This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Top-down Visual Saliency Guided by Captions

Vasili Ramanishka
Boston University
vram@bu.edu

Abir Das
Boston University
dasabir@bu.edu

Jianming Zhang
Adobe Research
jianmzha@adobe.com

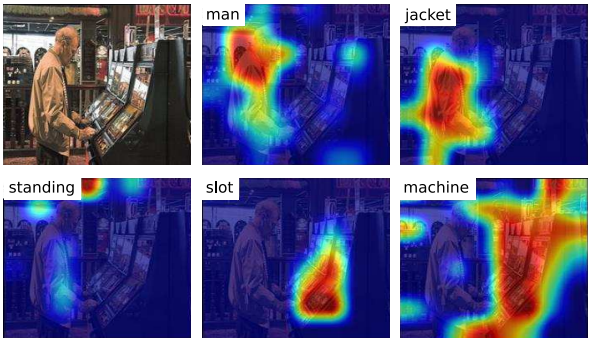
Kate Saenko
Boston University
saenko@bu.edu

Abstract

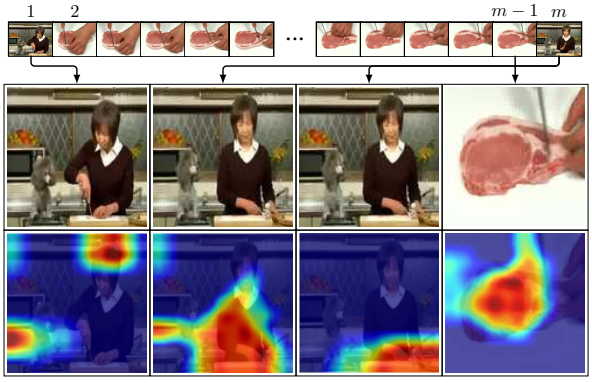
Neural image/video captioning models can generate accurate descriptions, but their internal process of mapping regions to words is a black box and therefore difficult to explain. Top-down neural saliency methods can find important regions given a high-level semantic task such as object classification, but cannot use a natural language sentence as the top-down input for the task. In this paper, we propose Caption-Guided Visual Saliency to expose the region-to-word mapping in modern encoder-decoder networks and demonstrate that it is learned implicitly from caption training data, without any pixel-level annotations. Our approach can produce spatial or spatiotemporal heatmaps for both predicted captions, and for arbitrary query sentences. It recovers saliency without the overhead of introducing explicit attention layers, and can be used to analyze a variety of existing model architectures and improve their design. Evaluation on large-scale video and image datasets demonstrates that our approach achieves comparable captioning performance with existing methods while providing more accurate saliency heatmaps. Our code is available at visionlearninggroup.github.io/caption-guided-saliency/.

1. Introduction

Neural saliency methods have recently emerged as an effective mechanism for top-down task-driven visual search [4, 31]. They can efficiently extract saliency heatmaps given a high-level semantic input, *e.g.*, highlighting regions corresponding to an object category, without any per-pixel supervision at training time. They can also explain the internal representations learned by CNNs [19, 30]. However, suppose we wanted to search a visual scene for salient elements described by a natural language sentence (Fig. 1(a)), or, given the description of an action, localize the most salient temporal and spatial regions corresponding to the subject, verb and other components (Fig. 1(b)). Classification-based saliency methods are insufficient for such language-driven tasks as they are limited to isolated object labels and cannot handle textual queries.



(a) **Input:** *A man in a jacket is standing at the slot machine*



(b) **Input:** *A woman is cutting a piece of meat*

Figure 1: Top-down *Caption-Guided Visual Saliency* approach that generates, for each word in a sentence, (a) spatial saliency in image and (b) spatiotemporal saliency in videos. For the video, we show temporally most important frames corresponding to the words at the bottom (arrows show positions of frames in the video) and spatial heatmaps indicating salient regions for these words.

Deep image and video captioning models [6, 23, 24, 28] excel at learning representations that translate visual input into language potentially discovering a mapping between visual concepts and words. However, despite the good captioning performance, they can be very hard to understand and are often criticized for being highly non-transparent "black boxes." They hardly provide any clear insight of the mapping learned internally between the image and the produced words. Consider for example, the video shown in Fig. 1(b). Which region in the model is used to predict

words like "woman" or "meat" in the generated caption? Is the word "woman" generated because the model recognized the woman in the video, or merely because the language model predicts that "A woman" is a likely way to start a sentence? Can the model learn to localize visual concepts corresponding to words while training only on weak annotations in the form of image or video-level captions? Can it localize words both in space and in time?

In this work, we address these questions by proposing a *Caption-Guided Visual Saliency* method that leverages deep captioning models to generate top-down saliency for both images and videos. Our approach is based on an encoder-decoder captioning model, and can produce spatial or spatiotemporal heatmaps for either a given input caption or a caption predicted by our model (Fig. 1). In addition to facilitating visual search, this allows us to expose the inner workings of deep captioning models and provide much needed intuition of what these models are actually learning. This, in turn, can lead to improved model design in the future. Previous attempts at such model introspection have analyzed LSTMs trained on text generation [13], or CNNs trained on image-level classification [31, 32]. Recent "soft" attention models [27, 28] produce heatmaps by learning an explicit attention layer that weighs the visual inputs prior to generating the next word, but require modification of the network and do not scale well. Thus, ours is the first attempt to analyze whether end-to-end visual captioning models can learn top-down saliency guided by linguistic descriptions without explicitly modeling saliency.

Our approach is inspired by the signal drop-out methods used to visualize convolutional activations in [30, 32], however we study LSTM based encoder-decoder models and design a novel approach based on information gain. We estimate the saliency of each temporal frame and/or spatial region by computing the information gain it produces for generating the given word. This is done by replacing the input image or video by a single region and observing the effect on the word in terms of its generation probability given the single region only. We apply our approach to both still image and video description scenarios, adapting a popular encoder-decoder model for video captioning [22] as our base model.

Our experiments show that LSTM-based encoder-decoder networks can indeed learn the relationship between pixels and caption words. To quantitatively evaluate how well the base model learns to localize words, we conduct experiments on the Flickr30kEntities image captioning dataset [17]. We also use our approach to "explain" what the base video captioning model is learning on the publicly available large scale Microsoft Video-to-Text (MSR-VTT) video captioning dataset [25]. We compare our approach to explicit "soft" attention models [27, 28] and show that we can obtain similar text generation performance with less computational overhead, while also enabling more accurate localization of words.

2. Related Work

Top-down neural saliency: Weak supervision in terms of class labels were used to compute the partial derivatives of CNN response with respect to input image regions to obtain class specific saliency map [19]. The authors in [30] used deconvolution with max-pooling layers that projects class activations back to the input pixels. While recent top-down saliency methods [4, 15, 31, 32] recover pixel importance for a given class using isolated object labels, we extend the idea to linguistic sentences.

Soft Attention: "Soft" attention architectures, developed for machine translation [2], were recently extended to image captioning [27]. Instead of treating all image regions equally, soft attention assigns different weights to different regions depending on the their content. Similarly, in video captioning, an LSTM with a soft attention layer attends to specific temporal segments of a video while generating the description [28]. Compared to our top-down saliency model, one drawback of soft attention is that it requires an extra recurrent layer in addition to the LSTM decoder, requiring additional designing of this extra layer parameters. The size of this layer scales proportionally to the number of items being weighted, *i.e.*, the number of frames or spatial regions. In contrast, our approach extracts the mapping between input pixels and output words from encoder-decoder models without requiring any explicit modeling of temporal or spatial attention and without modifying the network. Our intuition is that LSTMs can potentially capture the inter-dependencies between the input and the output sequences through the use of memory cells and gating mechanisms. Our framework visualizes both temporal and spatial attention without having to estimate additional weight parameters unlike explicit attention models, and can be used to analyse and provide explanations for a wide variety of encoder-decoder models.

Captioning Models: Captioning models based on a combination of CNN and LSTM networks have shown impressive performance both for image and video captioning [6, 23, 24, 28]. Dense captioning [11, 12] proposed to both localize and describe salient image regions. Works on referring expression grounding [10, 16, 18] localize input natural language phrases referring to objects or scene-parts in images. These methods use ground truth bounding boxes and phrases to learn a mapping between regions and phrases. We address the more difficult task of learning to relate regions to words and phrases without strong supervision of either, training only on images paired with their respective sentence captions. We also handle spatiotemporal grounding for videos in the same framework.

3. Background: Encoder-Decoder Model

We start by briefly summarizing our base captioning model. We utilize the encoder-decoder video description framework [23] which is based on sequence-to-sequence models proposed for neural translation [7, 20]. In Section 4 we will describe how our approach applies the same base model to caption still images.

Consider an input sequence of p video frames $\boldsymbol{x} = (x_1, \ldots, x_p)$ and a target sequence of n words $\boldsymbol{y} = (y_1, \ldots, y_n)$. The encoder first converts the video frames $\boldsymbol{x}$ into a sequence of m high-level feature descriptors:

$$V = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m) = \phi(\boldsymbol{x}) \tag{1}$$

where typically $\phi()$ is a CNN pre-trained for image classification. It then encodes the feature descriptors V into a fixed-length vector $\boldsymbol{z} = E(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)$, where E is some (potentially non-linear) function. In the S2VT [22], this is done by encoding V into a sequence of hidden state vectors $\boldsymbol{h}^e_i$ using an LSTM, where the state evolution equation is:

$$\boldsymbol{h}^e_i = f(\boldsymbol{v}_i, \boldsymbol{h}^e_{i-1}) \text{ for } i \in \{1, 2, \ldots, m\} \tag{2}$$

and then taking $\boldsymbol{z} = \boldsymbol{h}^e_m$, the last LSTM state. Another approach is to take the average of all m feature descriptors [23], *i.e.*, $\boldsymbol{z} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{v}_i$.

The decoder converts the encoded vector $\boldsymbol{z}$ into output sequence of words y_t, $t \in \{1, \ldots, n\}$. In particular, it sequentially generates conditional probability distribution for each element of the target sequence given encoded representation $\boldsymbol{z}$ and all the previously generated elements,

$$P(y_t|y_1, \ldots, y_{t-1}, \boldsymbol{z}) = D(y_{t-1}, \boldsymbol{h}^d_t, \boldsymbol{z}),$$
$$\boldsymbol{h}^d_t = g(y_{t-1}, \boldsymbol{h}^d_{t-1}, \boldsymbol{z}) \tag{3}$$

where $\boldsymbol{h}^d_t$ is the hidden state of the decoding LSTM and g is again a nonlinear function.

Soft Attention: Instead of using the last encoder LSTM state or averaging V, the authors in [28] suggest keeping the entire sequence V and having the encoder compute a *dynamic* weighted sum:

$$\hat{\boldsymbol{z}}_t = \sum_{i=1}^{m} \alpha_{ti} \boldsymbol{v}_i \tag{4}$$

Thus, instead of feeding an averaged feature vector into the decoder LSTM, at every timestep a weighted sum of the vectors is fed. The weights for every $\boldsymbol{v}_i$ are computed depending on previous decoder state $\boldsymbol{h}^d_{t-1}$ and encoded sequence $V = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)$. In video captioning, this allows for a search of related visual concepts in the whole video depending on the previously generated words. As a result, one can think about attention in this model as a generalization of simple mean pooling across video frames. Weights α_{ti} are obtained by normalizing e_{ti}, as follows,

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{k=1}^{m} exp(e_{tk})}$$
$$e_{ti} = \boldsymbol{w}^T tanh(\boldsymbol{W}_a \boldsymbol{h}_{t-1} + \boldsymbol{U}_a \boldsymbol{v}_i + \boldsymbol{b}_a) \tag{5}$$

where $\boldsymbol{w}$, $\boldsymbol{W}_a$, $\boldsymbol{U}_a$ and $\boldsymbol{b}_a$ are attention parameters of the attention module.

4. Approach

We propose a top-down saliency approach called *Caption-Guided Visual Saliency* which produces spatial and/or temporal saliency values (attention) for still images or videos based on captions. The saliency map can be generated for a caption predicted by the base model, or for an arbitrary input sentence. Our approach can be used to understand the base captioning model, *i.e.* how well it is able to establish a correspondence between objects in the visual input and words in the sentence. We use the encoder-decoder captioning model as our base model (equations 1, 2, 3).

For each word in the sentence, we propose to compute the saliency value of each item in the input sequence by measuring the decrease in the probability of predicting that word based on observing just that single item. This approach is flexible, does not require augmenting the model with additional layers, and scales well with input size. In contrast, in the soft attention model, the decoder selects relevant items from the input with the help of trainable attention weights. This requires additional layers to predict the weights. Furthermore, it can only perform either temporal or spatial mapping, but not both. Our method estimates both a temporal and a spatial mapping between input and output using the base LSTM encoder-decoder model by recovering the *implicit* attention from the model. We describe the more general case of video in Section 4.1 and then show how this model can be applied to still images in Section 4.2.

4.1. Video Saliency

In case of videos, we would like to compute the most salient spatiotemporal regions corresponding to words in the given sentence description of an event or activity. Figure 2 shows an overview of the approach. The intuition is that, although the encoder discards temporal and spatial positions of visual concept activations by encoding them into a fixed-length vector, this information can still be extracted from the model. The encoded representation, containing activations of all visual concepts detected in the entire video, is passed on to the decoder LSTM at the start of the sentence generation process. The decoder then chooses parts of this state vector using LSTM output gates to predict the word at time t. As each word is generated, the presence of visual concepts in the decoder LSTM state continually evolves, and the evolved state vector in turn interacts with the output

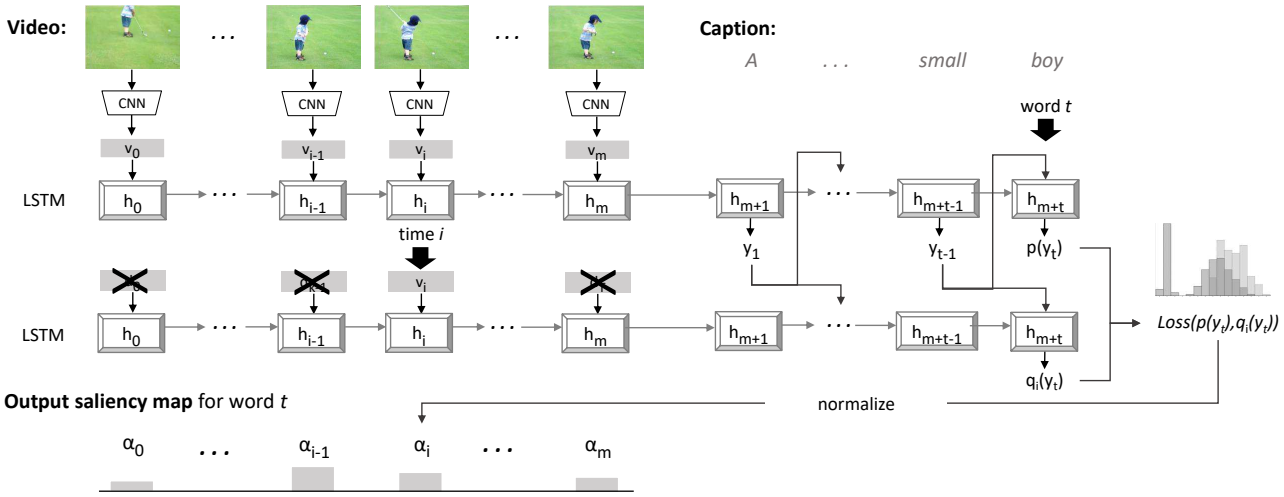


Figure 2: Overview of our proposed top-down *Caption-Guided Visual Saliency* approach for temporal saliency in video. We use an encoder-decoder model to produce temporal saliency values for each frame i and each word t in a given input sentence. The values are computed by removing all but the ith descriptor from the input sequence, doing a forward pass, and comparing to the original word probability distribution. A similar idea can be applied to spatial image saliency. See text for details.

gates to generate the next word. As this interaction is complex and non-linear, we devise an indirect scheme to extract the evidence for the generation of each word.

Our approach measures the amount of information lost when a single localized visual input is used to approximate the whole input sequence. The decoder predicts probability distributions $p(y_t)$ of words from the vocabulary at every step of the decoding process. We assume that this probability distribution is our "true" distribution. Then we measure how much information the descriptor of item i carries for the word at timestep t. To do this, we remove all descriptors from the encoding stage except for the ith descriptor. After computing a forward pass through the encoder and decoder, this gives us a new probability distribution $q_i(y_t)$. We then compute the loss of information as the KL-divergence between the two distributions,

$$\begin{aligned} p(y_t) &= P(y_t|y_{1:t-1}, v_{1:m}) \\ q_i(y_t) &= P(y_t|y_{1:t-1}, v_i) \\ Loss(t,i) &= D_{KL}(p(y_t)\|q_i(y_t)) \end{aligned} \tag{6}$$

With the above formulation we can easily derive top-down saliency for word w predicted at time t. We assume that the query sentence S has "one-hot" "true" distributions on every timestep. With this assumption Eq. 6 reduces to:

$$\begin{aligned} Loss(t,i,w) &= \sum_{k \in W} p(y_t = k) \log \frac{p(y_t = k)}{q_i(y_t = k)} \\ &= \log \frac{1}{q_i(y_t = w)} \end{aligned} \tag{7}$$

This process is not limited to produced word sequence only but can be used with any arbitrary query for a given video.

As the approximate receptive field of each descriptor can be estimated[1], we can define a saliency map for each word in the sentence by mapping $Loss(t,i,w)$ to the center of the receptive field and and upsampling the resulting heatmap. It follows from Eq. 7 that $Loss(t,i,w) \in [0; +\infty)$, where values which are closer to zero correspond to higher saliency. To obtain a saliency value e_{ti}, we negate the loss and linearly scale the resulting values to the $[0, 1]$ interval,

$$e_{ti} = scale(-Loss(t,i,w)) \tag{8}$$

It is important to discriminate between the values meant by Eq. 6 and. 7. The former can be used to evaluate the representativeness of individual descriptors compared to the full input sequence, while the latter induces top-down saliency maps for individual words at each time step. Finally, the saliency value for a group of words from the target sentence (e.g. a noun phrase "a small boy") is defined as sum of the corresponding saliency values for every word in the subsequence:

$$Loss(\{t_1, ..., t_q\}, i) = \sum_{j=1}^{q} Loss(t_j, i). \tag{9}$$

Next we describe how this approach is applied to generate both temporal and spatial saliency in videos.

Temporal attention: For an input frame sequence $V = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)$, the deterministic algorithm of sentence generation is given by the following recurrent relation:

$$w = \underset{y_t \in W}{\operatorname{argmax}}\, p(y_t|y_{0:t-1}, \boldsymbol{v}_{1:m}) \tag{10}$$

[1] for our video descriptors, the receptive field is a single frame

where y_0 and y_n are special "begin of sequence" and "end of sequence" tokens respectively. Given the word predicted at time t of the sentence, the relative saliency of the input frame $\boldsymbol{v}_i$ can be computed as e_{ti} (Eq. 8). In other words, we estimate the drop in probability of every word in the output sequence resulting from encoding only that input frame. Further, we normalize $\mathbf{e}_t = (e_{t1}, \ldots, e_{tm})$ to obtain stochastic vectors as in Eq. 5 and interpret the resulting vectors $\boldsymbol{\alpha}_t = (\alpha_{t1}, \ldots, \alpha_{tm})$ as saliency over the input sequence $V = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)$ for every word y_t of the output sequence. This also induces a direct mapping between predicted words and the most salient frames for these words.

Spatial attention: We can also estimate the attention on different frame patches as related to a particular word y_t of a sentence. Although spatial pooling in the CNN discards the spatial location of detected visual concepts, the different gates of the LSTM enable it to focus on certain concepts depending on the LSTM hidden state. Let $f_k(a, b)$ be the activation of unit k (corresponding to some visual concept) at spatial location (a, b) in the last convolutional layer of the encoder [32]. The CNN performs spatial average pooling to get a feature vector $\boldsymbol{v}_i$ for the i^{th} frame whose k^{th} element is $v_{ik} = \sum_{a,b} f_k(a, b)$. After that, the encoder embeds the descriptor into LSTM cell state according to the LSTM update rule. This process involves the LSTM input gate:

$$\rho_i = \sigma(W_{v\rho}\boldsymbol{v}_i + W_{h\rho}h_{i-1} + b_\rho) \tag{11}$$

where the LSTM selects activations $\boldsymbol{v}_{ik}$ by weighting them depending on the previous LSTM hidden state and $\boldsymbol{v}_i$ itself ($W_{v\rho}$, $W_{h\rho}$ and b_ρ are trainable parameters). Note that,

$$W_{v\rho}\boldsymbol{v}_i = \sum_k \boldsymbol{w}_k \boldsymbol{v}_{ik} = \sum_k \boldsymbol{w}_k \sum_{a,b} f_k(a, b) = \sum_{a,b}\sum_k \boldsymbol{w}_k f_k(a, b) \tag{12}$$

where $\boldsymbol{w}_k$ denotes the k^{th} column of matrix $W_{v\rho}$. Since each unit activation $f_k(a, b)$ represents a certain visual concept [30], we see that the input gate learns to select input elements based on relevant concepts detected in the frame, regardless of their location. The explicit spatial location information of these concepts is lost after the spatial average pooling in the last convolutional layer, however, we can recover it from the actual activations $f_k(a, b)$. This is achieved by computing the information loss for different spatial regions in a frame in a similar way as was done for temporal attention extraction. The relative importance of region (a, b) in frame $\boldsymbol{v}_i$ for word w predicted at time t can be estimated as:

$$\begin{aligned} e_{ti}^{(a,b)} &= -Loss(t, i, w), \\ \text{where } \; p(y_t) &= P(y_t | y_{0:t-1}, \boldsymbol{v}_{1:m}), \\ q_i(y_t) &= P(y_t | y_{0:t-1}, \boldsymbol{v}_i^{(a,b)}), \end{aligned} \tag{13}$$

and where $\boldsymbol{v}_{ik}^{(a,b)} = f_k(a, b)$. Assuming the number of spatial locations in a frame to be r, the prediction process (i.e. forward pass) is run m times to obtain temporal saliency maps and $r \times m$ times to obtain the spatial maps for the given video/sentence pair. This, in turn, involves $n + 1$ LSTM steps, so the total complexity is

$$O((\underbrace{r \times m + m}_{\text{spatial and temporal}}) \times (\underbrace{n + 1}_{\text{LSTM steps}})) \tag{14}$$

Since all $Loss(t, i, w)$ computations are performed independently, we can create a batch of size $r \times m + m$ and calculate all the saliency values efficiently in one pass.

4.2. Image Saliency

With minimal changes the above model can be applied to generate saliency for images. We accomplish this by rearranging the grid of descriptors produced by the last convolutional layer of the CNN into a "temporal" sequence $V = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)$ by scanning the image in a sequential manner (row by row), starting from the upper left corner and ending at the bottom right corner. Our model uses the encoder LSTM to scan the image locations and encode the collected visual information into hidden states and then decodes those states into the word sequence. Generating a spatial saliency map can now be achieved by the same process as described for temporal saliency in the previous section.

5. Experiments

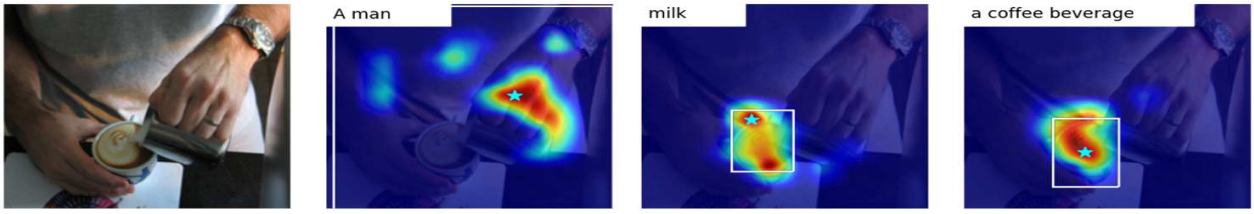
This section shows examples of caption-driven saliency recovered by our method for the base S2VT model from videos and still images. We evaluate the quality of the recovered heatmaps on an image dataset annotated with ground truth object bounding boxes. We also evaluate the caption generation performance on both images and videos and compare it with the soft attention approach.

Datasets We train and evaluate our model on two video description datasets, namely the Microsoft Video Description dataset (MSVD) [5] and the Microsoft Research Video to Text (MSR-VTT) [25] dataset. Both datasets have "in the wild" Youtube videos and natural language descriptions. MSVD contains 1970 clips of average length 10.2s with 80,827 natural language descriptions. MSR-VTT provides 41.2 hours of web videos as 10,000 clips of approx. 14.8s each and 200K natural language descriptions. In addition, we evaluated on one of the largest image captioning datasets, Flickr30kEntities [17] which is an extension of the original Flick30k [29] dataset with manual bounding box annotations for all noun phrases in all 158k image captions.

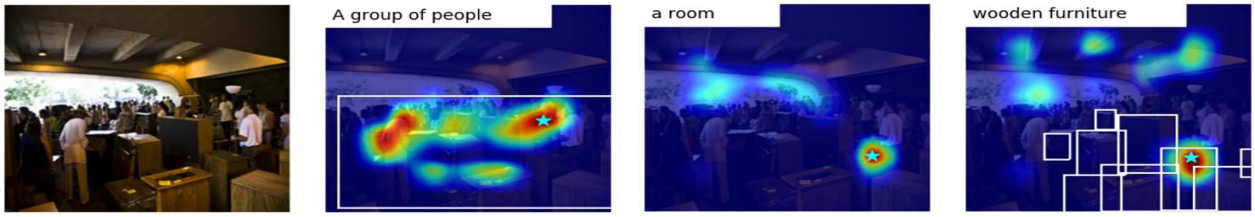
Model details We implemented our model in TensorFlow [1] using InceptionV3 [21] pretrained on ImageNet [8] as CNN feature extractor. We use $v_1, \ldots, v_{26}, v_i \in \mathbb{R}^{2048}$ for the video representation. v_i were extracted from the average pooling layer *pool_3* for 26 evenly spaced frames. For images we use feature outputs from the last convolutional layer *mixed_10* as the input sequence to the encoder.

(a) A man in blue jeans and a white t-shirt is working on a window with rail guards.

(b) Two people are in a room , one man is putting on a shirt and tie.



(c) A man is adding steamed milk to a coffee beverage.



(d) A group of people are standing in a room filled with wooden furniture.

Figure 3: Saliency maps (red to blue denotes high to low value) in Flickr30kentities generated for an arbitrary query sentence (shown below). Each row shows saliency map for different noun-phrases (shown at top-left corner) extracted from the query. Maximum saliency point is marked with asterisk and ground truth boxes are shown in white.

A woman in a red and white outfit is riding a bicycle.

Figure 4: Saliency maps generated for a caption (shown below the image) predicted by the model.

Thus, for video and image captioning the input sequences have length $m = 26$ and $m = 64$ respectively. The order of spatial descriptors for image captioning is described in Section 4.2. All images and video frames were scaled

Table 1: Evaluation of the proposed method on localizing all noun phrases from the ground truth captions in the Flickr30kEntities dataset using the pointing game protocol from [31]. "Baseline random" samples the point of maximum saliency uniformly from the whole image and "Baseline center" corresponds to always pointing to the center.

	bodyparts	animals	people	instruments	vehicles	scene	other	clothing	Avg per NP
Baseline random	0.100	0.240	0.318	0.179	0.275	0.524	0.246	0.151	0.268
Baseline center	**0.201**	0.599	**0.647**	**0.496**	0.644	0.652	0.384	**0.397**	0.492
Our Model	0.194	**0.690**	0.601	0.458	**0.645**	**0.667**	**0.427**	0.360	**0.501**

Table 2: Evaluation of the proposed method on Flickr30kEntities using the *attention correctness metric* and evaluation protocol from [14] (including the frame cropping procedure). Soft attention performance is taken from [14] as reported there. *Baseline** shows our re-evaluation of the uniform attention baseline.

	bodyparts	animals	people	instruments	vehicles	scene	other	clothing	Avg per NP
Baseline [14]	-	-	-	-	-	-	-	-	0.321
Soft attention [27]	-	-	-	-	-	-	-	-	0.387
Soft attention supervised [14]	-	-	-	-	-	-	-	-	0.433
Baseline*	0.100	0.371	0.410	0.278	0.350	0.470	0.236	0.197	0.325
Our model	**0.155**	**0.657**	**0.570**	**0.502**	**0.615**	**0.582**	**0.348**	**0.345**	**0.473**

to 299x299. Note that the CNN was trained on ImageNet and was not finetuned during the training of the captioning model. A fully-connected layer reduced the dimensionality of the input descriptors from 2048 to 1300 before feeding them into the LSTM. The model was trained using the Adam optimizer with initial learning rate 0.0005. Dimensionality of the word embedding layer was set to 300.

Evaluation of captioning performance Quantitative evaluation of the captioning performance was done using the METEOR [3] metric. Table 3 (higher numbers are better) shows the results and demonstrates that despite not using explicit attention layers, our model performs comparably to the soft attention method. The best model in terms of the METEOR metric on the validation split of Flickr30k was selected for the evaluation of saliency as presented below.

Quantitative evaluation of saliency Given a pretrained model for image captioning, we test our method quantitatively using the *pointing game* strategy [31] and *attention correctness metric* [14]. To generate saliency maps, we feed ground truth captions from the test split of Flickr30k into our model. In pointing game evaluation, we obtain the maximum saliency point inside the image for each annotated noun phrase in each GT caption of Flickr30kEntities. We then test whether this point lies inside the bounding box or not. Accuracy is computed as $Acc = \frac{\#Hits}{\#Hits+\#Misses}$. To get a saliency map for noun phrases which are comprised of multiple tokens from the sentence, we sum loss values before their normalizing them to the $[0, 1]$.

Table 1 shows the mean accuracy over all noun phrases (NPs) along with accuracies corresponding to categories (in different columns) from Flickr30kEntities. We compare to "Baseline random", where the maximum saliency point is sampled uniformly from the whole image and to a much stronger baseline denoted as "Baseline center". This baseline is designed to mimic the center bias present in consumer photos and assumes that the maximum saliency point is always at the center of the image. Compared to the random baseline, the accuracy of the proposed method is better on average (last column) as well as for all the individual categories (rest of the columns). While the average accuracy compared to the much stronger center baseline is only slightly better, the accuracy gain for some of the categories is significant. One possible reason may be that the objects in these categories, *e.g.*, 'animals' or 'other' objects, tend to be away from the central region of an image, while people tend to be in the center of the photo.

Table 2 provides a direct comparison of our method to the soft attention model [27] in terms of the *attention correctness metric* proposed in [14]. This metric measures average value for integral of attention function over bounding boxes. We directly report the results from [14] for their implementation of uniform baseline, soft-attention model and its improved version where a captioning model was trained to focus on relevant objects in supervised manner. Our method outperforms all three of them.

We also provide the category specific values as we obtained from our own implementation of the uniform baseline (called "Baseline*"). "Baseline random" in Table 1 should roughly correspond to "Baseline" and "Baseline*" in Table 2. Evidently, the exact values will be different as the evaluation protocols in the two tables are different. To compare the results fairly, we followed the same protocol as [14] where the authors performed a central crop of both test and training images. Human-captured images or videos tend to put the objects of interest in the central region. Thus, any cropping operation which enhances this "central ten-

Figure 5: Spatial and temporal saliency maps in videos. For each word, darker grey indicates higher relative saliency of the frame. For better visualization, saliency values are not normalized but linearly mapped to the range [0, 1]. Most relevant frames for each word are shown at the bottom, highlighted with the same color.

Table 3: Comparison of the captioning performance of our model and soft-attention on two video (MSVD, MSR-VTT) and one image (Flickr30k) datasets. Higher numbers are better.

Model	Dataset	METEOR [9]
Soft-Attn [28]	MSVD	30.0
Our Model	MSVD	31.0
Soft-Attn [26]	MSR-VTT	25.4
Our Model	MSR-VTT	25.9
Soft-Attn [27]	Flickr30k	18.5
Our Model	Flickr30k	18.3

dency" will, inherently, give a better measure of attention. This frame cropping strategy is another source of discrepancy in the baseline values in Table 1 and 2.

Saliency visualizations in images Figures 3 and 4 show example saliency maps on images from Flickr30kEntities for arbitrary query sentences and model-predicted captions, respectively. The arbitrary query comes from the ground truth descriptions. For each nounphrase, the saliency map is generated by summing the responses for each token in the phrase and then renormalizing them. The map is color coded where red shows the highest saliency while blue is the lowest. The maximum saliency point is marked with an asterisk, while the ground truth boxes for the noun-phrases are shown in white. It can be seen that our model almost always localizes humans correctly. For some other objects the model makes a few intuitive mistakes. For example, in Fig. 3a, though the saliency for "window" is not pointing to the groundtruth window, it focuses its highest attention (asterisk) on the gate which looks very similar to a window. In Fig. 4, the saliency map the predicted caption fof an image is shown. Some non-informative words (*e.g.*, "a", "is" *etc.*) may appear to have concentrated saliency, however, this is merely a result of normalization. One surprising observation is that the model predicts 'a woman in a red and white outfit', however only the 'red' spatial attention is on the cyclist, while the 'white' attention is on other parts of the scene.

Saliency visualizations in videos Fig. 5 shows examples of spatial and temporal saliency maps for videos from MSR-VTT dataset with model-predicted sentences. Most discriminative frames for each word are outlined in the same color as the word. Darker gray indicates higher magnitude of temporal saliency for the word. We omit visualization for uninformative words like articles, helper verbs and prepositions. An interesting observation about the top video is that the most salient visual inputs for "skating" are regions with snow, with little attention on the skier, which could explain the mistake.

Additional results and source code are available at visionlearninggroup.github.io/caption-guided-saliency/.

6. Conclusion

We proposed a top-down saliency approach guided by captions and demonstrated that it can be used to understand the complex decision processes in image and video captioning without making modifications such as adding explicit attention layers. Our approach maintains good captioning performance while providing more accurate heatmaps than existing methods. The model is general and can be used to understand a wide variety of encoder-decoder architectures.

7. Acknowledgements

This research was supported in part by NSF IIS-1212928, DARPA, Adobe Research and a Google Faculty grant. We thank Subhashini Venugopalan for providing an implementation of S2VT [22] and Stan Sclaroff for many useful discussions.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. 5

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations*, 2015. 2

[3] S. Banerjee and A. Lavie. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Association for Computational Linguistics Workshop*, volume 29, pages 65–72, 2005. 7

[4] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, D. Ramanan, and T. S. Huang. Look and Think Twice: Capturing Top-Down Visual Attention With Feedback Convolutional Neural Networks. In *IEEE International Conference on Computer Vision*, December 2015. 1, 2

[5] D. L. Chen and W. B. Dolan. Collecting Highly Parallel Data for Paraphrase Evaluation. In *Association for Computational Linguistics: Human Language Technologies*, pages 190–200, 2011. 5

[6] X. Chen and C. L. Zitnick. Mind's Eye: A Recurrent Visual Representation for Image Caption Generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2422–2431, 2015. 1, 2

[7] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Empirical Methods in Natural Language Processing*, pages 1724–1734, 2014. 3

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 5

[9] M. Denkowski and A. Lavie. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *EACL Workshop on Statistical Machine Translation*, 2014. 8

[10] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural Language Object Retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[11] J. Johnson, A. Karpathy, and L. Fei-Fei. DenseCap: Fully Convolutional Localization Networks for Dense Captioning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[12] A. Karpathy and L. Fei-Fei. Deep Visual-Semantic Alignments for Generating Image Descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015. 2

[13] A. Karpathy, J. Johnson, and F.-F. Li. Visualizing and Understanding Recurrent Networks. *arXiv preprint arXiv:1506.02078*, 2015. 2

[14] C. Liu, J. Mao, F. Sha, and A. L. Yuille. Attention Correctness in Neural Image Captioning. In *Association for the Advancement of Artificial Intelligence Conference*, 2017. 7

[15] A. Mahendran and A. Vedaldi. Understanding Deep Image Representations by Inverting Them. In *IEEE conference on computer vision and pattern recognition*, 2015. 2

[16] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and Comprehension of Unambiguous Object Descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[17] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. In *IEEE International Conference on Computer Vision*, pages 2641–2649, 2015. 2, 5

[18] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of Textual Phrases in Images by Reconstruction. In *European Conference on Computer Vision*, 2016. 2

[19] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *International Conference on Learning Representations Workshops*, 2013. 1, 2

[20] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning with Neural Networks. In *Neural Information Processing Systems*, pages 3104–3112, 2014. 3

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 5

[22] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to Sequence – Video to Text. In *IEEE International Conference on Computer Vision*, 2015. 2, 3, 8

[23] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating Videos to Natural Language Using Deep Recurrent Neural Networks. In *Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies*, 2015. 1, 2, 3

[24] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and Tell: A Neural Image Caption Generator. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015. 1, 2

[25] J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 5

[26] J. Xu, T. Mei, T. Yao, and Y. Rui. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language [Supplementary Material]. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8

[27] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015. 2, 7, 8

[28] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing Videos by Exploiting Temporal Structure. In *IEEE International Conference on Computer Vision*, 2015. 1, 2, 3, 8

[29] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference over Event Descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 5

[30] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer vision*, pages 818–833. 2014. 1, 2, 5

[31] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down Neural Attention by Excitation Backprop. *European Conference on Computer vision*, 2016. 1, 2, 7

[32] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 5