang, J.; Teng, X.; Li, Z.; Yu, Q.; Bian, Y.; Wei, J. VSAI: A Multi-View Dataset for Vehicle Detection in Complex Scenarios Using Aerial Images. *Drones* **2022**, *6*, 161. [CrossRef]
10. Yang, M.-D.; Tseng, H.-H.; Hsu, Y.-C.; Yang, C.-Y.; Lai, M.-H.; Wu, D.-H. A UAV Open Dataset of Rice Paddies for Deep Learning Practice. *Remote Sens.* **2021**, *13*, 1358. [CrossRef]
11. Mou, L.; Hua, Y.; Jin, P.; Zhu, X.X. ERA: A Data Set and Deep Learning Benchmark for Event Recognition in Aerial Videos [Software and Data Sets]. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 125–133. [CrossRef]
12. Bashmal, L.; Bazi, Y.; Al Rahhal, M.M.; Alhichri, H.; Al Ajlan, N. UAV Image Multi-Labeling with Data-Efficient Transformers. *Appl. Sci.* **2021**, *11*, 3974. [CrossRef]
13. Alshehri, A.; Bazi, Y.; Ammour, N.; Almubarak, H.; Alajlan, N. Deep Attention Neural Network for Multi-Label Classification in Unmanned Aerial Vehicle Imagery. *IEEE Access* **2019**, *7*, 119873–119880. [CrossRef]
14. Zeggada, A.; Benbraika, S.; Melgani, F.; Mokhtari, Z. Multilabel Conditional Random Field Classification for UAV Images. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 399–403. [CrossRef]

15. Zeggada, A.; Melgani, F.; Bazi, Y. A Deep Learning Approach to UAV Image Multilabeling. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 694–698. [CrossRef]
16. Moranduzzo, T.; Melgani, F.; Mekhalfi, M.L.; Bazi, Y.; Alajlan, N. Multiclass Coarse Analysis for UAV Imagery. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 6394–6406. [CrossRef]
17. Bang, S.; Kim, H. Context-Based Information Generation for Managing UAV-Acquired Data Using Image Captioning. *Autom. Constr.* **2020**, *112*, 103116. [CrossRef]
18. Bashmal, L.; Bazi, Y.; Alajlan, N. Space Time Attention Transformer for Non-Event Detection in UAV Videos. In Proceedings of the IGARSS 2022—2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 17–22 July 2022; IEEE: Piscataway, NJ, USA; pp. 1920–1923.
19. Jin, P.; Mou, L.; Hua, Y.; Xia, G.-S.; Zhu, X.X. FuTH-Net: Fusing Temporal Relations and Holistic Features for Aerial Video Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–13. [CrossRef]
20. Hoxha, G.; Melgani, F. A Novel SVM-Based Decoder for Remote Sensing Image Captioning. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
21. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P.J. PEGASUS: Pre-Training with Extracted Gap-Sentences for Abstractive Summarization. In Proceedings of the 37th International Conference on Machine Learning, ICML 2020, Online, 13–18 July 2020.
22. 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 on Association for Computational Linguistics—ACL '02, Philadelphia, PA, USA; Association for Computational Linguistics: Cedarville, OH, USA, 2001; p. 311.
23. Tu, Z.; Talebi, H.; Zhang, H.; Yang, F.; Milanfar, P.; Bovik, A.; Li, Y. MaxViT: Multi-Axis Vision Transformer. In *Proceedings of the Computer Vision—ECCV 2022*; Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., Eds.; Springer Nature Switzerland; Cham, Switzerland.
24. Hendrycks, D.; Gimpel, K. Gaussian Error Linear Units (GELUs). *arXiv* **2020**, arXiv:1606.08415.
25. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
26. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
28. Bertasius, G.; Wang, H.; Torresani, L. Is Space-Time Attention All You Need for Video Understanding? *arXiv* **2021**, arXiv:2102.05095.
29. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models Are Unsupervised Multitask Learners. *Open AI. Blog* **2019**, *1*, 9.
30. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2017**, arXiv:1412.6980.
31. Denkowski, M.; Lavie, A. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In Proceedings of the Ninth Workshop on Statistical Machine Translation, Baltimore, MD, USA, 26–27 June 2014; Association for Computational Linguistics: Cedarville, OH, USA, 2014; pp. 376–380.
32. Lin, C.-Y. ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings of the Workshop Text Summarization Branches out, Barcelona, Spain, 10–17 July 2004; p. 10.
33. Vedantam, R.; Zitnick, C.L.; Parikh, D. CIDEr: Consensus-Based Image Description Evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, 7–12 June 2015.
34. Miech, A.; Zhukov, D.; Alayrac, J.-B.; Tapaswi, M.; Laptev, I.; Sivic, J. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: Piscataway, NJ, USA; pp. 2630–2640.
35. Carreira, J.; Noland, E.; Banki-Horvath, A.; Hillier, C.; Zisserman, A. A Short Note about Kinetics-600. *arXiv* **2018**, arXiv:1808.01340v1.
36. Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. The Kinetics Human Action Video Dataset. *arXiv* **2017**, arXiv:1705.06950.

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