Learning deep spatiotemporal features for video captioning

Eleftherios Daskalakis[a], Maria Tzelepi[a], Anastasios Tefas[a]

[a]*Aristotle University of Thessaloniki, Department of Informatics, Thessaloniki, Greece*

ABSTRACT

In this paper, we propose a novel automatic video captioning system which translates videos to sentences, utilizing a deep neural network that is composed of three building parts of convolutional and recurrent structure. That is, the first subnetwork operates as feature extractor of single frames. The second subnetwork is a three-stream network, capable of capturing spatial semantic information in the first stream, temporal semantic information in the second stream, and global video concept information in the third stream. The third subnetwork generates relevant textual captions using as input the spatiotemporal features of the second subnetwork. The experimental validation indicates the effectiveness of the proposed model, achieving superior performance over competitive methods.

1. Introduction

Over the past few years, deep learning algorithms, [5], and principally the deep Convolutional Neural Network (CNN) architectures, [16], have been established as one of the most promising avenues of research in computer vision area, providing outstanding performance in a plethora of computer vision tasks, such as image classification, [14], and face recognition, [24]. Furthermore, recent works, [11, 28], indicate that by combining convolutional and recurrent neural networks comprising of Long Short-Term Memory (LSTM) modules, [9], hybrid models can be trained that are capable of translating image or video content to regular text, carving new routes in computer vision research.

The task of automatic video captioning constitutes a fundamental challenge in computer vision. Among the most beneficial applications is the assistance of the visually impaired. Furthermore, considering the large amount of video data uploaded every day on popular sites such as YouTube, as well as the amount of videos that are poorly tagged, automatic video captioning can aid indexing by providing more accurate search terms.

In this paper, we propose a novel automatic video captioning system which translates videos to sentences, using a deep neural network that is composed of three subnetworks. The first one operates as feature extractor of single frames. The second one is a three-stream network, capable of capturing different kinds of information. That is, it captures spatial semantic information in the

first stream, temporal semantic information in the second stream, and global video concept information in the third stream. Finally, the third subnetwork generates relevant textual captions using as input the spatiotemporal features of the second subnetwork.

We built upon the work of [11] and [28] which takes advantage of a CNN followed by a Recurrent Neural Network (RNN) composed of LSTM modules. These models are designed for static image captioning tasks, however in this work we focus on generating sentences of text descriptions to video snippets, providing results on a new, larger and more challenging video dataset with paired natural language descriptions, [29, 18].

The remainder of the manuscript is structured as follows. Section 2 discusses prior work. The proposed video captioning system is described in detail in Section 3. The utilized datasets are presented in Section 4. Experimental results are provided in Section 5. Finally, conclusions are drawn in Section 6.

2. Prior work

In this Section we survey previous image and video captioning works. Image captioning flourished over the recent years, [7, 11, 31, 28, 12]. Briefly, in image captioning, a recurrent network is trained upon single feature vectors that are produced by a preceding convolutional network, translating all visual information in the input image, following a one-to-many translation approach, as in [11] and [28].

Subsequently, motivated by the success in image captioning, research attention naturally focused on generating sentences that describe videos. The task of video captioning bears an additional challenge, that is to exploit both spatial and temporal information, since videos contain both spatial information about a depicted scene as well as temporal information.

In [27], a model composed of a CNN followed by a RNN is proposed. A video is provided as input and one in ten frames is sampled and forwarded through a CNN, which in turn produces a feature vector from the activations of the last fully connected layer, that encodes the spatial content of the sampled frame. Mean pooling is performed over all the feature vectors of the frames, in order to produce a single feature vector for the entire video, which is forwarded on to the RNN which in turn provides the text caption. The major flaw of this model is that only global information of the frames is exploited, disregarding the temporal information hidden in the sequence of subsequent frames, as well as the localized spatial information in each scene.

Subsequently, in [26], a model that aims to exploit temporal information, and uses an LSTM to convert a sequence of video frames into a sequence of words, is proposed. The authors consider the task of video captioning analogous to machine translation, where a sequence of words in the input language is translated into a sequence of words in the output language. However, we argue that this does not depict sufficiently the problem since when translating languages we have a limited vocabulary of specific words of the input language and a limited set of word combinations, while here we encounter sequences of high dimensional feature vectors, which can greatly vary, preventing the model from generalizing.

In [30] a model that exploits the temporal structure in videos, utilizing a temporal attention mechanism is proposed. This mechanism allows the decoder to selectively focus on a subset of frames, similarly to our model, however it considers local short variations in the time domain.

In [15], the authors proposed a framework to automatically generate descriptions for video clips, by applying a temporal attention mechanism to the sequence-to-sequence LSTM model.

Subsequently, from a different viewpoint the authors in [32] focus on the sentence decoder, and propose a hierarchical model containing a sentence and a paragraph generator: short sentences are produced by a Gated Recurrent Unit (GRU) [4] layer conditioned on video features, while another recurrent layer is in charge of generating paragraphs by combining sentence vectors and contextual information. The paragraph generator can therefore capture inter-sentence dependencies and generate a sequence of related and consecutive sentences.

In [19] a 2-D/3-D CNN is utilized to extract visual features of selected video frames/clips and the video representations are produced by mean pooling over these visual features. The attributes from two sources are fused and leveraged for enhancing video captioning. Then, a LSTM for generating video description is learnt by feeding into both video representations and semantic attributes mined from images and videos. In addition, to better leverage the attributes from two sources, a transfer unit is devised to dynamically balance the influence in between given the input word and the hidden state in LSTM.

Finally, in [1] the authors propose a recurrent video encoder which takes as input a sequence of visual features and outputs a sequence of vectors as the representation for the whole video. In their encoder, the connectivity schema of the layer varies with respect to both the current input and the hidden state, so it is thought as an activation instead of being a non learnable hyperparameter. To this aim, a time boundary-aware recurrent cell is defined, which can modify the layer connectivity through time. This ensures that the input data following a time boundary are not influenced by those seen before the boundary, and generates a hierarchical representation of the video in which each chunk is composed by homogeneous frames.

3. Proposed video captioning model

In this paper we propose a complete video captioning system, that is composed of three building parts of convolutional and recurrent structure. The first subnetwork is a CNN model, which operates as feature extractor of single frames. Towards this end, we utilize a common CNN pretrained model, that is the VGG-16, [22] pretrained on ILSVRC-2014 [21] for classifying 1.3 million images to 1000 ImageNet classes. The second subnetwork, is a three-stream network capable of capturing various kinds of information, while the third subnetwork is a RNN consisting of LSTM modules, that aims to generate relevant textual captions utilizing features obtained from the second subnetwork.

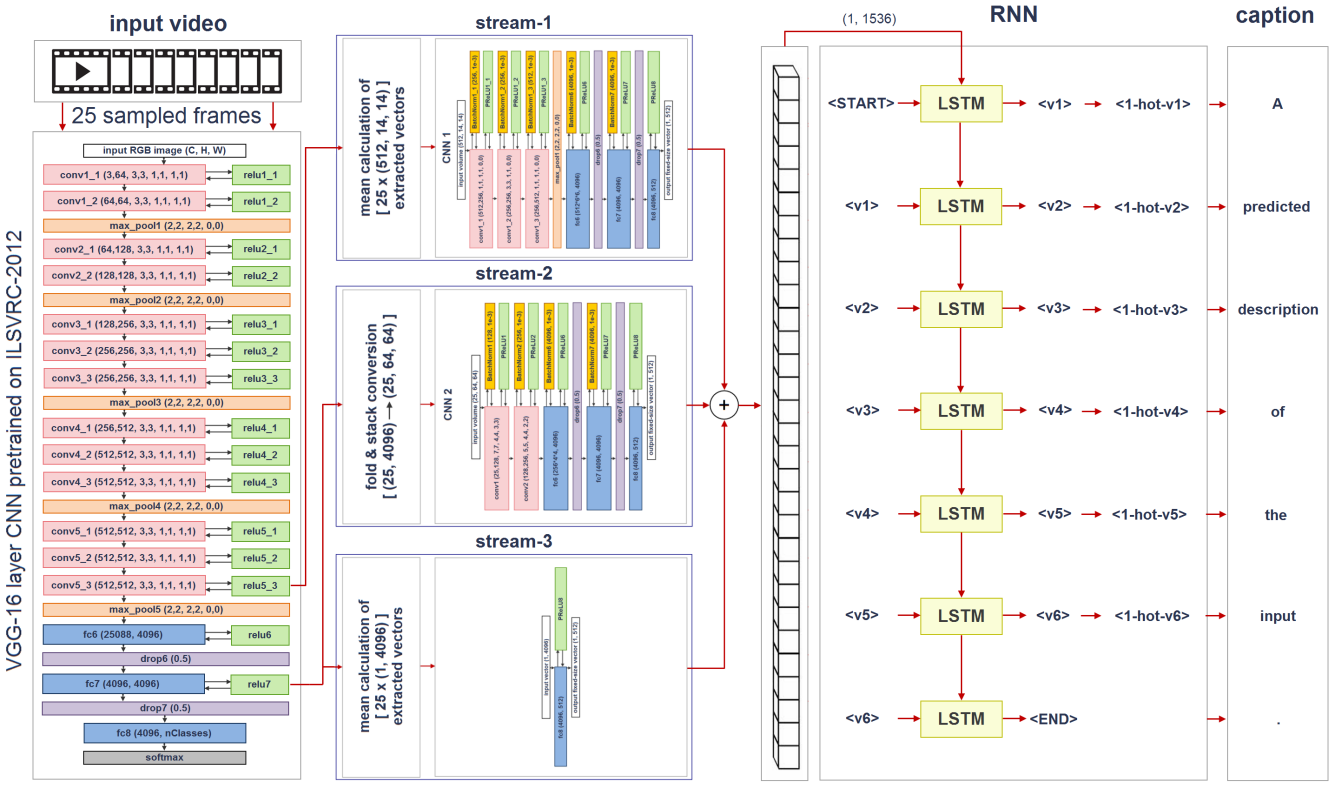

Fig. 1: The proposed video captioning system architecture.

As other state-of-the-art works are based on ideas emerged in the image captioning domain, and aim to introduce temporal information, in this work, we built upon a successful image captioning method, and we propose a three stream network, capable of capturing spatial semantic, temporal semantic, and global video concept information. That is, our major contribution lies in the intermediate incorporation of three parallel streams. Each stream is responsible for acquiring:

- summarization of localized dynamic concepts (stream-1)
- global dynamic concepts (stream-2)
- summarization of global dynamic concepts (stream-3)

Thus, each stream focuses on specific tasks. Collecting spatial, temporal and global encodings, about a video scene, produces another type of encoding for the recurrent network to translate with every stream having its contributing factor to the resulting encoding. Hence, by combining all the extracted spatiotemporal information, we obtain more complete and accurate captions in videos. The three building parts composing the proposed captioning model are described in the following Subsections. An overview of the proposed video captioning system architecture is provided in Fig. 1.

The idea of the proposed work is rooted in the standard utilization of the two-subnetwork approach, involving a convolutional

network and a recurrent network, in image captioning models [11, 28]. Each of the subnetworks plays a crucial role for the task of image captioning. That is, the first subnetwork acts as an encoder of images to feature vectors, while the second subnetwork acts as a decoder, translating the information hidden in the encodings to text sentences. Hence, in this paper, addressing the additional challenge of video captioning, that is the demand of acquiring information both for the objects depicted in the scene as for the timings of the scene's events, we propose the insertion of an intermediate 3-stream subnetwork that collects spatial, temporal and global encodings, about a video scene. Thus, it is produced another type of encoding for the recurrent network to translate and every stream has its contributing factor to the resulting encoding. Surveying the relevant literature, we observe the utilization of either spatial, temporal or global features separately. For example, we have seen the mean of feature vectors extracted from certain fully connected layer (i.e. fc7) of the convolutional subnetwork [27], to be used as a global encoding forwarded to the following recurrent network, but this approach falls short to provide detailed information about both the object positions and event times in a video scene. Table 1 summarizes the different kinds of information captured from the three-stream network. That is, it is shown the utilized layer for the feature extraction, the dimensions of the extracted features, and the kind of captured information.

Table 1: Three-Stream - Information

Feature Representation		**Information**		
Layer	**Dim**	**Temporal**	**Spatial**	**Global**
Last Convolutional	[512,14,14]	-	✓	-
Fully Connected	[25,64,64]	✓	-	✓
Fully Connected	[1,4096]	-	-	✓

3.1. Part I: CNN model

As mentioned previously, in this work we utilize the VGG 16-layer model pretrained on the IILSVRC-2014 to classify 1,000 ImageNet classes. The model consists of sixteen trained neural layers; the first thirteen are convolutional and the remaining three are fully connected. Max-pooling layers follow the second, forth, seventh, tenth, and thirteenth convolutional layers, while the ReLU non-linearity ($f(x) = max(0, x)$) is applied to every convolutional and fully connected layer, except the last fully connected layer. The output of the last fully connected layer is a distribution over 1000 ImageNet classes. The softmax loss is used during the training.

3.1.1. Feature Extraction

For a given video dataset (including caption-video correspondence), we sample a fixed number of 25 frames per video at equal frame distances. Each frame is passed through the VGG-16 model. We utilize the last convolutional layer, since the convolutional

layers capture spatial information, and for each frame we derive its feature representation, which is a volume of size $[512, 14, 14]$. Then we calculate the mean feature vector of the 25 frames and thus obtain one spatial representation per video.

Subsequently, we utilize the second fully connected layer, since the fully connected layers are meant to capture high level semantic concepts. Thus for each frame we obtain a 1-D feature vector consisting of 4096 positive floating point numbers. After extraction, we fold it into a 2-D vector of size $[64, 64]$ and stack each sampled and folded feature vector, forming a 3-D volume of size $[25, 64, 64]$. Regarding the "folding" procedure, we should note that there is no guarantee that meaningful patterns will be formed, however we have observed in the experiments that indeed such patterns are formed since this 2d rearrangement enhances the performance, and also allows us to use 2d convolutions. Subsequently, based on this observation, we also performed experiments aiming to create a more meaningful 2d arrangement using rearrangement of the 4096 concepts based on their correlation. However, we did not observe any significant improvement, while this rearrangement was also computationally expensive. Thus, we kept the initial order of the features. We should also note that using convolutional layers instead of dense ones keeps the number of trainable parameters to a level that acts also as a regularizer.

Lastly, we calculate the mean vector of the 25 feature vectors extracted at the fully connected layer per video. This corresponds to a 1-D vector of size $[1, 4096]$ which constitutes a summarization of the global dynamic concept per video.

3.2. Part II: The intermediate 3-stream neural network

The second part of the proposed model, is a three-stream network architecture, allowing for treating the extracted vectors of sizes $[512, 14, 14]$, $[25, 64, 64]$ and $[1, 4096]$. The 3-stream neural network, consists of two CNNs (streams 1 and 2) and a simple fully-connected layer (stream 3), placed in parallel. The network outputs a vectorized encoding of 512 numbers in each stream, and the concatenated result (which is a 1-D fixed size vector of 1536 numbers) is forwarded to the LSTM so that the predicted captions are generated.

3.2.1. Stream-1

The CNN, placed in the first stream, consists of 1 main bottleneck convolutional layer divided into 3 convolutional sublayers, each using kernels of size 1×1, 3×3 and 1×1, with no padding and stride 1, and three fully connected layers. All layers are followed by a Parametric Rectified Linear Unit (PReLU [8]) and a Batch Normalization (BatchNorm [10]) module, while max pooling is applied at the last convolutional sublayer. This stream receives its input from the last convolutional layer of the VGG-16 model and is responsible for capturing spatial semantic information.

3.2.2. Stream-2

The CNN, placed in the second stream is responsible for action recognition. It consists of 2 convolutional and three fully connected layers. The first convolutional layer uses kernels of size 7×7 with padding 3 and stride 4 and the second uses kernels

5×5 with padding 2 and stride 4, for performing convolutions, respectively. All layers are followed by a PReLU and BatchNorm module.

The reshaped feature representations, allowing the 2-D convolutions to be performed, of the second fully connected layer of size $[25, 64, 64]$, are fed to the second stream, aiming to exploit temporal information in frame sequences. Note that by transforming a 1-D feature vector of size $[1, 4096]$ into a 2-D vector of size $[64, 64]$ we do not change anything in the encodings, but provide additional spatial dimensionality for the CNN. Moreover, the entire 3-D stack of size $[25, 64, 64]$ contains temporal information about the whole video in depth.

3.2.3. Stream-3

The third stream is a simple fully-connected layer, which transforms a vector $[1, 4096]$ into a vector $[1, 512]$. We use the PReLU non-linearity, as in the previous two streams. This stream receives its input from the second fully connected layer of the pretrained model, by averaging the feature representations of all the video frames. We note that the feature used here is just a mean of the sampled fc7 vectors as used in [27]. This serves only as a global dynamic concept for the entire video as all temporal information would be lost through the mean operation.

3.3. Part III: The LSTM network

The language modeller which corresponds to the recurrent part of our video captioner consists of a 1-layer network of 512 LSTM units. The network is provided a special <*START*> token ($\boldsymbol{x}_0$) to initialize along with the concatenated fixed size vector (h_0) extracted from the 3-stream CNN as input. In time step t the network outputs a vector $\boldsymbol{y}_t$, which in turn is transformed into a 1-hot-encoding, that corresponds to a specific word in our vocabulary. The vector $\boldsymbol{y}_t$ is then fed into the next time step as $\boldsymbol{x}_{t+1}$, as input along with the vector $\boldsymbol{h}_t$ which carries new and past information about the sentence. This procedure continues until a special <*END*> token is generated.

The vocabulary used, is created by a suitable dataset, containing videos and sentence descriptions. These descriptions are processed, in such a way that words with a small number of occurrences or strange characters are omitted. The vocabulary then is made of N unique words and defines the length of the 1-hot-encoding vector. This vector is used to point to a specific word in the vocabulary. The RNN's goal is to perform a translation of the fixed size vector to a sequence of words forming a natural sentence describing the initial video content.

In every unit's core and at every time step, there is a memory cell encoding information of inputs that have been observed until that point. The cell's behaviour is controlled, by "gates" (layers that are applied and thus can either keep a value, if the gate returns 1, or reject it, if it returns 0). In a LSTM unit, there are three gates which control whether to forget the current cell value (*forget gate f*), read its input (*input gate i*) or output the new cell value (*output gate o*). The definition of gates, cell update and output are

described by the following equations:

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f) \tag{1}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i) \tag{2}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o) \tag{3}$$

$$\tilde{\boldsymbol{C}}_t = \tanh(\boldsymbol{W}_C \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_C) \tag{4}$$

$$\boldsymbol{C}_t = \boldsymbol{f}_t \odot \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{C}}_t \tag{5}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{C}_t) \tag{6}$$

where $\odot$ represents element-wise multiplication, $\boldsymbol{W}_f$, $\boldsymbol{W}_i$, $\boldsymbol{W}_o$, $\boldsymbol{W}_C$, $\boldsymbol{b}_f$, $\boldsymbol{b}_i$, $\boldsymbol{b}_o$ and $\boldsymbol{b}_C$ are trainable parameters and biases, $\boldsymbol{f}_t$, $\boldsymbol{i}_t$ and $\boldsymbol{o}_t$ are the forget, input and output gate outputs, respectively, $\tilde{\boldsymbol{C}}_t$ is the cell update, $\boldsymbol{C}_t$ is the new cell state and finally $\boldsymbol{h}_t$ is the cell output which is then fed into a Softmax, that will produce a probability distribution over all words in the vocabulary.

3.4. Training

We argue that the most important feature that differentiates images from videos is the temporal information exists in videos, that allows us to better recognize the temporal actions. Especially, for describing the content of a video that depicts people, recognizing their actions is crucial for the correct description. This is indeed a drawback of all the competitive methods that do not utilize temporal information. Thus, in this work, before including our stream networks in the training step of our entire video captioning model, we chose to train the weights of the second stream of the second subnetwork on the action classification dataset UCF-101 [23], since this is exactly the main task this stream should solve. That is to extract representations that help distinguishing the different human actions which in turn will help the RNN to generate correct video descriptions. This dataset consists of 13500 videos divided in 101 action related categories. According to the recommended splits, the extracted feature vectors of 9537 videos were used for training and 3783 for validation.

We then proceed with the training of the whole video captioning model in two steps. First, we use a pretrained VGG network as a feature extractor, in order to build the input dataset for the second stage preserving caption correspondence. Note that the VGG layer could also be finetuned in an end-to-end fashion, however this comes with additional computational cost. At the second stage, the created dataset of extracted feature vector triplets is passed as input to the pretrained 3-stream CNN. Its output is then forwarded on to the LSTM, which eventually yields the predicted caption for the initial video.

Subsequently, following the feature vector triplet input concept and similarly to [28], we propose to maximize the probability of the correct description given the vector triplet ($\boldsymbol{v}_1$[512, 14, 14], $\boldsymbol{v}_2$[25, 64, 64] and $\boldsymbol{v}_3$[1, 4096]), by using the following formulation:

$$\boldsymbol{\theta}^* = arg\max_{\boldsymbol{\theta}} \sum_{(\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_3,\boldsymbol{s})} \log p[\boldsymbol{s}|(\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_3);\boldsymbol{\theta}] \tag{7}$$

where $\boldsymbol{\theta}$ are the parameters of our model, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are the VGG-extracted volumes of sizes [512, 14, 14] and [25, 64, 64], respectively, $\boldsymbol{v}_3$ is the average of 25 reshaped vectors taken from $\boldsymbol{v}_2$ ([25, 64, 64] is reshaped to [25, 4096] which in turn is averaged to [1, 4096]) and $\boldsymbol{s}$ is the corresponding correct caption of the initial input video. Since $\boldsymbol{s}$ represents any sentence its length is unbounded. Thus, applying the chain rule to the model, the joint probability over $\boldsymbol{s}_0, ..., \boldsymbol{s}_N$, where N is the sentence length (in words) is expressed:

$$\log p[\boldsymbol{s}|(\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_3)] = \sum_{t=0}^{N} \log p[\boldsymbol{s}_t|(\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_3),\boldsymbol{s}_0,...,\boldsymbol{s}_{t-1}] \tag{8}$$

where the dependency on $\boldsymbol{\theta}$ is dropped for convenience. At training time, $\boldsymbol{s}$ and $(\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_3)$ is a training example triplet and we optimize the sum of the log probabilities as described in (8) over the whole training set using the adaptive moment estimation algorithm (Adam) [13], instead of the simple gradient descent.

4. Dataset

We trained and tested our model on the Microsoft Research Video Description Corpus (hereafter MSVD) [2], which contains 1970 short video snippets collected from YouTube (typically shorter than 10 seconds in length). There are almost 40 available English descriptions per video, totalling approximately 85550 captions. Data augmentation methods (e.g. horizontal flipping and cropping) also applied, resulting from 1200 training samples to 43200. In our experiments, we followed the following settings:

- Train set: 1200 videos and 49208 captions
- Validation set: 100 videos and 4098 captions
- Test set: 670 videos and 27402 captions

5. Experiments

In order to validate the performance of the proposed model, we conducted four experiments performing changes to our video captioner, leading to four distinct models. These changes mainly concern the intermediate added CNNs. Two of them are obtained by enabling the use of either stream 1 or 2 in the 3-stream CNN, the third is obtained by using both streams 1 and 2 in parallel and the fourth is obtained by using all three streams in parallel.

Additionally, at test time, our system can be executed as fully assembled, including the VGG network. This enables us to also perform real-time video captioning, given the trained model.

For evaluating the accuracy of the generated captions we used the COCO-caption API[1] [3], which returns a set of commonly used evaluation metrics. The most commonly used metric so far in the image/video captioning literature has been the BLEU score [20], which is a form of precision of word n-grams between generated and reference sentences (BLEU 1-4 scores correspond to n-grams with n=1-4 respectively). Typically an output score of '1' matches perfectly with the reference sentence, and a '0' means that the output sentence is completely unrelated to it. Even though this metric has some drawbacks, it has been shown to correlate well with human evaluations. More recently, a novel metric called CIDER [25] measures consistency between n-gram occurrences in generated and reference sentences, where this consistency is weighted by n-gram saliency and rarity. In addition, we provide results using METEOR [6] and ROUGE [17] metrics. All of the prementioned approaches are quite similar in that they measure syntactic similarities between two pieces of text, while each evaluation metric is designed to be correlated to some extent with human judgment.

5.1. Experimental Results

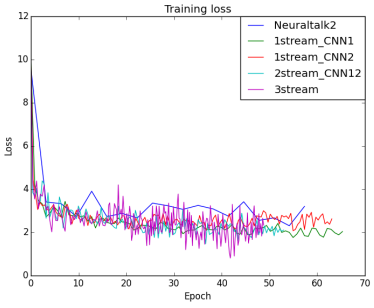

Fig. 2: Training loss per trained model

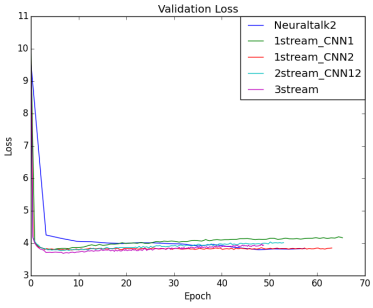

Fig. 3: Validation loss per trained model

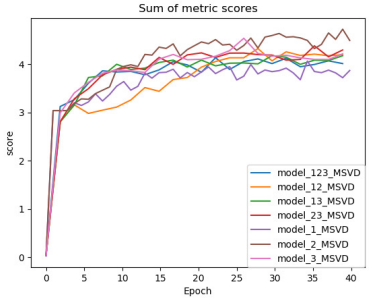

Fig. 4: Per model sums of metric scores on the MSVD validation set

In Table 2, we present the accuracy scores achieved in action classification over the corresponding validation sets of the UCF-101 dataset.

Table 2: Average classification accuracy on validation set of UCF-101

Split	Stream-2	
	top-1	top-5
split-1	72.29	88.95
split-2	72.11	89.39
split-3	69.06	87.20
average	**71.15**	**88.51**

In Table 3 we provide the captioning scores on the validation set of MSVD, while in Table 4 we the MSVD test set (consisting

[1] https://github.com/tylin/coco-caption

of 670 videos) using four trained configurations (two singlestreams, one dualstream and the best perfoming triplestream).

Table 3: Captioning scores on the validation set of MSVD

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Meteor	Rouge-L	CIDEr
Singlestream1	0.7277	0.6021	0.5029	0.4001	0.3173	0.6790	0.7298
Singlestream2	0.8042	0.7005	0.6078	0.5069	0.3629	0.7249	1.0148
Dualstream12	0.7733	0.6795	0.5981	0.5038	0.3366	0.7070	0.7230
Triplestream123	0.7943	0.6966	0.6031	0.4984	0.3553	0.7237	0.7242

Table 4: Captioning scores on the test set of MSVD

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Meteor	Rouge-L	CIDEr
Singlestream1	0.7167	0.5738	0.4706	0.3633	0.2817	0.6368	0.4172
Singlestream2	0.7553	0.6317	0.5353	0.4361	0.3138	0.6745	0.5405
Dualstream12	0.7576	0.6331	0.5336	0.4319	0.3185	0.6729	0.5970
Triplestream123	**0.7811**	**0.6643**	**0.5593**	**0.4502**	**0.3380**	**0.6962**	**0.6328**

In Table 5, we present the comparisons results of the proposed method against competitive ones on the MSVD dataset. As we can observe the proposed method outperforms the previous methods presented in [27], [26], [30], and [15].

We trained each model for at least 40 epochs. In Figures 2 and 3 we present the losses for the training and validation sets, respectively, while in Figure 4 we present the sums of all metric scores used on the MSVD validation set. In Fig. 5 we include a captioning example, produced by our system for visualization purposes. Additionally, we have uploaded a demonstration video on YouTube[2], featuring our model in action on a few video clips.

6. Conclusions

In this paper we proposed a novel automatic video captioning system, composing of three parts. The first one is a CNN model which acts as a feature extractor of single frames. The second is a three-stream network capable of capturing different kinds of information, and finally the third is a RNN model which generates the textual captions that are often not exactly the same as any of the sentences in the ground-truth set, which is an indication that the network succeeds on generalizing and is not overfitted. Experimental validation in a challenging video dataset indicated the effectiveness of the proposed captioning system.

Acknowledgments

Maria Tzelepi was supported by the General Secretariat for Research and Technology (GSRT) and the Hellenic Foundation for Research and Innovation (HFRI) (PhD Scholarship No. 2826).

[2] https://youtu.be/X2OOhhoipzQ

Table 5: Comparisons to other Video captioning methods on the test set of MSVD - Best results of each method are printed in bold

Model	BLEU	Meteor	CIDEr
[27] LSTM-YT (Basic)	0.3119	0.2687	-
[27] LSTM-YT (Flickr30k)	0.3203	0.2787	-
[27] LSTM-YT (COCO)	**0.3329**	**0.2907**	**-**
[27] LSTM-YT (COCO+Flickr30k)	0.3329	0.2888	-
[30] Enc-Dec (Basic)	0.3869	0.2868	0.4478
[30] + Local (3D-CNN)	0.3875	0.2832	0.5087
[30] + Global (Temporal Attention)	0.4028	0.2900	0.4801
[30] + Local + Global	**0.4192**	**0.2960**	**0.5167**
[15]- VGG16 non-attention	0.381	0.300	0.562
[15] - VGG16 dot	**0.411**	0.307	0.574
[15] - VGG16 bilinear	0.407	**0.310**	**0.615**
[15] - VGG16 concat	0.390	**0.310**	0.595
[15] - VGG16 sum	0.385	0.306	0.584
[26] S2VT RGB (VGG) random frame order	-	0.282	-
[26] S2VT RGB (VGG)	-	0.292	-
[26] S2VT RGB (VGG) + Flow (AlexNet)	-	**0.298**	-
Ours-Singlestream1	0.5311	0.2817	0.4172
Ours-Singlestream2	0.5896	0.3138	0.5405
Ours-Dualstream12	0.5891	0.3185	0.5970
Ours-Triplestream123	**0.6137**	**0.3380**	**0.6328**

References

[1] Baraldi, L., Grana, C., Cucchiara, R., 2017. Hierarchical boundary-aware neural encoder for video captioning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1657–1666.

[2] Chen, D.L., Dolan, W.B., 2011. Collecting highly parallel data for paraphrase evaluation, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, USA. pp. 190–200. URL: http://www.cs.utexas.edu/users/ai-lab/?chen:acl11.

[3] Chen, X., Fang, H., Lin, T., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L., 2015. Microsoft COCO captions: Data collection and evaluation server. CoRR abs/1504.00325. URL: http://arxiv.org/abs/1504.00325.

[4] Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches. CoRR abs/1409.1259. URL: http://arxiv.org/abs/1409.1259, arXiv:1409.1259.

[5] Deng, L., 2014. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing 3, e2.

[6] Denkowski, M., Lavie, A., 2014. Meteor universal: Language specific translation evaluation for any target language, in: Proceedings of the EACL 2014 Workshop on Statistical Machine Translation.

[7] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T., 2015. Long-term recurrent convolutional networks for visual recognition and description, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2625--2634.

[8] He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. CoRR abs/1502.01852. URL: http://arxiv.org/abs/1502.01852.

[9] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735--1780. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735, doi:10.1162/neco.1997.9.8.1735.

[10] Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR abs/1502.03167. URL: http://arxiv.org/abs/1502.03167.

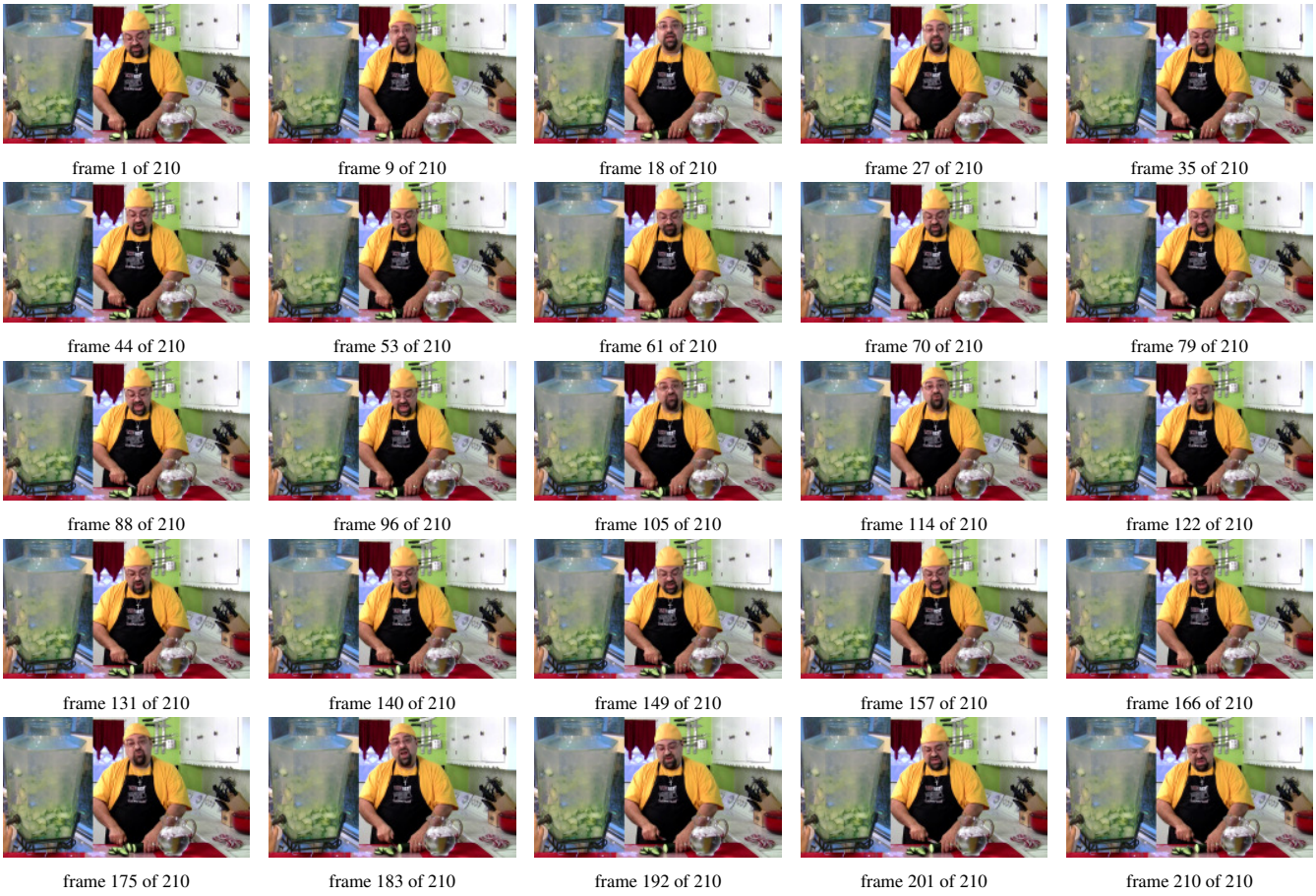

Fig. 5: Captioning example of MSVD video file (test split): "kquB3rIgfGk_525_532.avi", **ground truth captions:** *A guy is cutting cucumber. - A man cuts cucumber slices. - A man cuts up cucumber. - A man is chopping a cucumber into slices. - A man is chopping cucumbers. - A man is cutting a cucumber into slices. - A man is cutting cucumbers. - A man is cutting some vegetables. - A man is slicing a cucumber. - A man is slicing an unpeeled cucumber using a knife. - A man is slicing up some cucumbers. - The man cut up a cucumber. - The man is slicing cucumbers. - The man sliced a cucumber. - A man drinking fruit juice. - A man is making a drink. - A man is cutting a cucumber. - The man is cutting cucumber for making juice. - A man is slicing cucumbers. - Pictures of seven varieties. - The man is cutting vegetables. - A man is slicing a cucumber into pieces. - A man is cutting up a cucumber. - A man is cutting cucumber into pieces. - someone preparing something. - A man is cutting a vegetable. - Cooking with Jack shows how to make Lazy Man's drinks. - A man is slicing some cucumbers. - A chef slicing cucumber into circles. - A man cutting cucumber with a knife. - a man is cooking his kichen. - A chef slices cucumber. - a fat man is drinking. - How to make lazy mans drinks video.* **predicted captions:** *a man is putting sliced cucumbers into a pitcher of water.*

[11] Karpathy, A., Li, F., 2014. Deep visual-semantic alignments for generating image descriptions. CoRR abs/1412.2306. URL: http://arxiv.org/abs/1412.2306.

[12] Kinghorn, P., Zhang, L., Shao, L., 2018. A region-based image caption generator with refined descriptions. Neurocomputing 272, 416--424.

[13] Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. CoRR abs/1412.6980. URL: http://arxiv.org/abs/1412.6980.

[14] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 25. Curran Associates, Inc., pp. 1097--1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[15] Laokulrat, N., Phan, S., Nishida, N., Shu, R., Ehara, Y., Okazaki, N., Miyao, Y., Nakayama, H., 2016. Generating video description using sequence-to-sequence model with temporal attention, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 44--52.

[16] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278--2324.

[17] Lin, C.Y., 2004. Rouge: A package for automatic evaluation of summaries, in: Proc. ACL workshop on Text Summarization Branches Out, p. 10. URL: http://research.microsoft.com/~cyl/download/papers/WAS2004.pdf.

[18] Pan, Y., Mei, T., Yao, T., Li, H., Rui, Y., 2016. Jointly modeling embedding and translation to bridge video and language, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[19] Pan, Y., Yao, T., Li, H., Mei, T., 2017. Video captioning with transferred semantic attributes, in: CVPR.

[20] Papineni, K., Roukos, S., Ward, T., Zhu, W.J., 2002. Bleu: A method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA. pp. 311--318. URL: http://dx.doi.org/10.3115/1073083.1073135, doi:10.3115/1073083.1073135.

[21] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115, 211--252. doi:10.1007/s11263-015-0816-y.

[22] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556. URL: http://arxiv.org/abs/1409.1556.

[23] Soomro, K., Zamir, A.R., Shah, M., 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. CoRR abs/1212.0402. URL: http://arxiv.org/abs/1212.0402.

[24] Taigman, Y., Yang, M., Ranzato, M., Wolf, L., 2014. Deepface: Closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE. pp. 1701--1708.

[25] Vedantam, R., Zitnick, C.L., Parikh, D., 2014. Cider: Consensus-based image description evaluation. CoRR abs/1411.5726. URL: http://arxiv.org/abs/1411.5726.

[26] Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K., 2015a. Sequence to sequence -- video to text, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV).

[27] Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., Saenko, K., 2015b. Translating videos to natural language using deep recurrent neural networks, in: NAACL HLT.

[28] Vinyals, O., Toshev, A., Bengio, S., Erhan, D., 2014. Show and tell: A neural image caption generator. CoRR abs/1411.4555. URL: http://arxiv.org/abs/1411.4555.

[29] Xu, J., Mei, T., Yao, T., Rui, Y., 2016. Msr-vtt: A large video description dataset for bridging video and language, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[30] Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., Courville, A., 2015. Describing videos by exploiting temporal structure, in: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 4507--4515. doi:10.1109/ICCV.2015.512.

[31] You, Q., Jin, H., Wang, Z., Fang, C., Luo, J., 2016. Image captioning with semantic attention, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4651--4659.

[32] Yu, H., Wang, J., Huang, Z., Yang, Y., Xu, W., 2016. Video paragraph captioning using hierarchical recurrent neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4584--4593.