DenseCap: Fully Convolutional Localization Networks for Dense Captioning

Justin Johnson*   Andrej Karpathy*   Li Fei-Fei

Department of Computer Science, Stanford University

{jcjohns,karpathy,feifeili}@cs.stanford.edu

Abstract

We introduce the dense captioning task, which requires a computer vision system to both localize and describe salient regions in images in natural language. The dense captioning task generalizes object detection when the descriptions consist of a single word, and Image Captioning when one predicted region covers the full image. To address the localization and description task jointly we propose a Fully Convolutional Localization Network (FCLN) architecture that processes an image with a single, efficient forward pass, requires no external regions proposals, and can be trained end-to-end with a single round of optimization. The architecture is composed of a Convolutional Network, a novel dense localization layer, and Recurrent Neural Network language model that generates the label sequences. We evaluate our network on the Visual Genome dataset, which comprises 94,000 images and 4,100,000 region-grounded captions. We observe both speed and accuracy improvements over baselines based on current state of the art approaches in both generation and retrieval settings.

1. Introduction

Our ability to effortlessly point out and describe all aspects of an image relies on a strong semantic understanding of a visual scene and all of its elements. However, despite numerous potential applications, this ability remains a challenge for our state of the art visual recognition systems. In the last few years there has been significant progress in image classification [39, 26, 53, 45], where the task is to assign one label to an image. Further work has pushed these advances along two orthogonal directions: First, rapid progress in object detection [40, 14, 46] has identified models that efficiently identify and label multiple salient regions of an image. Second, recent advances in image captioning [3, 32, 21, 49, 51, 8, 4] have expanded the complexity of the label space from a fixed set of categories to sequence of words able to express significantly richer concepts.

However, despite encouraging progress along the label density and label complexity axes, these two directions have
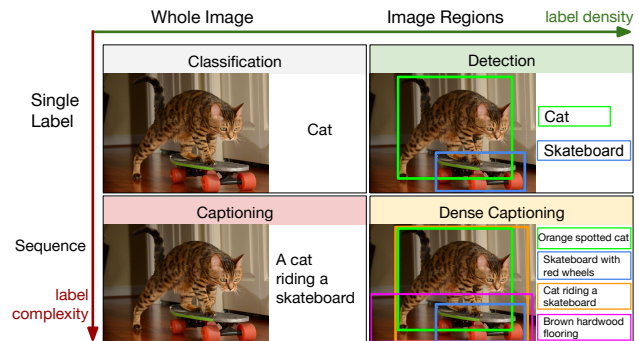

Figure 1. We address the Dense Captioning task (bottom right) with a model that jointly generates both dense and rich annotations in a single forward pass.

remained separate. In this work we take a step towards unifying these two inter-connected tasks into one joint framework. First, we introduce the dense captioning task (see Figure 1), which requires a model to predict a set of descriptions across regions of an image. Object detection is hence recovered as a special case when the target labels consist of one word, and image captioning is recovered when all images consist of one region that spans the full image.

Additionally, we develop a Fully Convolutional Localization Network (FCLN) for the dense captioning task. Our model is inspired by recent work in image captioning [49, 21, 32, 8, 4] in that it is composed of a Convolutional Neural Network and a Recurrent Neural Network language model. However, drawing on work in object detection [38], our second core contribution is to introduce a new dense localization layer. This layer is fully differentiable and can be inserted into any neural network that processes images to enable region-level training and predictions. Internally, the localization layer predicts a set of regions of interest in the image and then uses bilinear interpolation [19, 16] to smoothly crop the activations in each region.

We evaluate the model on the large-scale Visual Genome dataset, which contains 94,000 images and 4,100,000 region captions. Our results show both performance and speed improvements over approaches based on previous state of the art. We make our code and data publicly available to support further progress on the dense captioning task.

---

*Both authors contributed equally to this work.

2. Related Work

Our work draws on recent work in object detection, image captioning, and soft spatial attention that allows downstream processing of particular regions in the image.

Object Detection. Our core visual processing module is a Convolutional Neural Network (CNN) [29, 26], which has emerged as a powerful model for visual recognition tasks [39]. The first application of these models to dense prediction tasks was introduced in R-CNN [14], where each region of interest was processed independently. Further work has focused on processing all regions with only single forward pass of the CNN [17, 13], and on eliminating explicit region proposal methods by directly predicting the bounding boxes either in the image coordinate system [46, 9], or in a fully convolutional [31] and hence position-invariant settings [40, 38, 37]. Most related to our approach is the work of Ren *et al.* [38] who develop a region proposal network (RPN) that regresses from anchors to regions of interest. However, they adopt a 4-step optimization process, while our approach does not require training pipelines. Additionally, we replace their RoI pooling mechanism with a differentiable, spatial soft attention mechanism [19, 16]. In particular, this change allows us to backpropagate through the region proposal network and train the whole model jointly.

Image Captioning. Several pioneering approaches have explored the task of describing images with natural language [1, 27, 12, 34, 42, 43, 28, 20]. More recent approaches based on neural networks have adopted Recurrent Neural Networks (RNNs) [50, 18] as the core architectural element for generating captions. These models have previously been used in language modeling [2, 15, 33, 44], where they are known to learn powerful long-term interactions [22]. Several recent approaches to Image Captioning [32, 21, 49, 8, 4, 24, 11] rely on a combination of RNN language model conditioned on image information, possibly with soft attention mechanisms [51, 5]. Similar to our work, Karpathy and Fei-Fei [21] run an image captioning model on regions but they do not tackle the joint task of detection of description in one model. Our model is end-to-end and designed in such way that the prediction for each region is a function of the global image context, which we show also ultimately leads to stronger performance. Finally, the metrics we develop for the dense captioning task are inspired by metrics developed for image captioning [48, 7, 3].

3. Model

Overview. Our goal is to design an architecture that jointly localizes regions of interest and then describes each with natural language. The primary challenge is to develop a model that supports end-to-end training with a single step of optimization, and both efficient and effective inference. Our proposed architecture (see Figure 2) draws on architectural elements present in recent work on object detection, image captioning and soft spatial attention to simultaneously address these design constraints.

In Section 3.1 we first describe the components of our model. Then in Sections 3.2 and 3.3 we address the loss function and the details of training and inference.

3.1. Model Architecture

3.1.1 Convolutional Network

We use the VGG-16 architecture [41] for its state-of-the-art performance [39]. It consists of 13 layers of 3×3 convolutions interspersed with 5 layers of 2×2 max pooling. We remove the final pooling layer, so an input image of shape $3 \times W \times H$ gives rise to a tensor of features of shape $C \times W' \times H'$ where $C = 512$, $W' = \lfloor \frac{W}{16} \rfloor$, and $H' = \lfloor \frac{H}{16} \rfloor$. The output of this network encodes the appearance of the image at a set of uniformly sampled image locations, and forms the input to the localization layer.

3.1.2 Fully Convolutional Localization Layer

The localization layer receives an input tensor of activations, identifies spatial regions of interest and smoothly extracts a fixed-sized representation from each region. Our approach is based on that of Faster R-CNN [38], but we replace their RoI pooling mechanism [13] with bilinear interpolation [19], allowing our model to propagate gradients backward through the coordinates of predicted regions. This modification opens up the possibility of predicting affine or morphed region proposals instead of bounding boxes [19], but we leave these extensions to future work.

Inputs/outputs. The localization layer accepts a tensor of activations of size $C \times W' \times H'$. It then internally selects B regions of interest and returns three output tensors giving information about these regions:

1. **Region Coordinates**: A matrix of shape $B \times 4$ giving bounding box coordinates for each output region.

2. **Region Scores**: A vector of length B giving a confidence score for each output region. Regions with high confidence scores are more likely to correspond to ground-truth regions of interest.

3. **Region Features**: A tensor of shape $B \times C \times X \times Y$ giving features for output regions; is represented by an $X \times Y$ grid of C-dimensional features.

Convolutional Anchors. Similar to Faster R-CNN [38], our localization layer predicts region proposals by regressing offsets from a set of translation-invariant anchors. In particular, we project each point in the $W' \times H'$ grid of input features back into the $W \times H$ image plane, and consider k anchor boxes of different aspect ratios centered at this projected point. For each of these k anchor boxes, the localization layer predicts a confidence score and four
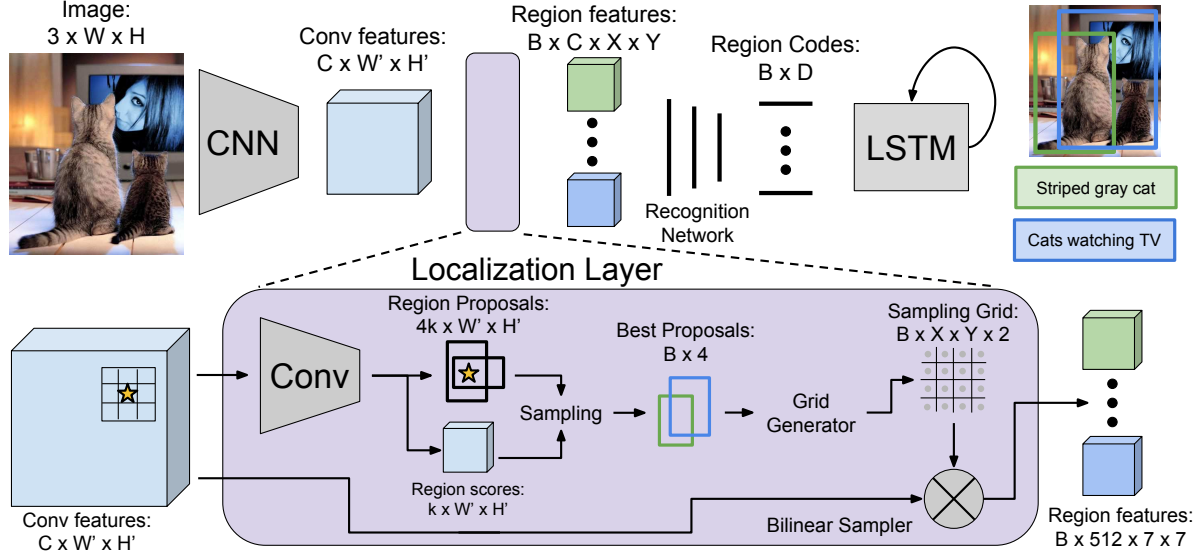
Figure 2. Model overview. An input image is first processed a CNN. The Localization Layer proposes regions and smoothly extracts a batch of corresponding activations using bilinear interpolation. These regions are processed with a fully-connected recognition network and described with an RNN language model. The model is trained end-to-end with gradient descent.

scalars regressing from the anchor to the predicted box coordinates. These are computed by passing the input feature map through a 3×3 convolution with 256 filters, a rectified linear nonlinearity, and a 1×1 convolution with $5k$ filters. This results in a tensor of shape $5k \times W' \times H'$ containing scores and offsets for all anchors.

Box Regression. We adopt the parameterization of [13] to regress from anchors to the region proposals. Given an anchor box with center (x_a, y_a), width w_a, and height h_a, our model predicts scalars (t_x, t_y, t_w, t_h) giving normalized offsets and log-space scaling transforms, so that the output region has center (x, y) and shape (w, h) given by

$$x = x_a + t_x w_a \qquad y = y_a + t_y h_a \qquad (1)$$
$$w = w_a \exp(t_w) \qquad h = h_a \exp(h_w) \qquad (2)$$

Note that the boxes are discouraged from drifting too far from their anchors due to L2 regularization.

Box Sampling. Processing a typical image of size $W = 720, H = 540$ with $k = 12$ anchor boxes gives rise to 17,280 region proposals. Since running the recognition network and the language model for all proposals would be prohibitively expensive, it is necessary to subsample them.

At training time, we follow the approach of [38] and sample a minibatch containing $B = 256$ boxes with at most $B/2$ positive regions and the rest negatives. A region is positive if it has an intersection over union (IoU) of at least 0.7 with some ground-truth region; in addition, the predicted region of maximal IoU with each ground-truth region is positive. A region is negative if it has IoU < 0.3 with all ground-truth regions. Our sampled minibatch contains $B_P \leq B/2$ positive regions and $B_N = B - B_P$ negative

regions, sampled uniformly without replacement from the set of all positive and all negative regions respectively.

At test time we subsample using greedy non-maximum suppression (NMS) based on the predicted proposal confidences to select the $B = 300$ most confident proposals.

The coordinates and confidences of the sampled proposals are collected into tensors of shape $B \times 4$ and B respectively, and are output from the localization layer.

Bilinear Interpolation. After sampling, we are left with region proposals of varying sizes and aspect ratios. In order to interface with the full-connected recognition network and the RNN language model, we must extract a fixed-size feature representation for each variably sized region proposal.

To solve this problem, Fast R-CNN [13] proposes an RoI pooling layer where each region proposal is projected onto the $W' \times H'$ grid of convolutional features and divided into a coarse $X \times Y$ grid aligned to pixel boundaries by rounding. Features are max-pooled within each grid cell, resulting in an $X \times Y$ grid of output features.

The RoI pooling layer is a function of two inputs: convolutional features and region proposal coordinates. Gradients can be propagated backward from the output features to the input features, but not to the input proposal coordinates. To overcome this limitation, we replace the RoI pooling layer with with bilinear interpolation [16, 19].

Concretely, given an input feature map U of shape $C \times W' \times H'$ and a region proposal, we interpolate the features of U to produce an output feature map V of shape $C \times X \times Y$. After projecting the region proposal onto U we follow [19] and compute a sampling grid G of shape $X \times Y \times 2$ associating each element of V with real-valued coordinates into U. If $G_{i,j} = (x_{i,j}, y_{i,j})$ then $V_{c,i,j}$ should be equal to U

at $(c, x_{i,j}, y_{i,j})$; however since $(x_{i,j}, y_{i,j})$ are real-valued, we convolve with a sampling kernel k and set

$$V_{c,i,j} = \sum_{i'=1}^{W} \sum_{j'=1}^{H} U_{c,i',j'} k(i' - x_{i,j}) k(j' - y_{i,j}). \quad (3)$$

We use bilinear sampling, corresponding to the kernel $k(d) = \max(0, 1 - |d|)$. The sampling grid is a linear function of the proposal coordinates, so gradients can be propagated backward into predicted region proposal coordinates. Running bilinear interpolation to extract features for all sampled regions gives a tensor of shape $B \times C \times X \times Y$, forming the final output from the localization layer.

3.1.3 Recognition Network

The recognition network is a fully-connected neural network that processes region features from the localization layer. The features from each region are flattened into a vector and passed through two full-connected layers, each using rectified linear units and regularized using Dropout. For each region this produces a code of dimension $D = 4096$ that compactly encodes its visual appearance. The codes for all positive regions are collected into a matrix of shape $B \times D$ and passed to the RNN language model.

In addition, we allow the recognition network one more chance to refine the confidence and position of each proposal region. It outputs a final scalar confidence of each proposed region and four scalars encoding a final spatial offset to be applied to the region proposal. These two outputs are computed as a linear transform from the D-dimensional code for each region. The final box regression uses the same parameterization as Section 3.1.2.

3.1.4 RNN Language Model

Following previous work [32, 21, 49, 8, 4], we use the region codes to condition an RNN language model [15, 33, 44]. Concretely, given a training sequence of tokens $s_1, \ldots, s_T$, we feed the RNN $T + 2$ word vectors $x_{-1}, x_0, x_1, \ldots, x_T$, where $x_{-1} = \text{CNN}(I)$ is the region code encoded with a linear layer and followed by a ReLU non-linearity, x_0 corresponds to a special START token, and x_t encode each of the tokens s_t, $t = 1, \ldots, T$. The RNN computes a sequence of hidden states h_t and output vectors y_t using a recurrence formula $h_t, y_t = f(h_{t-1}, x_t)$ (we use the LSTM [18] recurrence). The vectors y_t have size $|V|+1$ where V is the token vocabulary, and where the additional one is for a special END token. The loss function on the vectors y_t is the average cross entropy, where the targets at times $t = 0, \ldots, T - 1$ are the token indices for s_{t+1}, and the target at $t = T$ is the END token. The vector y_{-1} is ignored. Our tokens and hidden layers have size 512.

At test time we feed the visual information x_{-1} to the RNN. At each time step we sample the most likely next token and feed it to the RNN in the next time step, repeating the process until the special END token is sampled.

3.2. Loss function

During training our ground truth consists of positive boxes and descriptions. Our model predicts positions and confidences of sampled regions twice: in the localization layer and again in the recognition network. We use binary logistic losses for the confidences trained on sampled positive and negative regions. For box regression, we use a smooth L1 loss in transform coordinate space similar to [38]. The fifth term in our loss function is a cross-entropy term at every time-step of the language model.

We normalize all loss functions by the batch size and sequence length in the RNN. We searched over an effective setting of the weights between these contributions and found that a reasonable setting is to use a weight of 0.1 for the first four criterions, and a weight of 1.0 for captioning.

3.3. Training and optimization

We train the full model end-to-end in a single step of optimization. We initialize the CNN with weights pretrained on ImageNet [39] and all other weights from a gaussian with standard deviation of 0.01. We use stochastic gradient descent with momentum 0.9 to train the weights of the convolutional network, and Adam [23] to train the other components of the model. We use a learning rate of 1×10^{-6} and set $\beta_1 = 0.9, \beta_2 = 0.99$. We begin fine-tuning the layers of the CNN after 1 epoch, and for efficiency we do not fine-tune the first four convolutional layers of the network.

Our training batches consist of a single image that has been resized so that the longer side has 720 pixels. Our implementation uses Torch 7 [6] and [36]. One mini-batch runs in approximately 300ms on a Titan X GPU and it takes about three days of training for the model to converge.

4. Experiments

Dataset. Existing datasets that relate images and natural language either only include full image captions [3, 52], or ground words of image captions in regions but do not provide individual region captions [35]. We perform our experiments using the Visual Genome (VG) region captions dataset [25] [1]. Our version contained 94,313 images and 4,100,413 snippets of text (43.5 per image), each grounded to a region of an image. Images were taken from the intersection of MS COCO and YFCC100M [47], and annotations were collected on Amazon Mechanical Turk by asking workers to draw a bounding box on the image and describe its content in text. Example captions from the dataset include "cats play with toys hanging from a perch", "newspapers are scattered across a table", "woman pouring wine into a glass", "mane of a zebra", and "red light".

---

[1]Dataset can be downloaded at http://visualgenome.org/.

Figure 3. Example captions generated and localized by our model on test images. We render the top few most confident predictions. On the bottom row we additionally contrast the amount of information our model generates compared to the Full image RNN.

Preprocessing. We collapse words that appear less than 15 times into a special <UNK> token, giving a vocabulary of 10,497 words. We strip referring phrases such as "there is...", or "this seems to be a". For efficiency we discard all annotations with more than 10 words (7% of annotations). We also discard all images that have fewer than 20 or more than 50 annotations to reduce the variation in the number of regions per image. We are left with 87,398 images; we assign 5,000 each to val/test splits and the rest to train.

For test time evaluation we also preprocess the ground truth regions in the validation/test images by merging heavily overlapping boxes into single boxes with several reference captions. For each image we iteratively select the box with the highest number of overlapping boxes (based on IoU with threshold of 0.7), and merge these together (by taking the mean) into a single box with multiple reference captions. We then exclude this group and repeat the process.

4.1. Dense Captioning

In the dense captioning task the model receives a single image and produces a set of regions, each annotated with a confidence and a caption.

Evaluation metrics. Intuitively, we would like our model to produce both well-localized predictions (as in object detection) and accurate descriptions (as in image captioning). Inspired by evaluation metrics in object detection [10,

30] and image captioning [48], we propose to measure the mean Average Precision (AP) across a range of thresholds for both localization and language accuracy. For localization we use intersection over union (IoU) thresholds .3, .4, .5, .6, .7. For language we use METEOR score thresholds 0, .05, .1, .15, .2, .25. We adopt METEOR since this metric was found to be most highly correlated with human judgments in settings with a low number of references [48]. We measure the average precision across all pairwise settings of these thresholds and report the mean AP.

To isolate the accuracy of language in the predicted captions without localization we also merge ground truth captions across each test image into a bag of references sentences and evaluate predicted captions with respect to these references without taking into account their spatial position.

Baseline models. Following Karpathy and Fei-Fei [21], we train only the Image Captioning model (excluding the localization layer) on individual, resized regions. We refer to this approach as a *Region RNN model*. To investigate the differences between captioning trained on full images or regions we also train the same model on full images and captions from MS COCO (*Full Image RNN model*).

At test time we consider three sources of region proposals. First, to establish an upper bound we evaluate the model on ground truth boxes (GT). Second, similar to [21] we use

Region source	Language (METEOR)			Dense captioning (AP)			Test runtime (ms)			
	EB	RPN	GT	EB	RPN	GT	Proposals	CNN+Recog	RNN	Total
Full image RNN [21]	0.173	0.197	0.209	2.42	4.27	*14.11*	210ms	2950ms	**10ms**	3170ms
Region RNN [21]	0.221	0.244	0.272	1.07	4.26	*21.90*	210ms	2950ms	**10ms**	3170ms
FCLN on EB [13]	**0.264**	**0.296**	0.293	4.88	3.21	*26.84*	210ms	**140ms**	**10ms**	360ms
Our model (FCLN)	**0.264**	0.273	**0.305**	**5.24**	**5.39**	*27.03*	**90ms**	**140ms**	**10ms**	**240ms**

Table 1. Dense captioning evaluation on the test set of 5,000 images. The language metric is METEOR (high is good), our dense captioning metric is Average Precision (AP, high is good), and the test runtime performance for a 720×600 image with 300 proposals is given in milliseconds on a Titan X GPU (ms, low is good). EB, RPN, and GT correspond to EdgeBoxes [54], Region Proposal Network [38], and ground truth boxes respectively, used at test time. Numbers in GT columns (italic) serve as upper bounds assuming perfect localization.

an external region proposal method to extract 300 boxes for each test image. We use EdgeBoxes [54] (EB) due to their strong performance and speed. Finally, EdgeBoxes have been tuned to obtain high recall for objects, but our regions data contains a wide variety of annotations around groups of objects, stuff, etc. Therefore, as a third source of test time regions we follow Faster R-CNN [38] and train a separate Region Proposal Network (RPN) on the VG regions data. This corresponds to training our full model except without the RNN language model.

As the last baseline we reproduce the approach of Fast R-CNN [13], where the region proposals during training are fixed to EdgeBoxes instead of being predicted by the model (*FCLN on EB*). The results of this experiment can be found in Table 1. We now highlight the main takeaways.

Discrepancy between region and image level statistics. Focusing on the first two rows of Table 1, the Region RNN model obtains consistently stronger results on METEOR alone, supporting the difference in the language statistics present on the level of regions and images. Note that these models were trained on nearly the same images, but one on full image captions and the other on region captions. However, despite the differences in the language, the two models reach comparable performance on the final metric.

RPN outperforms external region proposals. In all cases we obtain performance improvements when using the RPN network instead of EB regions. The only exception is the FCLN model that was only trained on EB boxes. Our hypothesis is that this reflects people's tendency of annotating regions more general than those containing objects. The RPN network can learn these distributions from the raw data, while the EdgeBoxes method was designed for high recall on objects. In particular, note that this also allows our model (FCLN) to outperform the FCLN on EB baseline, which is constrained to EdgeBoxes during training (5.24 vs. 4.88 and 5.39 vs. 3.21). This is despite the fact that their localization-independent language scores are comparable, which suggests that our model achieves improvements specifically due to better localization. Finally, the noticeable drop in performance of the FCLN on EB model when evaluating on RPN boxes (5.39 down to 3.21) also suggests that the EB boxes have particular visual statistics,

and that the RPN boxes are likely out of sample for the FCLN on EB model.

Our model outperforms individual region description. Our final model performance is listed under the RPN column as 5.39 AP. In particular, note that in this one cell of Table 1 we report the performance of our full joint model instead of our model evaluated on the boxes from the independently trained RPN network. Our performance is quite a bit higher than that of the Region RNN model, even when the region model is evaluated on the RPN proposals (5.93 vs. 4.26). We attribute this improvement to the fact that our model can take advantage of visual information from the context outside of the test regions.

Qualitative results. We show example predictions of the dense captioning model in Figure 3. The model generates rich snippet descriptions of regions and accurately grounds the captions in the images. For instance, note that several parts of the elephant are correctly grounded and described ("trunk of an elephant", "elephant is standing", and both "leg of an elephant"). The same is true for the airplane example, where the tail, engine, nose and windows are correctly localized. Common failure cases include repeated detections (e.g. the elephant is described as standing twice).

Runtime evaluation. Our model is efficient at test time: a 720×600 image is processed in 240ms. This includes running the CNN, computing $B = 300$ region proposals, and sampling from the language model for each region.

Table 1 (right) compares the test-time runtime performance of our model with baselines that rely on EdgeBoxes. Regions RNN is slowest since it processes each region with an independent forward pass of the CNN; with a runtime of 3170ms it is more than $13\times$ slower than our method.

FCLN on EB extracts features for all regions after a single forward pass of the CNN. Its runtime is dominated by EdgeBoxes, and it is $\approx 1.5\times$ slower than our method.

Our method takes 88ms to compute region proposals, of which nearly 80ms is spent running NMS to subsample regions in the Localization Layer. This time can be drastically reduced by using fewer proposals: using 100 region proposals reduces our total runtime to 166ms.

	Ranking				Localization			
	R@1	R@5	R@10	Med. rank	IoU@0.1	IoU@0.3	IoU@0.5	Med. IoU
Full Image RNN [21]	0.10	0.30	0.43	13	-	-	-	-
EB + Full Image RNN [21]	0.11	0.40	0.55	9	0.348	0.156	0.053	0.020
Region RNN [21]	0.18	0.43	0.59	7	0.460	0.273	0.108	0.077
Our model (FCLN)	**0.27**	**0.53**	**0.67**	**5**	**0.560**	**0.345**	**0.153**	**0.137**

Table 2. Results for image retrieval experiments. We evaluate ranking using recall at k (R@K, higher is better) and median rank of the target image (Med.rank, lower is better). We evaluate localization using ground-truth region recall at different IoU thresholds (IoU@t, higher is better) and median IoU (Med. IoU, higher is better). Our method outperforms baselines at both ranking and localization.
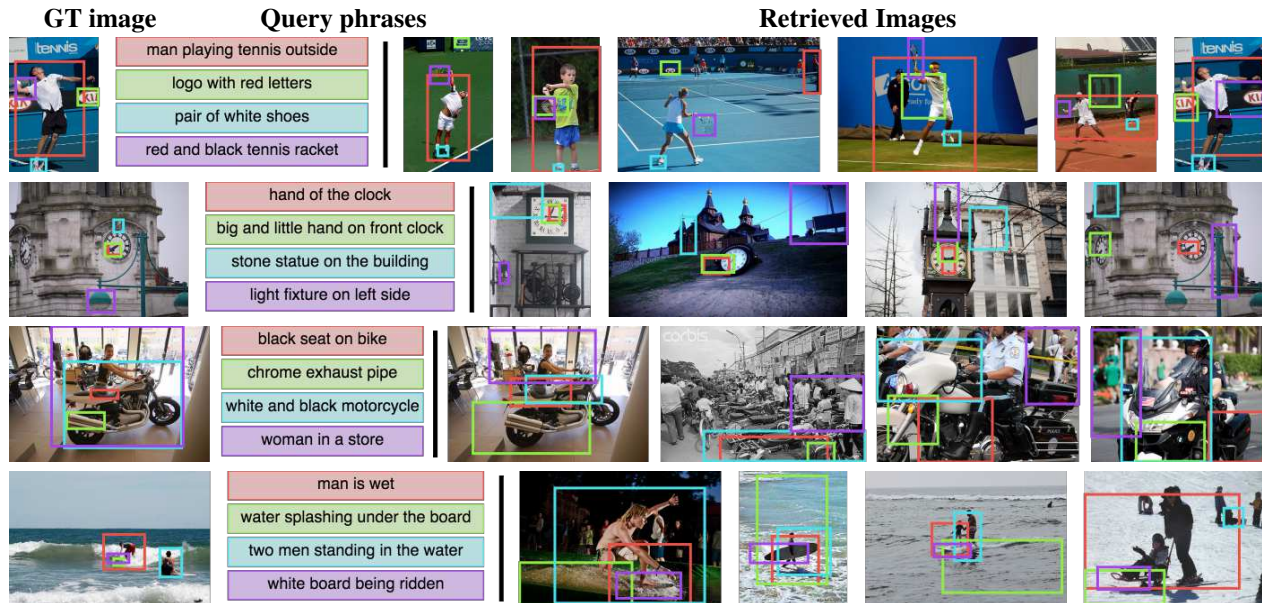

Figure 4. Example image retrieval results using our dense captioning model. From left to right, each row shows a ground-truth test image, ground-truth region captions describing the image, and the top images retrieved by our model using the text of the captions as a query. Our model is able to correctly retrieve and localize people, animals, and parts of both natural and man-made objects.

4.2. Image Retrieval using Regions and Captions

In addition to generating novel descriptions, our dense captioning model can support image retrieval using natural-language queries, and can localize these queries in retrieved images. We evaluate our model's ability to correctly retrieve images and accurately localize textual queries.

Experiment setup. We use 1000 random images from the VG test set for this experiment. We generate 100 test queries by repeatedly sampling four random captions from some image and then expect the model to correct retrieve the source image for each query.

Evaluation. To evaluate ranking, we report the fraction of queries for which the correct source image appears in the top k positions for $k \in \{1, 5, 10\}$ (recall at k) and the median rank of the correct image across all queries.

To evaluate localization, for each query caption we examine the image and ground-truth bounding box from which the caption was sampled. We compute IoU between this ground-truth box and the model's predicted grounding

for the caption. We then report the fraction of query caption for which this overlap is greater than a threshold t for $t \in \{0.1, 0.3, 0.5\}$ (recall at t) and the median IoU across all query captions.

Models. We compare the ranking and localization performance of full model with baseline models from Section 4.1.

For the Full Image RNN model trained on MS COCO, we compute the probability of generating each query caption from the entire image and rank test images by mean probability across query captions. Since this does not localize captions we only evaluate its ranking performance.

The Full Image RNN and Region RNN methods are trained on full MS COCO images and ground-truth VG regions respectively. In either case, for each query and test image we generate 100 region proposals using EdgeBoxes and for each query caption and region proposal we compute the probability of generating the query caption from the region. Query captions are aligned to the proposal of maximal probability, and images are ranked by the mean probability

head of a giraffe *legs of a zebra* *red and white sign* *white tennis shoes* *hands holding a phone* *front wheel of a bus*
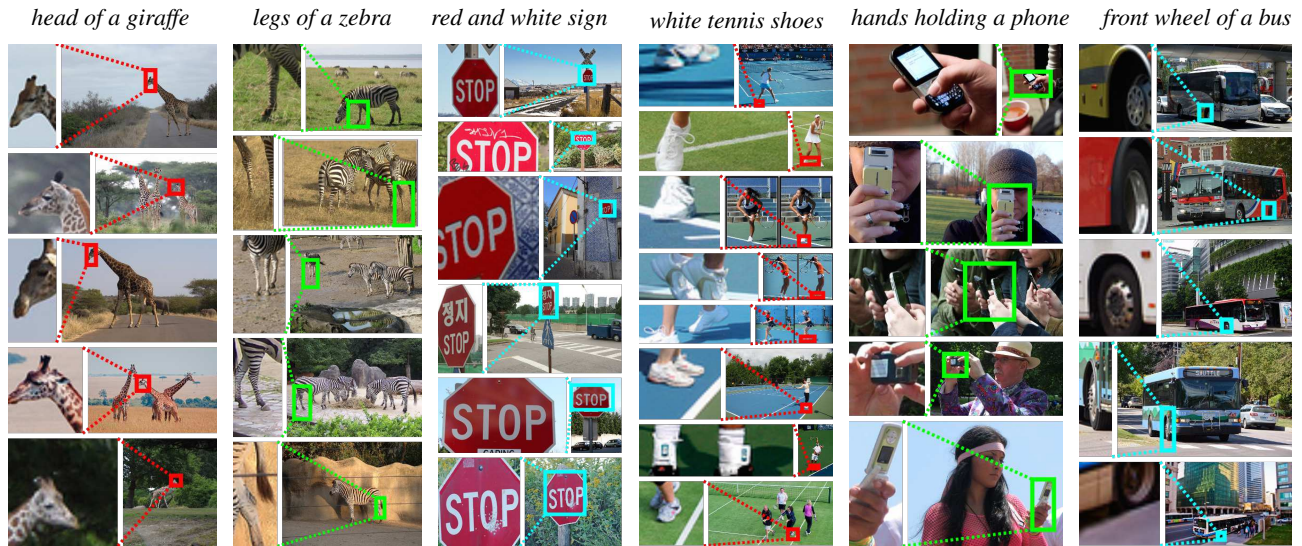
Figure 5. Example results for open world detection. We use our dense captioning model to localize arbitrary pieces of text in images, and display the top detections on the test set for several queries.

of aligned caption / region pairs.

The process for the full FCLN model is similar, but uses the top 100 proposals from the localization layer rather than EdgeBoxes proposals.

Discussion. Figure 4 shows examples of ground-truth images, query phrases describing those images, and images retrieved from these queries using our model. Our model is able to localize small objects ("hand of the clock", "logo with red letters"), object parts, ("black seat on bike", "chrome exhaust pipe"), people ("man is wet") and some actions ("man playing tennis outside").

Quantitative results comparing our model against the baseline methods is shown in Table 2. The relatively poor performance of the Full Image RNN model (Med. rank 13 vs. 9,7,5) may be due to mismatched statistics between its train and test distributions: the model was trained on full images, but in this experiment it must match region-level captions to whole images (Full Image RNN) or process image regions rather than full images (EB + Full Image RNN).

The Region RNN model does not suffer from a mismatch between train and test data, and outperforms the Full Image RNN model on both ranking and localization. Compared to Full Image RNN, it reduces the median rank from 9 to 7 and improves localization recall at 0.5 IoU from 0.053 to 0.108.

Our model outperforms the Region RNN baseline for both ranking and localization under all metrics, further reducing the median rank from 7 to 5 and increasing localization recall at 0.5 IoU from 0.108 to 0.153.

The baseline uses EdgeBoxes which was tuned to localize objects, but not all query phrases refer to objects. Our model achieves superior results since it learns to propose regions from the training data.

Open-world Object Detection Using the retrieval setup described above, our dense captioning model can also be used to localize arbitrary pieces of text in images. This enables "open-world" object detection, where instead of committing to a fixed set of object classes at training time we can specify object classes using natural language at test-time. We show example results for this task in Figure 5, where we display the top detections on the test set for several phrases.

Our model can detect animal parts ("head of a giraffe", "legs of a zebra") and also understands some object attributes ("red and white sign", "white tennis shoes") and interactions between objects ("hands holding a phone"). The phrase "front wheel of a bus" is a failure case: the model correctly identifies wheels of buses, but cannot distinguish between the front and back wheel.

5. Conclusion

We introduced the dense captioning task, which requires a model to simultaneously localize and describe regions of an image. To address this task we developed the FCLN architecture, which supports end-to-end training and efficient test-time performance. Our FCLN architecture is based on recent CNN-RNN models developed for image captioning but includes a novel, differentiable localization layer that can be inserted into any neural network to enable spatially-localized predictions. Our experiments in both generation and retrieval settings demonstrate the power and efficiency of our model with respect to baselines related tp previous work, and qualitative experiments show visually pleasing results. In future work we would like to relax the assumption of rectangular proposal regions and to discard test-time NMS in favor of a trainable spatial suppression layer.

6. Acknowledgments

References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *JMLR*, 2003. 2

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003. 2

[3] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollar, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 1, 2, 4

[4] X. Chen and C. L. Zitnick. Mind's eye: A recurrent visual representation for image caption generation. *CVPR*, 2015. 1, 2, 4

[5] K. Cho, A. C. Courville, and Y. Bengio. Describing multimedia content using attention-based encoder-decoder networks. *CoRR*, abs/1507.01053, 2015. 2

[6] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011. 4

[7] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014. 2

[8] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv preprint arXiv:1411.4389*, 2014. 1, 2, 4

[9] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. *CVPR*, 2014. 2

[10] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5

[11] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. Platt, et al. From captions to visual concepts and back. *CVPR*, 2015. 2

[12] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. *ECCV*, 2010. 2

[13] R. Girshick. Fast R-CNN. *ICCV*, 2015. 2, 3, 6

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 2014. 1, 2

[15] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013. 2, 4

[16] K. Gregor, I. Danihelka, A. Graves, and D. Wierstra. DRAW: A recurrent neural network for image generation. *ICML*, 2015. 1, 2, 3

[17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2015*, 2015. 2

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2, 4

[19] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. *NIPS*, 2015. 1, 2, 3

[20] Y. Jia, M. Salzmann, and T. Darrell. Learning cross-modality similarity for multinomial data. *ICCV*, 2011. 2

[21] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *CVPR*, 2015. 1, 2, 4, 5, 6, 7

[22] A. Karpathy, J. Johnson, and L. Fei-Fei. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015. 2

[23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 4

[24] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *TACL*, 2015. 2

[25] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. 4

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2

[27] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating simple image descriptions. *CVPR*, 2011. 2

[28] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Generalizing image captions for image-text parallel corpus. In *ACL (2)*, pages 790–796. Citeseer, 2013. 2

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. *ECCV*, 2014. 5

[31] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, 2015. 2

[32] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*, 2014. 1, 2, 4

[33] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010. 2, 4

[34] V. Ordonez, X. Han, P. Kuznetsova, G. Kulkarni, M. Mitchell, K. Yamaguchi, K. Stratos, A. Goyal, J. Dodge, A. Mensch, et al. Large scale retrieval and generation of image descriptions. *International Journal of Computer Vision (IJCV)*, 2015. 2

[35] 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. *ICCV*, 2015. 4

[36] qassemoquab. stnbhwd. https://github.com/qassemoquab/stnbhwd, 2015. 4

[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640*, 2015. 2

[38] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *NIPS*, 2015. 1, 2, 3, 4, 6

[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015. 1, 2, 4

[40] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*, 2014. 1, 2

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 2

[42] R. Socher and L. Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using un-aligned text corpora. *CVPR*, 2010. 2

[43] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *TACL*, 2014. 2

[44] I. Sutskever, J. Martens, and G. E. Hinton. Generating text with recurrent neural networks. *ICML*, 2011. 2, 4

[45] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2015. 1

[46] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov. Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*, 2014. 1, 2

[47] B. Thomee, B. Elizalde, D. A. Shamma, K. Ni, G. Friedland, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 4

[48] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. *CVPR*, 2015. 2, 5

[49] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *CVPR*, 2015. 1, 2, 4

[50] P. J. Werbos. Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, 1(4):339–356, 1988. 2

[51] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *ICML*, 2015. 1, 2

[52] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014. 4

[53] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *ECCV*, 2014. 1

[54] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. *ECCV*, 2014. 6