Recurrent Multimodal Interaction for Referring Image Segmentation

Chenxi Liu[1] Zhe Lin[2] Xiaohui Shen[2] Jimei Yang[2] Xin Lu[2] Alan Yuille[1]
Johns Hopkins University[1] Adobe Research[2]
{cxliu, alan.yuille}@jhu.edu {zlin, xshen, jimyang, xinl}@adobe.com

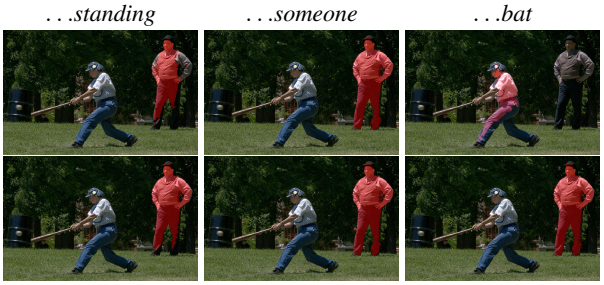
Abstract

In this paper we are interested in the problem of image segmentation given natural language descriptions, i.e. referring expressions. Existing works tackle this problem by first modeling images and sentences independently and then segment images by combining these two types of representations. We argue that learning word-to-image interaction is more native in the sense of jointly modeling two modalities for the image segmentation task, and we propose convolutional multimodal LSTM to encode the sequential interactions between individual words, visual information, and spatial information. We show that our proposed model outperforms the baseline model on benchmark datasets. In addition, we analyze the intermediate output of the proposed multimodal LSTM approach and empirically explain how this approach enforces a more effective word-to-image interaction.[1]

1. Introduction

In this paper, we study the challenging problem of using natural language expressions to segment an image. Given both an image and a natural language expression, we are interested in segmenting out the corresponding region referred by the expression. This problem was only introduced recently, but has great value as it provides new means for interactive image segmentation. Specifically, people can segment/select image regions of their interest by typing natural language descriptions or even speaking to the computer [20].

Given the success of convolutional neural networks in semantic segmentation [24, 2, 3], an immediate way to tackle this problem is to augment the convolutional semantic segmentation networks with a LSTM [11] sentence encoder [12], so that the image features and sentence representation can be combined to produce the desired mask. In fact, this sentence-to-image interaction scheme has been also adopted by recent methods on referring object localization [38] and visual question answering tasks [1].



Man in a vest and blue jeans standing watching someone swing a bat.

Figure 1: Given the image and the referring expression, we are interested in segmenting out the referred region. Each column shows segmentation result until after reading the underlined word. Our model (second row) explicitly learns the progression of multimodal interaction with convolutional LSTM, which helps long-term memorization and correctly segments out the referred region compared with the baseline model (first row) which uses language-only LSTM.

However, this sentence-to-image scheme does not reflect how humans tackle this problem. In sentence-picture verification, it is found through eye tracking that when pictures and sentences are presented together, people either follow a image-sentence-image reading sequence, or go back-and-forth between sentence and picture a number of times before making the decision [33]. In other words, the interaction between image and sentence should prevail from the beginning to the end of the sentence, instead of only happening at the end of the sentence. Presumably this is because the semantic information is more concrete and therefore more easily remembered when grounded onto the image. For example, consider the expression "the man on the right wearing blue". Without seeing an actual image, all information in the sentence needs to be remembered, meaning the sentence embedding needs to encode `IS_MAN`, `ON_RIGHT`, `WEAR_BLUE` jointly. However, with the actual image avail-

[1]Code is available at https://github.com/chenxi116/TF-phrasecut-public

able, the reasoning process can be decomposed as a sequential process, where the model first identifies all pixels that agree with IS_MAN, then prunes out those that do not correspond with ON_RIGHT, and finally suppresses those that do not agree with WEAR_BLUE.

Motivated by this sequential decision making theory, we propose a two-layered convolutional multimodal LSTM network that explicitly models word-to-image interaction. Different from the language-only LSTM encoder in previous works [12], the convolutional multimodal LSTM takes both visual feature and language representation as input to generate the hidden state that retains both the spatial and semantic information in memory. Therefore its hidden state models how the multimodal feature progresses over time or word-reading order. After seeing the last word, we use a convolution layer to generate the image segmentation result.

In summary, the contribution of our paper is three-fold:

- We propose a novel model, namely convolutional multimodal LSTM, to encode the sequential interactions between individual semantic, visual, and spatial information.
- We demonstrate the superior performance of the word-to-image multimodal LSTM approach on benchmark datasets over the baseline model.
- We analyze the intermediate output of the proposed multimodal LSTM approach and empirically explain how this approach enforces a more effective word-to-image interaction.

2. Related Work

In this section, we review recent studies that are tightly related to our work in the following three areas: semantic segmentation, referring expression localization, and multimodal interaction representation.

Semantic Segmentation Many state-of-the-art semantic segmentation models employ a fully convolutional network [24] architecture. FCN converts the fully connected layers in VGG network [32] into convolutional layers, thereby allowing dense (although downsampled) per-pixel labeling. However, too much downsampling (caused by pooling layers in the VGG architecture) prohibits the network from generating high quality segmentation results. DeepLab [2] alleviates this issue by discarding two pooling operations with atrous convolution. With Residual network [10] as its backbone architecture, DeepLab [3] is one of the leading models on Pascal VOC [7]. We use both ResNet-101 (with atrous convolution) and DeepLab ResNet-101 to extract image features in a fully convolutional manner. Following [2, 3], we also report the result of using DenseCRF [18] for refinement.

Referring Expression Localization More and more interest arise recently in the problem of localizing objects based on a natural language expression. In [26] and [14], image captioning models [27, 6] are modified to score the region proposals, and the one with the highest score is considered as the localization result. In [30], the alignment between the description and image region is learned by reconstruction with attention mechanism. [37] improved upon [26] by explicitly handling objects of the same class within the same image, while [29] focused on discovering interactions between the object and its context using multiple-instance learning. However all these works aim at finding a bounding box of the target object instead of segmentation mask. Perhaps the most relevant work to ours is [12], which studies the same problem of image segmentation based on referring expressions. Our approach differs in that we model the *sequential* property of interaction between natural language, visual, and spatial information. In particular, we update the segmentation belief after seeing each word.

Multimodal Interaction Representation Our work is also related to multimodal feature fusion in visual question answering [16, 9, 25] and image captioning [6]. In [6] the input to LSTM is the image feature and the previous word's embedding, whereas in [25] the input to LSTM is the image feature and individual question word's embedding. Attention mechanism [35, 34, 36, 22, 23] may also be applied, mostly to improve the relevance of image features. In both tasks the goal is to generate a textual sequence. Here instead, we use the LSTM hidden states to generate segmentation, which is not commonly considered a sequential task and requires preservation of spatial location. We achieve this by applying LSTM in a convolutional manner [31, 4, 8], unlike prior work on recurrent attention [28, 19].

3. Models

In this section, we first introduce our notation for this problem (section 3.1), and then describe the baseline model based on the sentence-to-image scheme [12] (section 3.2), which only models the progression of semantics. In section 3.3 we propose convolutional multimodal LSTM for fusing both modalities and model the progression of multimodal features in addition to the progression of semantics.

3.1. Notation

In the referring image segmentation problem, we are given both an image I and a natural language description $S = \{w_1, w_2, \ldots, w_T\}$, where w_t ($t \in \{1, 2, \ldots, T\}$) are individual words in the sentence. The goal is to segment out the corresponding region in the image. We will use R for prediction and $\hat{R}$ for ground truth. $R^{ij} \in (0, 1)$ represents the foreground probability of a pixel, where i and j are spatial coordinates. $\hat{R}^{ij} \in \{0, 1\}$, where 1 means the pixel is referred to by S and 0 otherwise.

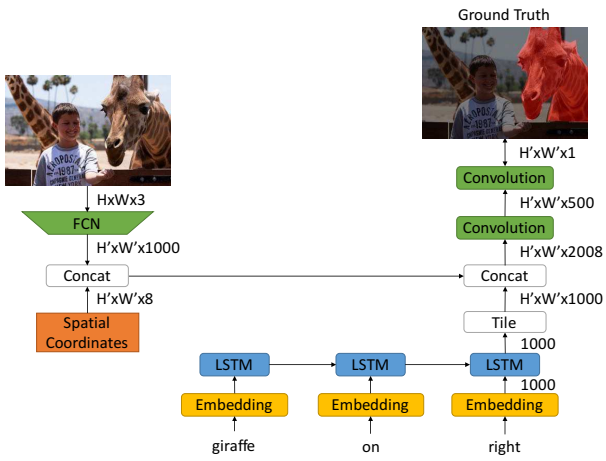


Figure 2: Network architecture of the baseline model described in section 3.2. In this model, the entire sentence is encoded into a fixed vector with language-only LSTM without using visual information.

3.2. Baseline Model

Our model is based on the model proposed in [12]. In [12], given an image of size $W \times H$, an FCN-32s [24] is used to extract image features with size $W' \times H' \times D_I$, where $W' = W/32$ and $H' = H/32$. The image features are then concatenated with spatial coordinates to produce a $W' \times H' \times (D_I + 8)$ tensor. The 8 spatial coordinate dimensions follow the implementation of [12]. The normalized horizontal/vertical position uses 3 dimensions each. The remaining 2 dimensions are $1/W'$ and $1/H'$. We use $\mathbf{v}^{ij} \in \mathbb{R}^{D_I+8}$ to represent the image-spatial feature at a specific spatial location.

As for the referring expression, every word w_t is one-hot encoded and mapped to a word embedding $\mathbf{w}_t$. The entire sentence is then encoded with an LSTM into a vector $\mathbf{h}_T$ of size D_S, where $\mathbf{h}_t$ represents the hidden state of LSTM at time step t:

$$\text{LSTM} : (\mathbf{w}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}) \rightarrow (\mathbf{h}_t, \mathbf{c}_t) \tag{1}$$

$$\begin{pmatrix} \mathbf{i} \\ \mathbf{f} \\ \mathbf{o} \\ \mathbf{g} \end{pmatrix} = \begin{pmatrix} \text{sigm} \\ \text{sigm} \\ \text{sigm} \\ \tanh \end{pmatrix} M_{4n, D_S+n} \begin{pmatrix} \mathbf{w}_t \\ \mathbf{h}_{t-1} \end{pmatrix} \tag{2}$$

$$\mathbf{c}_t = \mathbf{f} \odot \mathbf{c}_{t-1} + \mathbf{i} \odot \mathbf{g} \tag{3}$$

$$\mathbf{h}_t = \mathbf{o} \odot \tanh(\mathbf{c}_t) \tag{4}$$

where n is the size of the LSTM cell. $\mathbf{i}, \mathbf{f}, \mathbf{o}, \mathbf{g}$ are the input gates, forget gates, output gets, and memory gates respectively. $\mathbf{c}_t$ are the memory states at time step t.

The vector $\mathbf{h}_T$ is then concatenated with the image features and spatial coordinates at all locations to produce a $W' \times H' \times (D_I + D_S + 8)$ tensor. Two additional convolutional layers and one deconvolution layer are attached to the tensor to produce the final segmentation mask $R \in \mathbb{R}^{W \times H}$.

Given the ground truth binary segmentation mask $\hat{R}$, the loss function is

$$L_{high} = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} \Big(\hat{R}^{ij} * -\log(R^{ij}) + (1 - \hat{R}^{ij}) * -\log(1 - R^{ij}) \Big) \tag{5}$$

The whole network is trained with standard back-propagation.

Our baseline employs the same architecture, except that we use ResNet-101 [10] instead of FCN-32s to extract image features. One limitation of FCN-32s is that downsampling by 32 makes W' and H' too small. Therefore similar to the treatment of DeepLab [2, 3], we reduce the stride of conv4_1 and conv5_1 in ResNet-101 from 2 to 1, and use atrous convolution of rate 2 and 4 to compensate for the change. This operation reduces the downsampling rate from 32 to 8, which is relatively dense and allows loss to be computed at the feature resolution ($W' = W/8, H' = H/8$) instead of the image resolution. Therefore in our model, the loss function becomes

$$L_{low} = \frac{1}{W'H'} \sum_{i=1}^{W'} \sum_{j=1}^{H'} \Big(\hat{R}^{ij} * -\log(R^{ij}) + (1 - \hat{R}^{ij}) * -\log(1 - R^{ij}) \Big) \tag{6}$$

We use bilinear interpolation to upsample $R \in \mathbb{R}^{W' \times H'}$ at test time.

We are going to show in the experimental section that combining ResNet with atrous convolution results in a more competitive baseline model and easier training procedure.

3.3. Recurrent Multimodal Interaction Model

In the baseline model described above, segmentation is performed once, after the model has seen and memorized the entire referring expression. The memorization is the process of updating LSTM hidden states while scanning the words in the expression one by one. However, as discussed earlier, this requires the model to memorize all the attributes in the sentence jointly. We instead utilize the sequential property of natural language and turn referring image segmentation into a sequential process. This requires the language model to have access to the image from the beginning of the expression, allowing the semantics to be grounded onto the image early on. Therefore we consider modeling of the multimodal interaction, i.e. a scheme that can memorize the *multimodal* information (language, image, spatial information, and their interaction), which has direct influence on the segmentation prediction.

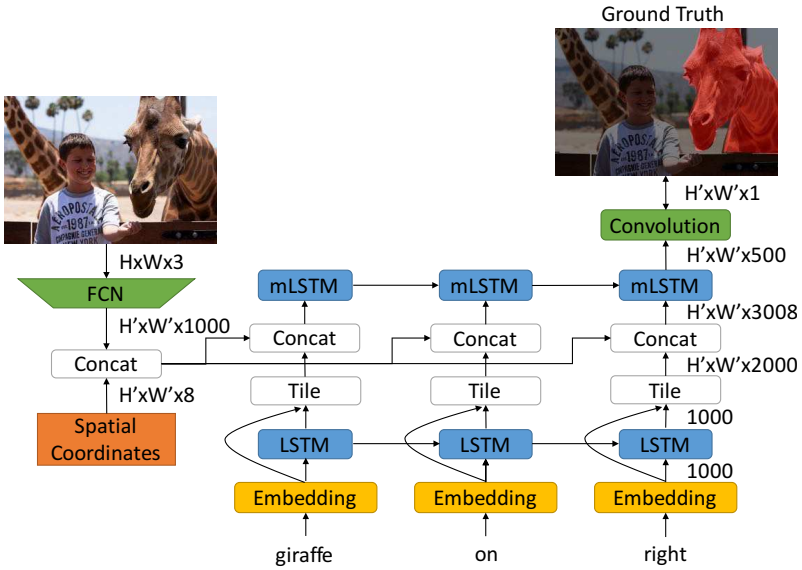


Figure 3: Network architecture of the RMI model described in section 3.3. By using the convolutional multimodal LSTM, our model allows multimodal interaction between language, image, and spatial information at each word. The mLSTM is applied to all location in the image and implemented as a 1×1 convolution.

We use a multimodal LSTM to capture the progression of rich multimodal information through time as shown in Fig. 3. Specifically, a multimodal LSTM (mLSTM) uses the concatenation of the language representation $\mathbf{l}_t \in \mathbb{R}^{D_S}$ and the visual feature at a specific spatial location $\mathbf{v}_{ij} \in \mathbb{R}^{D_I+8}$ as its input vector:

$$\text{mLSTM} : (\begin{bmatrix}\mathbf{l}_t \\ \mathbf{v}^{ij}\end{bmatrix}, \mathbf{h}_{t-1}^{ij}, \mathbf{c}_{t-1}^{ij}) \rightarrow (\mathbf{h}_t^{ij}, \mathbf{c}_t^{ij}) \quad (7)$$

The same mLSTM operation is shared for all image locations. This is equivalent to treating the mLSTM as a 1×1 convolution over the feature map of size $W' \times H' \times (D_I + D_S + 8)$. In other words, this is a convolutional LSTM that shares weights both across spatial location and time step.

The baseline model uses language-only LSTM (Equation 1) to encode the referring expression, and concatenate it with the visual feature to produce $\begin{bmatrix}\mathbf{h}_T \\ \mathbf{v}^{ij}\end{bmatrix}$. One advantage of multimodal LSTM is that either of the two components can be produced by it. The matrix M in multimodal LSTM will be of size $4n \times (D_S + D_I + 8 + n)$. If $M_{1:4n, D_S+1:D_S+D_I+8} = \mathbf{0}$, then the mLSTM will essentially ignore the visual part of the input, and encode only the semantic information. On the other hand, if the mLSTM ignores the language representation, the mLSTM will see the same input $\mathbf{v}_{ij}$ at all time steps, therefore very likely to retain that information.

From another perspective, multimodal LSTM forces word-visual interaction and generates multimodal feature at every recurrent step, which is key to good segmentation. In the baseline model, in order for the language representation to reach the multimodal level, it has to go through all subsequent LSTM cells as well as a convolution layer:

$$\mathbf{l}_t \xrightarrow{LSTM} \mathbf{h}_T \xrightarrow{Concat} \begin{bmatrix}\mathbf{h}_T \\ \mathbf{v}^{ij}\end{bmatrix} \xrightarrow{Conv} \text{multimodal feature} \quad (8)$$

while with multimodal LSTM this can be done with just the (multimodal) LSTM cells:

$$\mathbf{l}_t \xrightarrow{Concat} \begin{bmatrix}\mathbf{l}_t \\ \mathbf{v}^{ij}\end{bmatrix} \xrightarrow{mLSTM} \text{multimodal feature} \quad (9)$$

Note that the visual feature still only needs one weight layer to become multimodal.

In our Recurrent Multimodal Interaction (RMI) model, we take the language representation $\mathbf{l}_t$ to be the concatenation of language-only LSTM hidden state in Equation 1 and word embedding $\begin{bmatrix}\mathbf{h}_t \\ \mathbf{w}_t\end{bmatrix}$. This forms a two-layer LSTM structure, where the lower LSTM only encodes the semantic information, while the upper LSTM generates the multimodal feature. The lower language-only LSTM is spatial-agnostic, while the upper multimodal LSTM preserves feature resolution $H' \times W'$.

4. Experiments

4.1. Datasets

We use four datasets to evaluate our model: Google-Ref [26], UNC [37], UNC+ [37], and ReferItGame [15].

Table 1: Comparison of segmentation performance (IOU). In the first column, R means ResNet weights, D means DeepLab weights, and DCRF means DenseCRF.

	Google-Ref	UNC			UNC+			ReferItGame
	val	val	testA	testB	val	testA	testB	test
[12, 13]	28.14	-	-	-	-	-	-	48.03
R+LSTM	28.60	38.74	39.18	39.01	26.25	26.95	24.57	54.01
R+RMI	**32.06**	**39.74**	**39.99**	**40.44**	**27.85**	**28.69**	**26.65**	**54.55**
R+LSTM+DCRF	28.94	39.88	40.44	40.07	26.29	27.03	24.44	55.90
R+RMI+DCRF	**32.85**	**41.17**	**41.35**	**41.87**	**28.26**	**29.16**	**26.86**	**56.61**
D+LSTM	33.08	43.27	43.60	43.31	28.42	28.57	27.70	56.83
D+RMI	**34.40**	**44.33**	**44.74**	**44.63**	**29.91**	**30.37**	**29.43**	**57.34**
D+LSTM+DCRF	33.11	43.97	44.25	44.07	28.07	28.29	27.44	58.20
D+RMI+DCRF	**34.52**	**45.18**	**45.69**	**45.57**	**29.86**	**30.48**	**29.50**	**58.73**

Google-Ref contains 104560 expressions referring to 54822 objects from 26711 images selected from MS COCO [21]. These images all contain 2 to 4 objects of the same type. In general the expressions are longer and with richer descriptions, with an average length of 8.43 words. Although the dataset has primarily been used for referring object detection [26, 37, 29], where the goal is to return a bounding box of the referred object, it is also suitable for referring image segmentation, since the original MS COCO annotation contains segmentation masks. We use the same data split as [26].

UNC and UNC+ are also based on MS COCO images. Different from Google-Ref, these two datasets are collected interactively in a two-player game [15]. The difference between the two datasets is in UNC no restrictions are enforced on the referring expression, while in UNC+ no location words are allowed in the expression, meaning the annotator has to describe the object purely by its appearance. UNC consists of 142209 referring expressions for 50000 objects in 19994 images, and UNC+ consists of 141564 expressions for 49856 objects in 19992 images. We use the same data split as [37].

ReferItGame contains 130525 expressions referring to 96654 distinct objects in 19894 natural images. Different from the other three datasets, ReferItGame contains "stuff" segmentation masks, such as "sky" and "water", in addition to objects. In general the expressions are shorter and more concise, probably due to the collection process as a two-player game. We use the same data split as [12].

4.2. Implementation Details

[12, 13] both use the VGG network [32] pretrained on ImageNet [5] as visual feature extractor. We instead experiment with two alternatives: ResNet-101 pretrained on ImageNet, and DeepLab-101 finetuned on Pascal VOC [7].

In our experiments, we resize (while keeping aspect ratio) and pad (with zero) all images and ground truth segmentation to $W \times H$, and in all our experiments $W = H = 320$. As for the feature resolution $W' = H' = 40$. The image feature has dimension $D_I = 1000$, and the sentence vector has dimension $D_S = 1000$. We choose the cell size of mLSTM to be 500. For referring expressions of length more than 20, we only keep the first 20 words. All architecture details are in Fig. 2 3, where sizes of blobs are marked.

In [12] a three-stage training strategy is used. A detection network is first trained, which is used to initialize the low resolution version of the model. After training the low resolution version with $W' = H' = 16$, it is again used to initialize the high resolution version, where a deconvolution layer is learned. We instead only train once using the loss function defined in Equation 6, and observe fast convergence. This is probably due to the higher spatial resolution allowed by atrous convolution. We use the Adam [17] optimizer with a fixed learning rate of 0.00025. We set the batch size to 1 and weight decay to 0.0005.

We evaluate using two metrics: Precision@X ($X \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$) and Intersection-over-Union (IOU), where Precision@X means the percentage of images with IOU higher than X. This is consistent with previous work [12, 13] to allow for comparison. We report the most indicative IOU in the main paper, and the full tables containing Precision numbers are in the supplementary material.

In addition to evaluating the direct segmentation output, we also report results after applying DenseCRF [18] for refinement. We use the same hyperparameters used in [3].

4.3. Quantitative Results

The segmentation performance (IOU) on all datasets are summarized in Table 1.

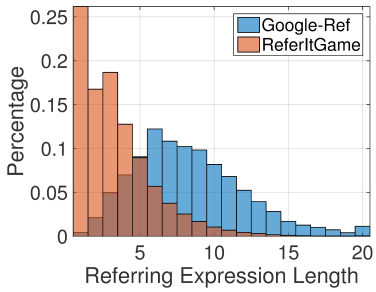


Figure 4: The distribution of referring expression length in the Google-Ref and ReferItGame test set. Most of the referring expressions in ReferItGame are short, with over 25 percent single word description. The distributions of UNC and UNC+ are very similar to that of ReferItGame since the data collection method is the same.

We first observe that the performance consistently increases by replacing the VGG-based FCN-32s with ResNet. This indicates that ResNet can provide better image features for segmentation purpose, which likely comes from both stronger network and higher spatial resolution. DeepLab delivers even higher baseline since its weights have been finetuned on segmentation datasets, which makes the knowledge transfer easier.

We then study the effect of mLSTM. Our RMI models with mLSTM consistently outperform those with language-only LSTM by a large margin regardless of the image feature extractor and dataset. This shows that mLSTM can successfully generate multimodal features that improve segmentation. Specifically, on the Google-Ref dataset using ResNet weights, we observe an IOU increase of nearly 3.5% over the baseline model.

By comparison, the performance increase using mLSTM is not as high on ReferItGame. One reason is that the dataset is easier as indicated by the metrics (over 20 percent higher IOU than Google-Ref), and the baseline model already performs well. Another reason is that the descriptions in this dataset are in general much shorter (see Fig. 4), and as a result sequential modeling does not have as much effect. In fact, over 25 percent images in the ReferItGame test set only has one word as its description.

Another interesting observation is that the performance is considerably worse on UNC+ than on UNC (over 10 percent IOU difference). As aforementioned, the only difference between the two datasets is in UNC+ there is no spatial/location indicator words that the model can utilize, and the model must understand the semantics in order to output correct segmentation. This suggests that the LSTM language encoder may be the main barrier in referring image segmentation performance.

We further show the advantage of our mLSTM model in sequential modeling by breaking down the IOU performance. Specifically, we want to study the relationship between IOU and referring expression length. To this end, we split the test set into 4 groups of increasing referring expression length with roughly equal size, and report the individual IOU on these groups. The results are summarized in Table 2 3 4 5. Our RMI model outperforms the baseline model in every group. More interestingly, the relative gain of our RMI model over the baseline model in general increases with the length of the referring expression. This suggests that mLSTM is better at fusing features over longer sequences, which we will also verify visually.

Finally, by applying the DenseCRF, we observe consistent improvement in terms of IOU. In addition, the IOU improvement on our RMI model is usually greater than the IOU improvement on the baseline model, suggesting that our model has better localization ability.

Table 2: IOU performance break-down on Google-Ref.

Length	**1-5**	**6-7**	**8-10**	**11-20**
R + LSTM	32.29	28.27	27.33	26.61
R + RMI	35.34	31.76	30.66	30.56
Relative Gain	9.44%	12.37%	12.17%	14.81%

Table 3: IOU performance break-down on UNC.

Length	**1-2**	**3**	**4-5**	**6-20**
R + LSTM	43.66	40.60	33.98	24.91
R + RMI	44.51	41.86	35.05	25.95
Relative Gain	1.94%	3.10%	3.15%	4.19%

Table 4: IOU performance break-down on UNC+.

Length	**1-2**	**3**	**4-5**	**6-20**
R + LSTM	34.40	24.04	19.31	12.30
R + RMI	35.72	25.41	21.73	14.37
Relative Gain	3.84%	5.67%	12.55%	16.85%

Table 5: IOU performance break-down on ReferItGame.

Length	**1**	**2**	**3-4**	**5-20**
R + LSTM	67.64	52.26	44.87	33.81
R + RMI	68.11	52.73	45.69	34.53
Relative Gain	0.69%	0.90%	1.82%	2.10%

4.4. Qualitative Results

As aforementioned, our RMI model is better than the baseline model in modeling long sequences. We can see from the examples in Fig. 5 that the language-only LSTM is more easily distracted by words at the end of the sentence, resulting in unsatisfactory segmentation, while our model remains unaffected.

We suspect the reason is because our model can turn seg-

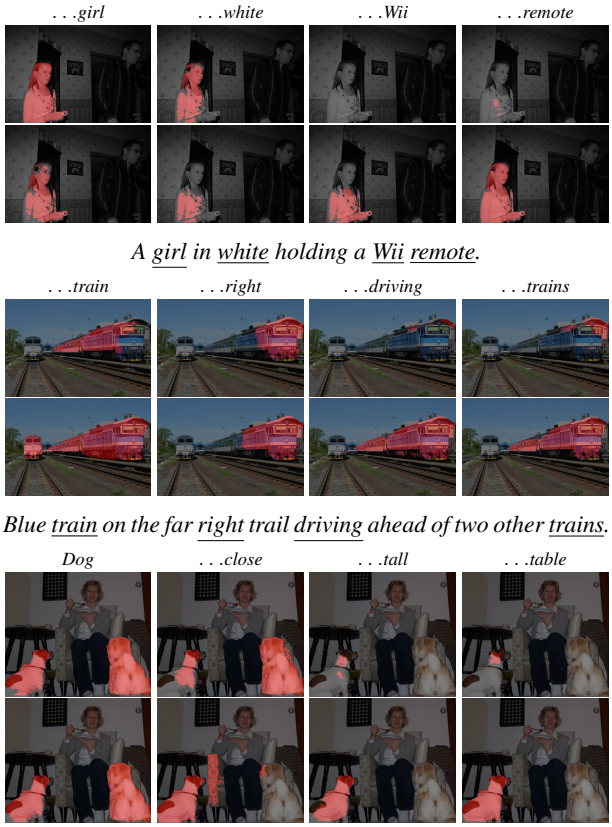


Figure 5: Comparison of D+LSTM+DCRF (first row) and D+RMI+DCRF (second row). Each column shows segmentation result until after reading the underlined word.

mentation into a sequential process, saving the burden on the LSTM hidden state to encode the entire sentence. We are therefore interested in visualizing how the multimodal LSTM hidden state progresses over time. Each mLSTM hidden state is a feature tensor of size $H' \times W' \times 500$. We visualize this tensor by first doing bilinear interpolation and then collapsing the feature dimension via meanpooling, generating a $H \times W$ response map. We provide three examples in Fig. 6. In the first example, after reading only "The bottom", the model is not sure about what objects is to be referred, and pays general attention to the bottom half of the image. As soon as it reads "luggage", the response map pinpoints the objects, and remembers the information until the end of the sentence to generate the correct segmentation output. In the second example, in the beginning after reading "The", the model appears unsure. After reading "The small vase", it discards the largest vase and focuses on the other two. As soon as the language mentions "middle", the response in the middle is enhanced, and retained till the end of the expression. In the third example there is no location words, but the response around the correct region gradually enhances with "leather" and "chair", and the response on people is gradually suppressed after reading more words. We can see that the mLSTM is successful at learning meaningful multimodal feature interaction in a sequential fashion that is consistent with our intuition in the introduction section. The meaningful multimodal features make it easier for the last convolution layer to do binary segmentation.

In Fig. 7 we provide some qualitative results of referring image segmentation on the four datasets. For Google-Ref, the language understanding is more challenging. In addition to handling longer sequences, it also needs to cope with all kinds of high level reasoning, e.g. "turning around a corner", and potentially redundant information, e.g. "listening to his music". For UNC, the expression is much shorter, and spatial words are allowed, e.g. "on left". For UNC+, the expression is more challenging. The image region could have just been described as "boy on right", but instead the model needs to reason from attributes like "strip shirt" and "eyes closed". For ReferItGame, the segmentation target is more flexible as it contains "stuff" segments in addition to objects. We show that by propagating the multimodal feature, our RMI model can better keep the intermediate belief, usually resulting in a more complete segmentation result. The effect of DenseCRF is also clearly demonstrated. For example, for the first image, DenseCRF can better refine the D+RMI result to align the prediction to the edges, and for the third image, DenseCRF can suppress the scattered wrong prediction in the D+LSTM result.

5. Conclusion

In this work we study the challenging problem of referring image segmentation. Learning a good multimodal representation is essential in this problem, since segmentation represents the correspondence or consistency between images and language. Unlike previous work, which encodes the referring expression and image into vector representation independently, we build on the observation that referring image segmentation is a sequential process, and perform multimodal feature fusion after seeing every word in the referring expression. To this end we propose the Recurrent Multimodal Interaction model, a novel two-layer recurrent architecture that encodes the sequential interactions between individual words, visual information, and spatial information as its hidden state.

We show the advantage of our word-to-image scheme over the sentence-to-image scheme. Our model achieves the new state-of-the-art on all large-scale benchmark datasets. In addition, we visualize the mLSTM hidden state and show that the learned multimodal feature is human-interpretable and facilitates segmentation. In the future we plan to introduce more structure in language understanding.

Acknowledgments We gratefully acknowledge support from NSF CCF-1231216 and a gift from Adobe.

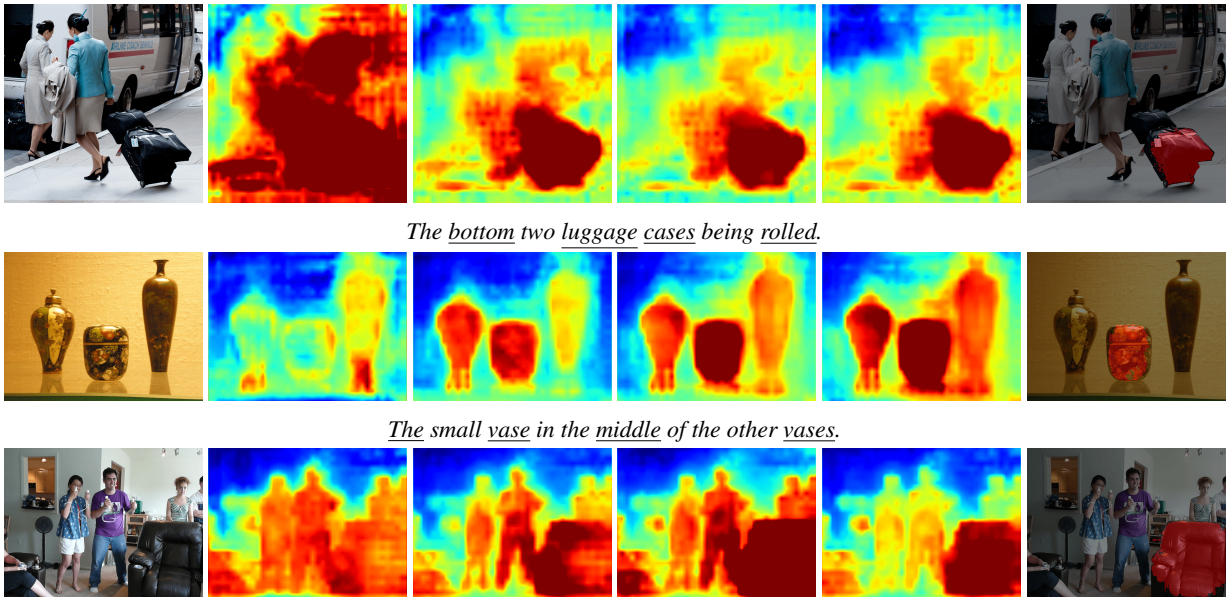


Figure 6: Visualizing and understanding convolutional multimodal LSTM in our RMI model. The first column is the original image, and the last column is the final segmentation output of D+RMI+DCRF. The middle columns visualize the output of mLSTM at underlined words by meanpooling the 500-dimensional feature.

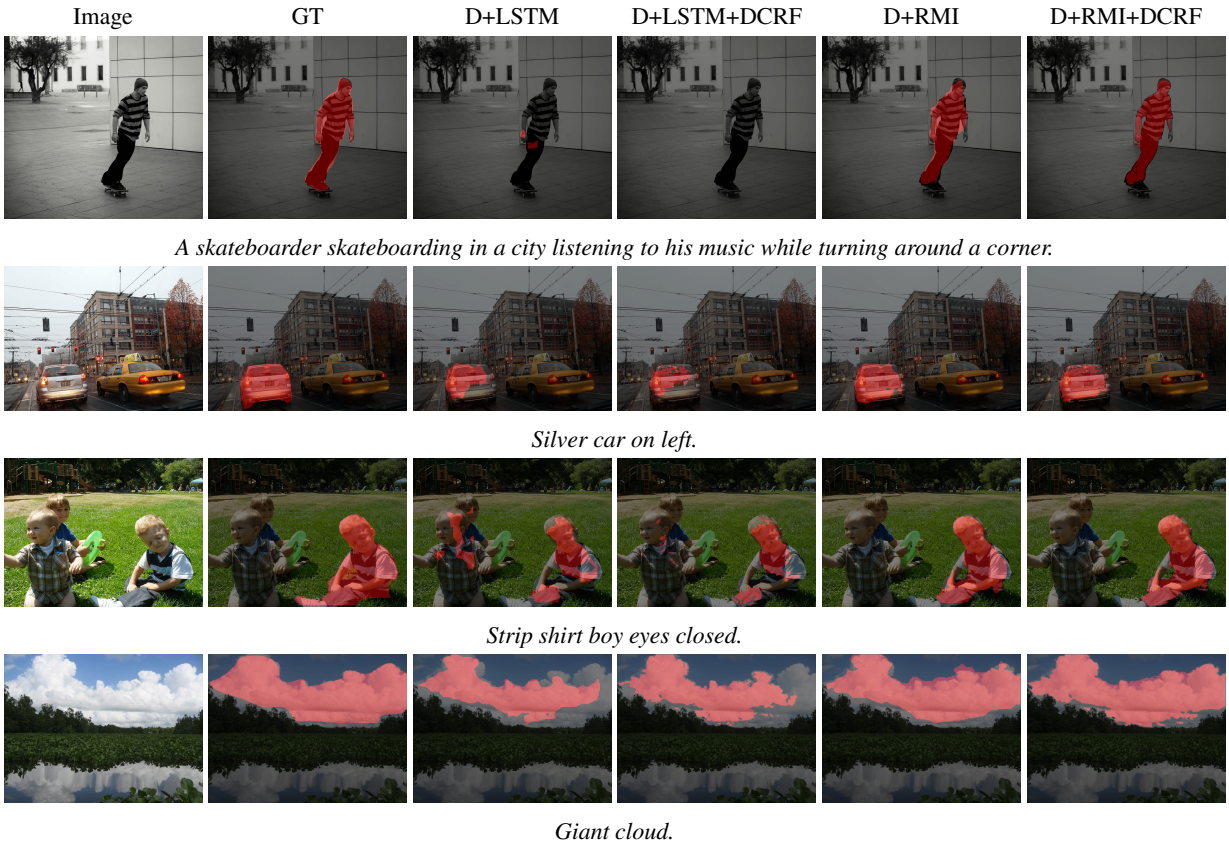


Figure 7: Qualitative results of referring image segmentation. From top down are images from Google-Ref, UNC, UNC+, ReferItGame respectively.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: visual question answering. In *ICCV*, pages 2425–2433. IEEE Computer Society, 2015. 1

[2] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2014. 1, 2, 3

[3] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *CoRR*, abs/1606.00915, 2016. 1, 2, 3, 5

[4] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV (8)*, volume 9912 of *Lecture Notes in Computer Science*, pages 628–644. Springer, 2016. 2

[5] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE Computer Society, 2009. 5

[6] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, pages 2625–2634. IEEE Computer Society, 2015. 2

[7] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 2, 5

[8] C. Finn, I. J. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *NIPS*, pages 64–72, 2016. 2

[9] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, pages 457–468. The Association for Computational Linguistics, 2016. 2

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016. 2, 3

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 1

[12] R. Hu, M. Rohrbach, and T. Darrell. Segmentation from natural language expressions. In *ECCV (1)*, volume 9905 of *Lecture Notes in Computer Science*, pages 108–124. Springer, 2016. 1, 2, 3, 5

[13] R. Hu, M. Rohrbach, S. Venugopalan, and T. Darrell. Utilizing large scale vision and text datasets for image segmentation from referring expressions. *CoRR*, abs/1608.08305, 2016. 5

[14] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *CVPR*, pages 4555–4564. IEEE Computer Society, 2016. 2

[15] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798. ACL, 2014. 4, 5

[16] J. Kim, S. Lee, D. Kwak, M. Heo, J. Kim, J. Ha, and B. Zhang. Multimodal residual learning for visual QA. In *NIPS*, pages 361–369, 2016. 2

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[18] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, pages 109–117, 2011. 2, 5

[19] J. Kuen, Z. Wang, and G. Wang. Recurrent attentional networks for saliency detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 3668–3677. IEEE Computer Society, 2016. 2

[20] G. Laput, M. Dontcheva, G. Wilensky, W. Chang, A. Agarwala, J. Linder, and E. Adar. Pixeltone: a multimodal interface for image editing. In *CHI*, pages 2185–2194. ACM, 2013. 1

[21] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *ECCV (5)*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. 5

[22] C. Liu, J. Mao, F. Sha, and A. L. Yuille. Attention correctness in neural image captioning. In S. P. Singh and S. Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 4176–4182. AAAI Press, 2017. 2

[23] C. Liu, F. Sun, C. Wang, F. Wang, and A. L. Yuille. MAT: A multimodal attentive translator for image captioning. *CoRR*, abs/1702.05658, 2017. 2

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440. IEEE Computer Society, 2015. 1, 2, 3

[25] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, pages 1–9. IEEE Computer Society, 2015. 2

[26] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, pages 11–20. IEEE Computer Society, 2016. 2, 4, 5

[27] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *CoRR*, abs/1412.6632, 2014. 2

[28] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2204–2212, 2014. 2

[29] V. K. Nagaraja, V. I. Morariu, and L. S. Davis. Modeling context between objects for referring expression understanding. In *ECCV (4)*, volume 9908 of *Lecture Notes in Computer Science*, pages 792–807. Springer, 2016. 2, 5

[30] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by re-

construction. In *ECCV (1)*, volume 9905 of *Lecture Notes in Computer Science*, pages 817–834. Springer, 2016. 2

[31] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*, pages 802–810, 2015. 2

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 2, 5

[33] G. Underwood, L. Jebbett, and K. Roberts. Inspecting pictures for information to verify a sentence: Eye movements in general encoding and in focused search. *Quarterly Journal of Experimental Psychology Section A*, 57(1):165–182, 2004. 1

[34] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2048–2057. JMLR.org, 2015. 2

[35] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. In *CVPR*, pages 21–29. IEEE Computer Society, 2016. 2

[36] Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen, and R. Salakhutdinov. Review networks for caption generation. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2361–2369, 2016. 2

[37] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *ECCV (2)*, volume 9906 of *Lecture Notes in Computer Science*, pages 69–85. Springer, 2016. 2, 4, 5

[38] Y. Zhu, O. Groth, M. S. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, pages 4995–5004. IEEE Computer Society, 2016. 1