Language-Based Image Editing with Recurrent Attentive Models

Jianbo Chen*, Yelong Shen†, Jianfeng Gao†, Jingjing Liu†, Xiaodong Liu†
University of California, Berkeley* and Microsoft Research†
jianbochen@berkeley.edu
yeshen, jfgao, jingjl, xiaodl@microsoft.com

Abstract

We investigate the problem of Language-Based Image Editing (LBIE). Given a source image and a natural language description, we want to generate a target image by editing the source image based on the description. We propose a generic modeling framework for two sub-tasks of LBIE: language-based image segmentation and image colorization. The framework uses recurrent attentive models to fuse image and language features. Instead of using a fixed step size, we introduce for each region of the image a termination gate to dynamically determine after each inference step whether to continue extrapolating additional information from the textual description. The effectiveness of the framework is validated on three datasets. First, we introduce a synthetic dataset, called CoSaL, to evaluate the end-to-end performance of our LBIE system. Second, we show that the framework leads to state-of-the-art performance on image segmentation on the ReferIt dataset. Third, we present the first language-based colorization result on the Oxford-102 Flowers dataset.

1. Introduction

In this work, we aim to develop an automatic Language-Based Image Editing (LBIE) system. Given a source image, which can be a sketch, a grayscale image or a natural image, the system will automatically generate a target image by editing the source image following natural language instructions provided by users. Such a system has a wide range of applications from Computer-Aided Design (CAD) to Virtual Reality (VR). As illustrated in Figure 1, a fashion designer presents a sketch of a pair of new shoes (i.e., the source image) to a customer, who can provide modifications on the style and color in verbal description, which can then be taken by the LBIE system to change the original design. The final output (i.e., the target image) is the revised and enriched design that meets the customers requirement. Figure 2 showcases the use of LBIE for VR. While most VR systems still use button-controlled or touchscreen interface, LBIE provides a natural user interface for future VR systems, where users can easily modify the virtual environment via natural language instructions.

Figure 1. In an interactive design interface, a sketch of shoes is presented to a customer, who then gives a verbal instruction on how to modify the design: "*The insole of the shoes should be brown. The vamp and the heel should be purple and shining*". The system colorizes the sketch following the customer's instruction. (images from [11]).

Figure 2. The image on the left is an initial virtual environment. The user provides a textual description: "The afternoon light flooded the little room from the window, shining the ground in front of a brown bookshelf made of wood. Besides the bookshelf lies a sofa with light-colored cushions. There is a blue carpet in front of the sofa, and a clock with dark contours above it...". The system modifies the virtual environment into the target image on the right.

Figure 3. Left: sketch image. Middle: grayscale image. Right: color image (from [18]). A language-based image editing system will take either of the first two images as the input, and generate the third color image following a natural language expression: *"The flower has red petals with yellow stigmas in the middle"*,.

LBIE covers a broad range of tasks in image generation: shape, color, size, texture, position, etc. This paper focuses on two basic sub-tasks: language-based segmentation and colorization for shapes and colors. As shown in

Figure 3, given a grayscale image and the expression *"The flower has red petals with yellow stigmas in the middle"*, the segmentation model will identify regions of the image as "*petals*", "*stigmas*", and the colorization model will paint each pixel with the suggested color. In this intertwined task of segmentation and colorization, the distribution of target images can be multi-modal in the sense that each pixel will have a definitive ground truth on segmentation, but not necessarily on color. For example, the pixels on petals in Figure 3 should be red based on the textual description, but the specific numeric values of the red color in the RGB space is not uniquely specified. The system is required to colorize the petals based on real-world knowledge. Another uncertainty lies in the fact that the input description might not cover every detail of the image. The regions that are not described, such as the leaves in the given example, need to be rendered based on common sense knowledge. In summary, we aim to generate images that not only are consistent with the natural language expressions, but also align with common sense.

Language-based image segmentation has been studied previously in [9]. However, our task is far more challenging because the textual description often contains multiple sentences (as in Figure 2), while in [9] most of the expressions are simple phrases. To the best of our knowledge, language-based colorization has not been studied systematically before. In most previous work, images are generated either solely based on natural language expressions [21],[32] or based on another image [11],[3],[33]. Instead, we want to generate a target image based on both the natural language expression and the source image. Related tasks will be discussed in detail in Section 2.

A unique challenge in language-based image editing is the complexity of natural language expressions and their correlation with the source images. As shown in Figure 2, the description usually consists of multiple sentences, each referring to multiple objects in the source image. When human edits the source image based on a textual description, we often keep in mind which sentences are related to which region/object in the image, and go back to the description multiple times while editing that region. This behavior of "going back" often varies from region to region, depending on the complexity of the description for that region. An investigation of this problem is carried out on CoSaL, which is a synthetic dataset described in Section 4.

Our goal is to design a generic framework for the two sub-tasks in language-based image editing. A diagram of our model is shown in Figure 4. Inspired by the observation aforementioned, we introduce a recurrent attentive fusion module in our framework. The fusion module takes as input the image features that encode the source image via a convolutional neural network, and the textual features that encode the natural language expression via an LSTM, and outputs the fused features to be upsampled by a deconvolutional network into the target image. In the fusion module, recurrent attentive models are employed to extract distinct textual features based on the spatial features from different regions of an image. A termination gate is introduced for each region to control the number of steps it interacts with the textual features. The Gumbel-Softmax reparametrization trick [12] is used for end-to-end training of the entire network. Details of the models and the training process are described in Section 3.

Our contributions are summarized as follows:

- We define a new task of language-based image editing (LBIE).
- We present a generic modeling framework based on recurrent attentive models for two sub-tasks of LBIE: language-based image segmentation and colorization.
- We introduce a synthetic dataset CoSaL designed specifically for the LBIE task.
- We achieve new state-of-the-art performance on language-based image segmentation on the ReferIt dataset.
- We present the first language-based colorization result on the Oxford-102 Flowers dataset, with human evaluations validating the performance of our model.

2. Related Work

While the task of language-based image editing has not been studied, the community has taken significant steps in several related areas, including Language Based object detection and Segmentation (LBS) [9],[10], Image-to-Image Translation (IIT) [11], Generating Images from Text (GIT) [20], [32], Image Captioning (IC) [13], [25], [30], Visual Question Answering (VQA) [2], [31], Machine Reading Comprehension (MRC) [8], etc. We summarize the types of inputs and outputs for these related tasks in Table 1.

	Inputs		Outputs	
	Text	Image	Text	Image
MRC	YES	NO	YES	NO
VQA	YES	YES	YES	NO
IIT	NO	YES	NO	YES
IC	NO	YES	YES	NO
GIT	YES	NO	NO	YES
LBS	YES	YES	NO	YES
LBIE	**YES**	**YES**	**NO**	**YES**

Table 1. The types of inputs and outputs for related tasks

Recurrent attentive models

Recurrent attentive models have been applied to visual question answering (VQA) to fuse language and image

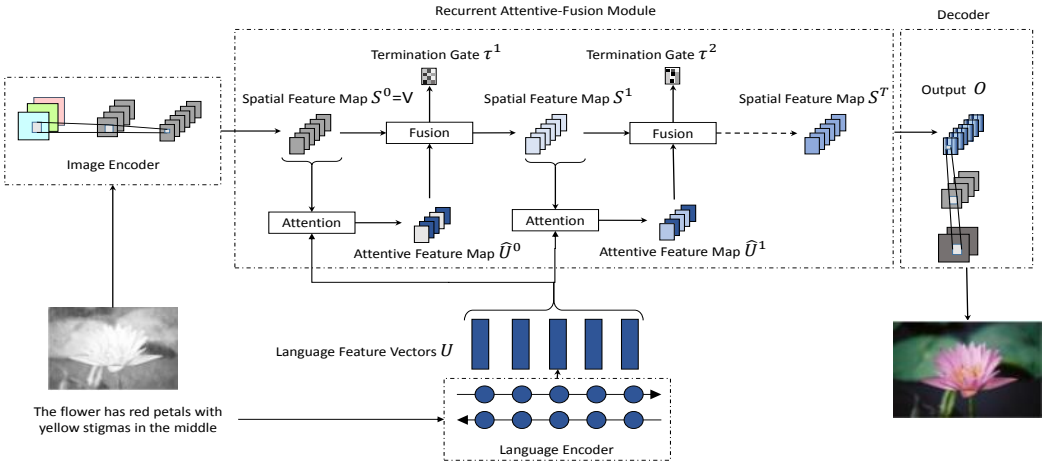

Figure 4. A high-level diagram of our model, composed of a convolutional image encoder, an LSTM text encoder, a fusion module, a deconvolutional upsampling layer, with an optional convolutional discriminator.

features [31]. The stacked attention network proposed in [31] identifies the image regions that are relevant to the question via multiple attention layers, which can progressively filter out noises and pinpoint the regions relevant to the answer. In image generation, a sequential variational auto-encoder framework, such as DRAW[7], has shown substantial improvement over standard variational auto-encoders (VAE) [15]. Similar ideas have also been explored for machine reading comprehension, where models can take multiple iterations to infer an answer based on the given query and document [4], [27], [26], [29], [16]. In [23] and [22], a novel neural network architecture called ReasoNet is proposed for reading comprehension. ReasoNet performs multi-step inference where the number of steps is determined by a termination gate according to the difficulty of the problem. ReasoNet is trained using policy gradient methods.

Segmentation from language expressions

The task of language-based image segmentation is first proposed in [9]. Given an image and a natural language description, the system will identify the regions of the image that correspond to the visual entities described in the text. The authors in [9] proposed an end-to-end approach that uses three neural networks: a convolutional network to encode source images, an LSTM network to encode natural language descriptions, and a fully convolutional classification and upsampling network for pixel-wise segmentation.

One of the key differences between their approach and ours is the way of integrating image and text features. In [9], for each region in the image, the extracted spatial features are concatenated with the same textual features. Inspired by the alignment model of [13], in our approach, each spatial feature is aligned with different textual features based on attention models. Our approach yields superior segmentation results than that of [9] on a benchmark dataset.

Conditional GANs in image generation

Generative adversarial networks (GANs) [6] have been widely used for image generation. Conditional GANs [17] are often employed when there are constraints that a generated image needs to satisfy. For example, deep convolutional conditional GANs [19] have been used to synthesize images based on textual descriptions [21] [32]. [11] proposed the use of conditional GANs for image-to-image translation. Different from these tasks, LBIE takes both image and text as input, presenting an additional challenge of fusing the features of the source image and the textual description.

3. The Framework

Overview The proposed modeling framework, as shown in 4, is based on neural networks, and is generic to both the language-based image segmentation and colorization tasks. The framework is composed of a convolutional image encoder, an LSTM text encoder, a fusion network that generates a fusion feature map by integrating image and text features, a deconvolutional network that generates pixel-wise outputs (the target image) by upsampling the fusion feature map, and an optional convolutional discriminator used for training colorization models.

Image encoder The image encoder is a multi-layer convolutional neural network (CNN). Given a source image of size $H \times W$, the CNN encoder produces a $M \times N$ spatial feature map, with each position on the feature map containing a D-dimensional feature vector (D channels), $V = \{v_i : i = 1, \ldots, M \times N\}, v_i \in \mathbb{R}^D$.

Language encoder The language encoder is a recurrent Long Short-Term Memory (LSTM) network. Given a natural language expression of length L, we first embed each word into a vector through a word embedding matrix, then use LSTM to produce for each word a contextual vector that encodes its contextual information such as word order and word-word dependencies. The resulting language feature map is $U = \{u_i : i = 1, \ldots, L\}, u_i \in \mathbb{R}^K$.

Recurrent attentive fusion module The fusion network fuses text information in U into the $M \times N$ image feature map V, and outputs an $M \times N$ fusion feature map, with each position (image region) containing an editing feature vector, $O = \{o_i : i = 1, \ldots, M \times N\}, o_i \in \mathbb{R}^D$.

The fusion network is devised to mimic the human image editing process. For each region in the source image v_i, the fusion network reads the language feature map U repeatedly with attention on different parts each time until enough editing information is collected to generate the target image region. The number of steps varies from region to region.

Internal state The internal state at time step t is denoted as $S^t = \{s_i^t, i = 1, \ldots, M \times N\}$, which is a spatial feature map, with each poition (image region) containing a vector representation of the editing information state. The initial state is the spatial feature map from the source image, $S^0 = V$. The sequence of internal states is modeled by Convolutional Gated Recurrent Units (C-GRUs) which will be described below.

Attention The attention at time step t is denoted as $\hat{U}^t = \{\hat{u}_i^t, i = 1, \ldots, M \times N\}$, which is a spatial feature map generated based on the current internal state S^t and the language feature map U:

$$\hat{U}^t = \textbf{Attention}(U, S^t; \theta_a),$$

where **Attention**(.) is implemented as follows:

$$\beta_{ij} \propto \exp\{{s_i^t}^T W u_j\},$$

$$\hat{u}_i^t = \sum_{j=1}^{L} \beta_{ij} u_j.$$

C-GRUs C-GRUs update the current internal state S^t by infusing the attention feature map $\hat{U}^t$:

$$S^{t+1} = \textbf{C-GRUs}(S^t, \hat{U}^t; \theta_c).$$

The **C-GRUs**(.) is implemented as follows:

$$\mathbf{z} = \sigma(W_1 \otimes S^t + W_2 \otimes \hat{U}^t + b_1),$$

$$\mathbf{r} = \sigma(W_3 \otimes S^t + W_4 \otimes \hat{U}^t + b_2),$$

$$\mathbf{c} = \text{ReLU}(W_5 \otimes (\mathbf{r} \odot S^t) + W_6 \otimes \hat{U}^t + b),$$

$$\hat{O}^t = \mathbf{h} = (1 - \mathbf{z}) \odot S^t + \mathbf{z} \odot \mathbf{c},$$

$$S^{t+1} = W_7 \otimes \mathbf{h},$$

where $\odot$ is the elementwise-product, and $\otimes$ is the convolutional operator. Note that $\hat{O}^t$ is the intermediate output of the fusion feature map at time step t.

Termination gates There are $M \times N$ termination gates, each for one image region v_i in V. Each termination gate generates a binary random variable according to the current internal state of its image region: $\tau_i^t \sim p(\cdot f_{tg}(s_i^t; \theta_{tg}))$. If $\tau_i^t = 1$, the fusion process for the image region v_i stops at t, and the editing feature vector for this image region is set as $o_i = \hat{o}_i^t$. When all terminate gates are true, the fusion process for the entire image is completed, and the fustion network outputs the fusion feature map O. We define $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \ldots, \zeta_{M \times N})$, where $\zeta_i = (\tau_i^1, \tau_i^2, \ldots, \tau_i^T)$, a categorical distribution with $p(\zeta_i = e_t) = \beta_i^t$, where

$$\beta_i^t = f_{tg}(s_i^t; \theta_{tg}) \prod_{k<t} (1 - f_{tg}(s_i^k; \theta_{tg})).$$

the probability of stopping the fusion process at the i-th image region of the feature map at time t.

Inference Algorithm 1 describes the stochastic inference process of the fusion network. The state sequence $S^{(1:T)}$ is hidden and dynamic, chained through attention and C-GRU in a recurrent fashion. The fusion network outputs for each image region v_i an editing feature vector o_i at the t_i-th step, where t_i is controlled by the ith termination gate, which varies from region to region.

Algorithm 1 Stochastic Inference of the Fusion Network

Require: $V \in \mathbb{R}^{D \times (M \times N)}$: Spatial feature map of image.
Require: $U \in \mathbb{R}^{K \times L}$: Language feature map of expression.
Ensure: Fusion feature map $O \in \mathbb{R}^{D \times (M \times N)}$.

```
function FUSION(V, U)
    Initialize S^0 = V.
    for all t = 0 to t_max − 1 do
        Û^t = Attention(U, S^t; θ_a)
        S^{t+1}, Ô^t = C-GRUs(S^t, Û^t; θ_c)
        Sample τ^{t+1} ~ p(·|f_tg(S^{t+1}; θ_tg))
        if τ_i^{t+1} = 1 and τ_i^s = 0 for s ≤ t then
            Set O_i = Ô_i^{t+1}.
        end if
    end for
    for all i = 1 to M × N do
        if τ_i = 0 then
            Set o_i = ô_i^{t_max − 1}
        end if
    end for
end function
```

Image decoder The image decoder is a multi-layer deconvolutional network. It takes as input the $M \times N$ fusion feature map O produced by the fusion module, and unsamples from O to produce a $H \times W \times D_e$ editing map E of the same size as the target image, where D_e is the number of classes in segmentation and 2 (ab channels) in colorization.

Discriminator The discriminator $D_\phi(E)$ takes in a generated image and its corresponding language description and outputs the probability of the image being realistic. The discriminator uses a convolutional neural network to extract features from the image, as in [21], and uses an LSTM to encode language. The language features are extracted using the attention mechanism and aligned to features extracted from each region of the image respectively. Parameters of the LSTM and the attention map are not shared with those of the previous language encoder.

Loss and training Denote the loss as $L(\theta) = \mathbb{E}_{\boldsymbol{\zeta}}[l(E(\boldsymbol{\zeta}, \theta), Y)]$, where the expectation is taken over the categorical variables $\boldsymbol{\zeta}$ generated by termination gates, and $l_\theta(\boldsymbol{\zeta}) = l(E(\boldsymbol{\zeta}, \theta), Y)$ is the loss of output at $\boldsymbol{\zeta}$, and Y is the target image (i.e., the class labels in segmentation or the ab channels in colorization). Denote the probability mass function of $\boldsymbol{\zeta}$ by $p_\theta(\boldsymbol{\zeta})$. Because the sample space is of exponential size $T^{M\times N}$, it is intractable to sum over the entire sample space. A naive approach to approximation is to subsample the loss and update parameters via the gradient of Monte Carlo estimate of loss:

$$\begin{aligned}\nabla_\theta L(\theta) &= \nabla_\theta \mathbb{E}_{\boldsymbol{\zeta}}[l_\theta(\boldsymbol{\zeta})] \\ &= \nabla_\theta \mathbb{E}_{\boldsymbol{\zeta}}[l_\theta(\boldsymbol{\zeta})] \\ &= \sum_{\boldsymbol{\zeta}} p_\theta(\boldsymbol{\zeta}) \big(l_\theta(\boldsymbol{\zeta}) \nabla_\theta \log p_\theta(\boldsymbol{\zeta})\big) + \nabla_\theta l_\theta(\boldsymbol{\zeta})\big) \\ &\approx \frac{1}{|\mathbf{S}|} \sum_{\boldsymbol{\zeta}\in\mathbf{S}} l_\theta(\boldsymbol{\zeta}) \nabla_\theta \log p_\theta(\boldsymbol{\zeta}) + \nabla_\theta l_\theta(\boldsymbol{\zeta}),\end{aligned}$$

where $\mathbf{S}$ is a subset of $\boldsymbol{\zeta}$ sampled from the distributon $p_\theta(\boldsymbol{\zeta})$. The above update is called a REINFORCE-type algorithm [28]. In experiments, we found that the above Monte Carlo estimate suffers from high variance. To resolve this issue, we employ the Gumbel-Softmax reparameterization trick [12], which replaces every $\zeta_i \in \{0,1\}^T$ sampled from $\text{Cat}(\beta_1, \beta_2, \ldots, \beta_T)$ by another random variable z_i generated from Gumbel-Softmax distribution:

$$z_i^t = \frac{\exp((\log \beta_i^t + \varepsilon_i^t)/\lambda)}{\sum_{k=1}^T \exp((\log \beta_i^t + \varepsilon_i^t)/\lambda)},$$

where λ is a temperature annealed via a fixed schedule and the auxiliary random variables $\varepsilon_i^1, \ldots, \varepsilon_i^T$ are i.i.d. samples drawn from $\text{Gumbel}(0,1)$ independent of the parameters β_i:

$$\varepsilon_i^t = -\log(-\log u_i^t), u_i^t \sim \text{Unif}(0,1).$$

Define $\mathbf{z}(\boldsymbol{\varepsilon}, \theta) = (z_1, z_2, \ldots, z_{MN})$. The loss can be rewritten as $L(\theta) = \mathbb{E}_{\boldsymbol{\varepsilon}}[l_\theta(\boldsymbol{z}(\boldsymbol{\varepsilon}, \theta))]$, and the update is approximated by taking the gradient of Monte Carlo estimates of the loss obtained from sampling $\boldsymbol{\varepsilon}$.

We use two different losses for segmentation and colorization, respectively.

Segmentation In segmentation, we assume there is a unique answer for each pixel on whether or not it is being referred in the stage of segmentation. The response map E is of size $H \times W \times D_e$, which produces a log probability for each class for each pixel. We use a pixel-wise softmax cross-entropy loss during training:

$$l(E, Y) = \text{Cross-Entropy}(\text{Softmax}(E), Y).$$

Colorization In colorization, the high-level goal is to generate realistic images under the constraint of natural language expressions and input scene representations, we introduce a mixture of GAN loss and $L1$ loss for optimization as in [11]. A discriminator D_ϕ parametrized by ϕ is introduced for constructing the GAN loss.

The response map E is the predicted ab color channels. It is combined with the grayscale source image to produce a generated color image E'. The generator loss is a GAN loss taking E' as input, and $L1$ loss between the ab channels of the target image Y and the response map E:

$$l(E, Y) = \log(1 - D_\phi(E)) + \gamma \|E - Y\|_1 \ (\gamma = 0.01).$$

The discriminator D_ϕ is trained by first generating a sample E via Algorithm 1, combined with the grayscale source image to produce E', and optimize the following loss over ϕ:

$$\log(D_\phi(E')) + \log(1 - D_\phi(Y)).$$

The generator loss and the discriminator loss are optimized alternatively in the training stage.

4. Experiments

We conducted three experiments to validate the performance of the proposed framework. A new synthetic dataset CoSaL (Colorizing Shapes with Artificial Language) was introduced to test the capability of understanding multi-sentence descriptions and associating the inferred textual features with visual features. Our framework also yielded state-of-the-art performance on the benchmark dataset ReferIt [14] for image segmentation. A third experiment was carried out on the Oxford-102 Flowers dataset [18], for the language-based colorization task. All experiments were coded in TensorFlow. Codes for reproducing the key results are available online[1].

4.1. Experiments on CoSaL

Dataset Each image in the CoSaL dataset consists of nine shapes, paired with a textual description of the image. The task is defined as: given a black-white image and its corresponding description, colorize the nine shapes following the textual description. Figure 5 shows an example. It requires sophisticated coreference resolution, multi-step inference and logical reasoning to accomplish the task.

The dataset was created as follows: first, we divide a white-background image into 3×3 regions. Each region contains a shape randomly sampled from a set of S shapes (e.g., squares, fat rectangles, tall rectangles, circles, fat

[1] https://github.com/Jianbo-Lab/LBIE

ellipses, tall ellipses, diamonds, etc.) Each shape is then filled with one of C color choices, chosen at random. The position and the size of each shape are generated by uniform random variables. As illustrated in Figure 5, the difficulty of this task increases with the number of color choices. In our experiments, we specify $C = 3$.

The descriptive sentences for each image can be divided into two categories: direct descriptions and relational descriptions. The former prescribes the color of a certain shape (e.g., *Diamond is red*), and the latter depicts one shape conditional of another (e.g., *The shape left to Diamond is blue*). To understand direct descriptions, the model needs to associate a specified shape with its textual features. Relational description adds another degree of difficulty, which requires advanced inference capability of relational/multi-step reasoning. The ratio of direct descriptions to relational descriptions varies among different images, and all the colors and shapes in each image are uniquely determined by the description. In our experiment, we randomly generated $50,000$ images with corresponding descriptions for training purpose, and $10,000$ images with descriptions for testing.

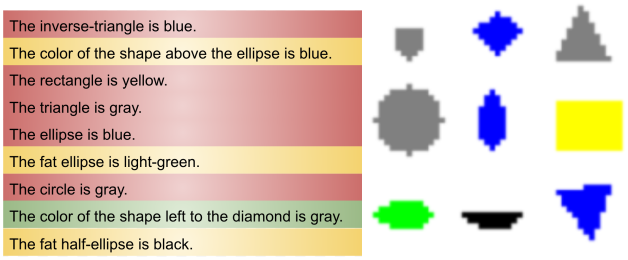

Figure 5. Right: ground truth image. Left: illustration of which sentences are attended to at each time step. Red, yellow and green represent the first, second and third time step, respectively.

		Number of direct descriptions		
T	Attention	4	6	8
1	No	0.2107	0.2499	0.3186
1	Yes	0.4030	0.5220	**0.7097**
4	Yes	**0.5033**	**0.5313**	0.7017

Table 2. The average IoU of two models, without attention at $T = 1$ and with attention at $T = 1, 4$. Performance varies among datasets with different ratios of direct to relational descriptions.

Metric For this task, we use *average IoU over nine shapes and the background* as the evaluation metric. Specifically, for each region, we compute the intersection-over-union (IoU), which is the ratio of the total intersection area to the total union area of predicted colors and ground truth colors. We also compute the IoU for the background (white) of each image. The IoU for 10 classes (9 shapes + 1 background) are computed over the entire test set and then averaged.

Model Implementation A six-layer convolutional network is implemented as the image feature extractor. Each layer has a 3×3 kernel with stride 1 and output dimension $4, 4, 8, 8, 16, 16$. ReLU is used for nonlinearity after each layer, and a max-pooling layer with a kernel of size 2 is inserted after every two layers. Each sentence in the textual description is encoded with bidirectional LSTMs that share parameters. Another LSTM with attention is put on top of the encoded sentences. The LSTMs have 16 units. In the fusion network, the attention model has 16 units, the GRU cells use 16 units, and the termination gate uses a linear map on top of the hidden state of each GRU cell. Two convolutional layers of kernel size 1×1 with the output dimension of $16, 7$ are put on top of the fused features as a classifier. Then an upsampling layer is implemented on top of it, with a single-layer deconvolutional network of kernel size 16, stride 8 to upsample the classifier to the original resolution. The upsampling layer is initialized with bilinear transforms. The maximum of termination steps T vary from 1 to 4. When $T = 1$, the model is reduced to simply concatenating features extracted from the convolutional network with the last vector from LSTM.

Results Results in Table 2 show that the model with attention and $T = 4$ achieves a better performance when there are more relational descriptions in the dataset. When there are more direct descriptions, the two models achieve similar performance. This demonstrates the framework's capability of interpreting multiple-sentence descriptions and associating them with their source image.

Figure 5 illustrates how the model with $T = 3$ interprets the nine sentences during each inference step. In each step, we take the sentence with the largest attention score as the one being attended to. Sentences in red are attended to in the first step. Those in yellow and green are attended to in the next two consecutive steps. We observe that the model tends to first extract information from direct descriptions, and then extract information from relational descriptions via reasoning.

4.2. Experiments on ReferIt

Dataset The ReferIt dataset is composed of $19,894$ photographs of real world scenes, along with $130,525$ natural language descriptions on $96,654$ distinct objects in those photographs [14]. The dataset contains 238 different object categories, including animals, people, buildings, objects and background elements (e.g., grass, sky). Both training and development datasets include $10,000$ images.

Metric Following [9], we use two metrics for evaluation: 1) *overall intersection-over-union (overall IoU)* of the predicted and ground truth of each region, averaged over the entire test set; 2) *precision@threshold*, the percentage of test data whose (per image) IoU between prediction and

Model	Precision@0.5	Precision@0.6	Precision@0.7	Precision@0.8	Precision@0.9	IoU
SCRC bbox [10]	9.73%	4.43%	1.51%	0.27%	0.03%	21.72%
GroundeR bbox [5]	11.08%	6.20%	2.74%	0.78%	0.20%	20.50%
Hu, etc.[9]	**34.02%**	26.71%	**19.32%**	11.63%	3.92%	48.03%
Our model	32.53%	**27.9%**	18.76%	**12.37%**	**4.37%**	**50.09%**

Table 3. The results of previous models and our model on the ReferIt dataset.

ground truth is above the threshold. Thresholds are set to $0.5, 0.6, 0.7, 0.8, 0.9$.

Model Implementation A VGG-16 model [24] is used as the image encoder for images of size 512×512. Textual descriptions are encoded with an LSTM of $1,024$ units. In the fusion network, the attention model uses 512 units and the GRU cells $1,024$ units, on top of which is a classifier and an upsampling layer similar to the implementation in Section 4.1. The maximum number of inference steps is 3. ReLU is used on top of each convolutional layer. $L2$-normalization is applied to the parameters of the network.

Results Table 3 shows the experimental results of our model and the previous methods on the ReferIt dataset. We see that our framework yields a better IoU and precision than [9]. We attribute the superior performance to the unique attention mechanism used by our fusion network. It efficiently associates individual descriptive sentences with different regions of the source image. There is not much discrepancy between the two models with $T = 1$ and $T = 3$, probably due to the fact that most textual descriptions in this dataset are simple.

4.3. Experiments on Oxford-102 Flower Dataset

Dataset The Oxford-102 Flowers dataset [18] contains $8,189$ images from 102 flower categories. Each image has five textual descriptions [21]. Following [21], [20] and [1], we split the dataset into 82 classes for training and 20 classes for testing. The task is defined as follows: Given a grayscale image of a flower and a description of the shapes and colors of the flower, colorize the image according to the description.

Model Implementation A 15-layer convolutional network similar to [33] is used for encoding 256×256 images. Textual descriptions are encoded with an bidirectional LSTM of 512 units. In the fusion network, the attention model uses 128 units and the GRU cells 128 units. The image encoder is composed of 2 deconvolutional layers, each followed by 2 convolutional layers, to upsample the fusion feature map to the target image space of $256 \times 256 \times 2$. The maximum length of the spatial RNN is 1. The discriminator is composed of 5 layers of convolutional networks of stride 2, with the output dimension $256, 128, 64, 32, 31$. The discriminator score is the average of the final output. ReLU is used for nonlinearity following each convolutional layer, except for the last one which uses the sigmoid function.

Setup Due to the lack of available models for the task, we compare our framework with a previous model developed for image-to-image translation as baseline, which colorizes images without text descriptions. We carried out two human evaluations using Mechanical Turk to compare the performance of our model and the baseline. For each experiment, we randomly sampled 1,000 images from the test set and then turned these images into black and white. For each image, we generated a pair of two images using our model and the baseline, respectively. Our model took into account the caption in generation while the baseline did not. Then we randomly permuted the 2,000 generated images. In the first experiment, we presented to human annotators the 2,000 images, together with their original captions, and asked humans to rate the consistency between the generated images and the captions in a scale of 0 and 1, with 0 indicating no consistency and 1 indicating consistency. In the second experiment, we presented to human annotators the same 2,000 images without captions, but asked human annotators to rate the quality of each image without providing its original caption. The quality was rated in a scale of 0 and 1, with 0 indicating low quality and 1 indicating high quality.

Results The results of comparison are shown in Table 4. Our model achieves better consistency with captions and also better image quality by making use of information in captions. The colorization results on 10 randomly-sampled images from the test set are shown in Figure 6. As we can see, without text input, the baseline approach often colorizes images with the same color (in this dataset, most images are painted with purple, red or white), while our framework can generate flowers similar to their original colors which are specified in texts. Figure 7 provides some example images generated with arbitrary text description using our model.

	Our Model	BaseLine	Truth
Consistency	**0.849**	0.27	N/A
Quality	0.598	0.404	**0.856**

Table 4. The average rate of consistency with captions and image quality for our model and the baseline model respectively, averaged over $1,000$ images. The average quality of 1,000 truth images from the data set is also provided for comparison.

Figure 6. First row: original images. Second row: results from the image-to-image translation model in [11], without text input. Third row: results from our model, taking textual descriptions into account. The textual descriptions and more examples can be found in supplementary materials.

Figure 7. First row: original images. Remaining rows: results generated from our framework with arbitrary text input: *"The flower is white/red/orange/yellow/blue/purple in color"*.

5. Conclusion and Future Work

In this paper we introduce the problem of Language-Based Image Editing (LBIE), and propose a generic modeling framework for two sub-tasks of LBIE: language-based image segmentation and colorization. At the heart of the proposed framework is a fusion module that uses recurrent attentive models to dynamically decide, for each region of an image, whether to continue the text-to-image fusion process. Our models have demonstrated superior empirical results on three datasets: the ReferIt dataset for image segmentation, the Oxford-102 Flower dataset for colorization, and the synthetic CoSaL dataset for evaluating the end-to-end performance of the LBIE system. In future, we will extend the framework to other image editing subtasks and build a dialogue-based image editing system that allows users to edit images interactively.

References

[1] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2015. 7

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 2

[3] Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015. 2

[4] B. Dhingra, H. Liu, Z. Yang, W. W. Cohen, and R. Salakhutdinov. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549*, 2016. 3

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

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3

[7] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015. 3

[8] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701, 2015. 2

[9] R. Hu, M. Rohrbach, and T. Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124. Springer, 2016. 2, 3, 6, 7

[10] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrell. Natural language object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4555–4564, 2016. 2, 7

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016. 1, 2, 3, 5, 8

[12] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 2, 5

[13] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015. 2, 3

[14] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referit game: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 5, 6

[15] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[16] X. Liu, Y. Shen, K. Duh, and J. Gao. Stochastic answer networks for machine reading comprehension. *arXiv preprint arXiv:1712.03556*, 2017. 3

[17] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3

[18] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008. 1, 5, 7

[19] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 3

[20] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016. 2, 7

[21] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069, 2016. 2, 3, 5, 7

[22] Y. Shen, P.-S. Huang, M.-W. Chang, and J. Gao. Implicit reasonet: Modeling large-scale structured relationships with shared memory. *arXiv preprint arXiv:1611.04642*, 2016. 3

[23] Y. Shen, P.-S. Huang, J. Gao, and W. Chen. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055. ACM, 2017. 3

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[25] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015. 2

[26] S. Wang and J. Jiang. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*, 2016. 3

[27] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015. 3

[28] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992. 5

[29] C. Xiong, V. Zhong, and R. Socher. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*, 2016. 3

[30] 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

[31] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016. 2, 3

[32] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1612.03242*, 2016. 2, 3

[33] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016. 2, 7