Fast Dense Captioning with SSD

Mihir Chauhan
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
mihirhem@buffalo.edu

Abuzar Shaikh
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
mshaikh2@buffalo.edu

Abstract

We present a faster approach to solve dense captioning task, which helps to delimit and detail important regions in images in natural language form. Our approach uses Single Shot Multi-Box Detection (SSD) in conjunction with Long Short Term Memory (LSTM) creates sequence of captions. To solve for localization and description task jointly we propose SSD architecture that eliminates region proposal generation and encapsulates all computation in a single network. Experimental results on the Visual Genome dataset, which comprises of 108,700 images and 5.4 Million region descriptions. We notice significant improvements in system's speed and accuracy using current state of the art approach for object detection and localization.

1 Introduction

> *"The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves."*
>
> *- Jeff Hawkins, Founder Palm Computing.*

Following Jeff Hawkins's, one of the dreams of the field of Artificial Intelligence is to enable computers to see and understand the rich visual world around us and endow them with the ability to communicate with us in natural language. Humans find it easy to comprehend a visual understanding in an image in natural language. This ability is so natural and effortless for us but is a big challenge for a computer to understand. A computer visualizes an image as one large array of numbers indicating brightness at any position. An image consists of millions of these pixels and a computer must find patterns of brightness to create semantic relationship between patterns of objects in an image. The challenge here however is to associate all possible patterns related to an object. The challenges on the language side are no less severe because it involves complex process of identifying important features of an image and mapping it to sequence of words in a vocabulary. The ability to associate images with natural language sentences that describe what is depicted in them is a hallmark of image understanding, and a prerequisite for applications such as sentence-based image search. In this paper, we have developed a technique for using natural language as a label space for computer vision tasks. We have unified object detection and image captioning into one joint framework. Additionally, we use SSD to address the dense captioning task. Our model is inspired by recent work in object detection [1] in that the elimination of region proposal helps speed up our system. The fundamental improvement in speed comes from eliminating bounding box proposals and the feature resampling stages. We evaluate our model on the large-scale Visual Genome dataset, which contains 108,700 images and 5.4 Million region captions. Our result shows both performance and speed improvements over approaches based on previous state of the art. This technique is a step towards a future in which we can interact with computers in real time.

2 Related Work

The work in this paper draws inspiration from recent work in object detection and language processing models. An image classification problem is predicting the label of an image among the predefined labels. It assumes that there is single object of interest in the image and it covers a significant portion of image. Detection is about not only finding the class of object but also localizing the extent of an object in the image.

Object Detection. Traditional method of detection involved using a block wise orientation histogram (SIFT or HOG) feature which could not achieve high accuracy in standard datasets such a PASCAL VOC. These methods encode a very low level characteristics of the objects and therefore are not able to distinguish well among the different labels. Deep learning based methods have become the state of the art in object detection in image. Starting from Girshick et. al. paper, a flurry of papers have been published which have either focused on improving run-time efficiency or accuracy. These have used more or less the same pipeline involving CNN (convolutional neural network) as feature extractor. One brute force method is to run classification on all the sub-windows formed by sliding different sized patches all through the image. This will be a tedious process from computational time point of view as each sub-window would require passing it through CNN and calculating the feature for that region. R-CNN therefore uses an object proposal algorithm like selective search in its pipeline which gives out a number (~2000) of tentative object locations and extents on the basis of local cues like color RGB, HSV etc. SPP (spatial pyramid pooling) is one of the turning points for making a highly accurate RCNN pipeline feasible in run-time. RCNN had to pass all the ~2000 regions from SS independently through CNN and is therefore a very slow algorithm. SPP allows the whole image (not individual regions but the whole image) to be passed through the convolutional layer only once. This saves a lot of time because same patch may belong to multiple regions and convolutions on them are not calculated multiple time as done in RCNN thereby enabling a shared computation of conv. layers among the regions. Since major chunk of time (~90%) is spent on the convolutional layers it reduces the computation time drastically. Still SPP has certain downsides that it does not use the full potential of CNN because training is not end-to-end. Fast RCNN tackles the downsides by installing the net with the capacity to back-propagate the gradients from FC layer to conv. layers. Multitask objective is a salient feature of Fast-RCNN as it no longer requires training of the network independently for classification and localization. The change reduces the overall training time and increases the accuracy in comparison to SPP net because of end to end learning of CNN. In the pipeline of Fast-RCNN, the slowest part is generating regions from SS (~2s) or edge-boxes (~0.2s). Faster-RCNN replaces SS with CNN itself for generating the region proposals (called RPN-region proposal network) which gives out tentative regions at almost negligible amount of time. Our work is most related to SSD for object detection modality. The fundamental betterment done in SSD comes from eliminating bounding box proposals and the feature resampling stage from Faster-RCNN. This results in significant improvement in speed (59 FPS with mAP 74.3% on VOC2007 test, vs. Faster R-CNN 7 FPS with mAP 73.2% or YOLO 45 FPS with mAP 63.4%)

Image Captioning. Several pioneering approaches have explored the task of describing images with natural language [1, 27, 13, 34, 41, 42, 28, 21]. More recent approaches based on neural networks have adopted Recurrent Neural Networks (RNNs) [49, 19] as the core architectural element for generating captions. These models have previously been used in language modeling [2, 16, 33, 43], where they are known to learn powerful long-term interactions [23]. The RNN is trained in the context of single “end-to-end” network. Several recent approaches to Image Captioning [32, 22, 48, 9, 5, 25, 12] rely on a combination of RNN language model conditioned on image information. A recent related approach is the work of Xu et al. [50] who use a soft attention mechanism [6] over regions of the input image with every generated word. Our approach to spatial attention is more general in that the network can process arbitrary affine regions in the image instead of only discrete grid positions in an intermediate conv. volume. However, for simplicity, during generation we follow Vinyals et al. [48], where the visual information is only passed to the language model once on the first time step.

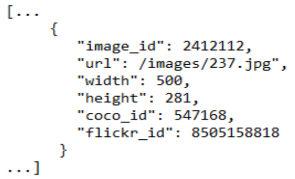
Finally, the metrics we develop for the dense captioning task are inspired by metrics developed for image captioning.

3 Dataset Definition

Datasets that provide connection between the objects in the image and natural language description gives only the base words associated to the image or the entire statements associated with an image. Similar to Karpathy et al we perform our model evaluations on Visual Genome (VG) dataset. This dataset contains 108,077 images and 5.4 Million snippets of text (50 per image), each grounded to a region of an image, 35 Objects per image, 33877 object categories and 21 relationships per image. Images were taken from combination of MS COCO and YFCC100M [46], 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. A typical example of the object caption looks like "a cat sitting on the table" where the base words are connected by the prepositional and verb connectors like "on" and "sitting" For our experiments we use the images, meta data and the region descriptions from the dataset.
Images: The combination of images extracted from MS COCO and Flickr8.
Meta data: Information about the image.

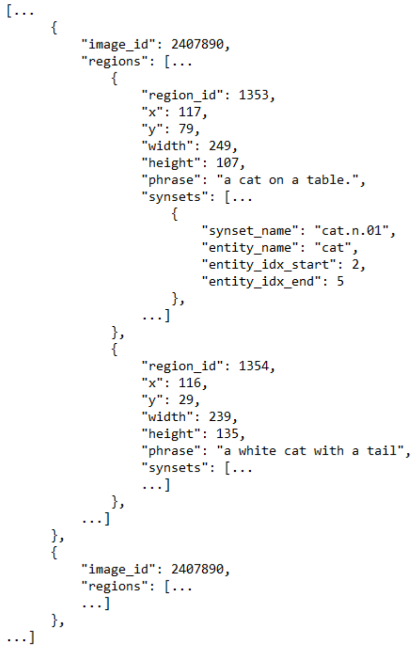
Table 1: Object Position in Image

JSON	Name	Type	Description

 [...
{
"image_id": 2412112,
"url": /images/237.jpg",
"width": 500,
"height": 281,
"coco_id": 547168,
"flickr_id": 8505158818
}
...]
	Image_id	Int	ID of image

	URL	Hyperlink string	Location of the dataset
	Width	Int	Width of image in px
	Height	Int	Height of image in px
	Coco_id	Int	Image ID in COCO dataset
	Flickr_id	Int	Image ID in Flickr dataset

Table 2: Region Descriptions in Image

JSON	Name	Type	Description
A Sample Region Description JSON Object :  [... { "image_id": 2407890, "regions": [... { "region_id": 1353, "x": 117, "y": 79, "width": 249, "height": 107, "phrase": "a cat on a table.", "synsets": [... { "synset_name": "cat.n.01", "entity_name": "cat", "entity_idx_start": 2, "entity_idx_end": 5 }, ...] }, { "region_id": 1354, "x": 116, "y": 29, "width": 239, "height": 135, "phrase": "a white cat with a tail", "synsets": [... ...] }, ...] }, { "image_id": 2407890, "regions": [... ...] }, ...]	Image_id	Int	ID of image
	Regions	Array	Region Array descriptions
	• .region_id	Int	Region ID descriptions
	• .x	Int	x-coordinate of region box
	• .y	Int	y-coordinate of region box
	• .width	Int	Width of region box
	• .height	Int	Height of region box
	• .phrase	Str	Region-description-phrase
	• .synsets	Array	Synsets in description
	○ .synset_name	Str	Synset name
	○ .entity_name	Str	String from phrase
	○ .enitity_idx_ start	Int	Start index of synset in phrase
	○ .enitity_idx_ end	Int	End index of synset in the phrase

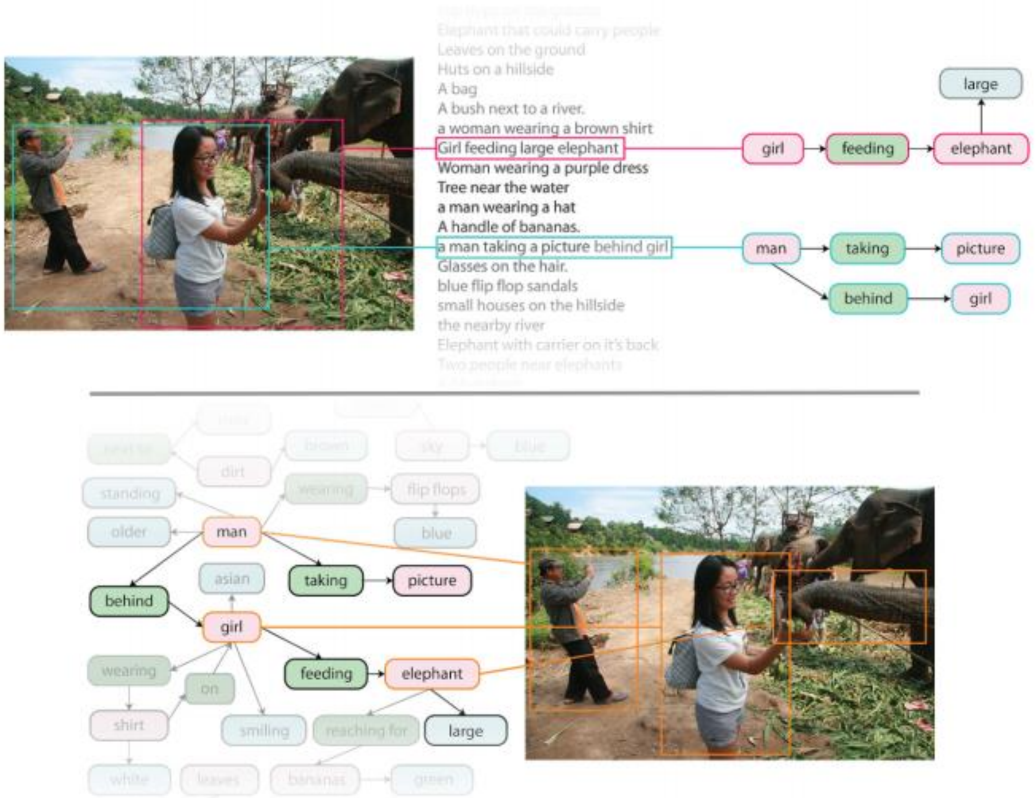


Figure 1. An overview of the data needed to move from perceptual awareness to cognitive understanding of images.

4 Pre-Processing

Preprocessing of the region description JSON data file containing regions and captions are converted into a single large JSON file containing a list which describes a single image containing: id of region, image id, height, width, phrase, x-coordinate & y-coordinate. All the images on disk will be preprocessed into an HDF5 file and a JSON file with some auxiliary information. The captions will be tokenized with some basic preprocessing (split by words, remove special characters).

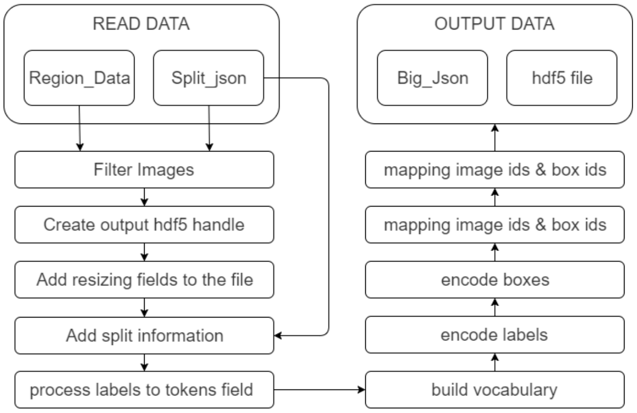

READ DATA
Region_Data
Split_json
Filter Images
Create output hdf5 handle
Add resizing fields to the file
Add split information
process labels to tokens field
OUTPUT DATA
Big_Json
hdf5 file
mapping image ids & box ids
mapping image ids & box ids
encode boxes
encode labels
build vocabulary

Figure 2. Flowchart for Pre-Processing

5 Model

Overview: Our goal towards building this model was to increase the accuracy and speed of detecting objects and relating them with natural language. The first challenge was to identify how the existing state of the art architecture works and disentangling the factors of variations, lastly merging the single shot detection with recurrent neural network, and training the networks jointly while still reducing the losses. The architecture proposed in below figure (Fig 1) delineate the elements in our model that take care of the object detection and the language model.

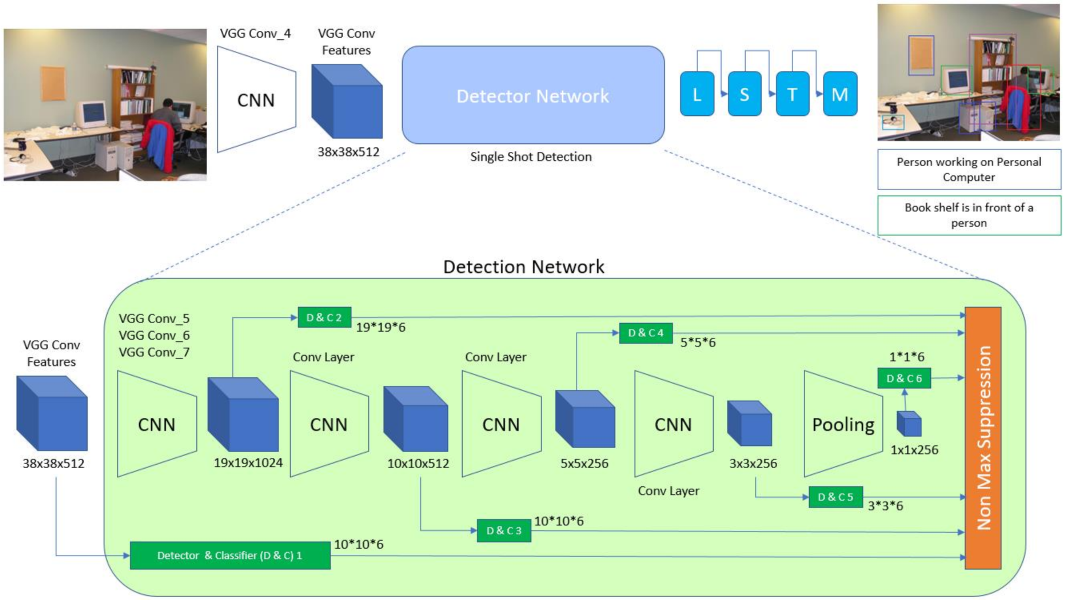


Figure 3. Architecture of Fast Dense Captioning

We will discuss these foundations in Section 5.1 and then describe the loss function in the subsequent section.

5.1. Model Architecture

5.1.1 Feature Extraction Network

We use the VGG-16 architecture [40] for its state-of-the-art performance [38]. It consists of 13 layers of 3 × 3 convolutions intermixed with 5 layers of 2 × 2 max pooling. We remove the final pooling layer, so an input image of shape 3 × W × H gives rise to a tensor of features of shape 512xW/16xH/16. This final layer is the feature maps representation of the original image and is an input to one of the detector and classifier that contribute to predicting regions.

5.1.2 Single Shot Detection Network

We have replaced the Region Proposal Network that generated 300 region proposals in [55] with the auxiliary convolution layers that are added after the base VGG-16 model. The CNN layers go on decreasing in dimension as we progress ahead in the network, catering to object-detection of varying scale in the original image. Each conv. layer emits object proposal offsets and the classification score of each proposed object. Finally, the features of 300 proposals with the highest confidence score are considered as an input to the next RNN layer.

Each Detector & Classifier (D & C) layer predicts 6 anchor boxes like anchor boxes predicted by RPN in Faster RCNN [37]. As the width and height of the feature maps decrease the number of grids decrease proportionally, consequently reducing number of proposals as displayed in the Detection Network in figure. With the reduction in the number of grids the size of each cell in the grid increases, resulting in a wider anchor box (see figure 3)

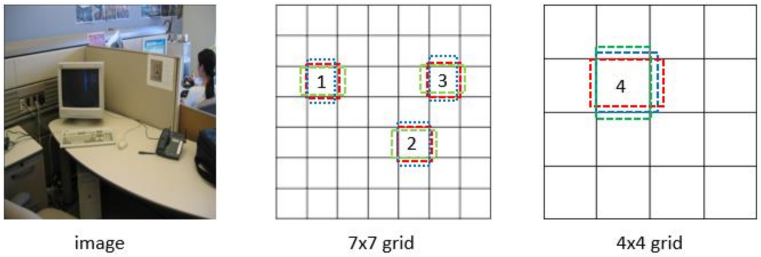


Figure 4. Varying feature maps of image

As shown in figure 4. A grid of 7x7 detects a phone with highest confidence while the grid of 4x4 detects computer monitor with highest confidence. Lastly, we get a total of around 8700 detected regions from all the auxiliary convolutions combined, to which we apply a Non-Max Suppression (NMS) [54] to get top 300 most confident proposals, as it would be very expensive to train the language model with ~8700 samples.
Next, we warp these regions to a fixed size of 40x40 to input in the language model, to overcome the challenge of varying sizes and shapes.

5.1.3 Long Short Term Memory (LSTM)

We train the language model using a Stateful LSTM model, where our first time-step is the encoded image region vector (I), in the next time-steps (*t)* we pass the START (t_0) token followed by the word vector (t_1 to t_n) related to the region followed by the END token, where t_x is the words in the description associated with the region. Once we complete one forward RNN pass for a sample region we reset and forget the states.
During prediction time, we input the region information (I) and predict the next token (t_x), we feed the RNN with t_t to predict the subsequent time-step t_{t+1} until an END token is encountered.

5.1. Loss Function

We train our model on visual regions in each image and their respective descriptions. The model predicts the offsets and the class scores of each region, for this box regression we use smooth L1 loss. The overall objective loss function of the detection network is the weighted sum of localization loss (L) and confidence loss (C).

$$L\_total(x,c,l,g,q) = \frac{1}{N}\Big(C(x,c) + \alpha L(x,l,g) + \beta(\frac{1}{M}\sum_{i=1}^{M}\log_2 q(w[i]))\Big)$$

We regress for offsets of center c_x and c_y of the anchor box to get the width (w) and height (h) of each box and update the L.

The classification loss is the simple SoftMax loss over multiple confidence scores. The third term in our loss function is the cross-entropy classification loss of predicting the next token in each time step, where M is the size of sequence and q(w) is the probability of word estimated from the training set.

We normalize the entire loss function by the batch size (N) and the RNN by M. Post experiments we set weights $\alpha = 0.1$ and $\beta = 1.0$ in the loss function.

6 Results of Experimentation

The task to be performed is annotating set of regions in an image with a confidence score and a caption.

Evaluation metrics. We have used mean Average Precision (mAP) across range of thresholds for localization and language accuracy. Specifically, we use Intersection over union (IoU) threshold for localization and METEOR score for language model. METEOR scores are most correlated with human judgement and hence serves as a good evaluation scheme.

Baseline model. Image captioning model trained on individual, resized regions is considered as a baseline model for our application. We additionally train on full images using captions provided by MS COCO dataset.

At run-time there are four comparison models:
1. Evaluation on Ground Truth (GT) boxes

2. Evaluation with external Region Proposal Method - EdgeBoxes [EB] with Fast R-CNN
3. Region proposal network as in Faster-RCNN
4. Elimination of proposal mechanism like Single Shot Detection

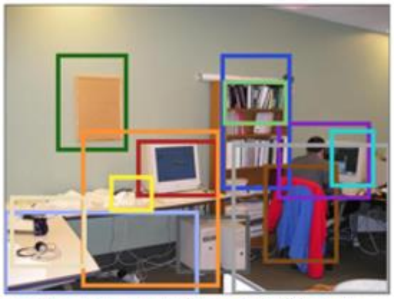
Table 3. Dense captioning evaluation on the test set of 5,000 images.

Region Source	Dense Captioning(AP)			Test runtime(ms)			
	EB	RPN	GT	Proposals	CNN+Recog	RNN	Total
Base Model	2.42	4.27	14.11	210ms	2950ms	10ms	3170ms
Region RNN	1.07	4.26	21.90	210ms	2950ms	10ms	3170ms
Fast RCNN	4.88	3.21	26.84	210ms	140ms	10ms	360ms
Faster RCNN	5.24	5.39	27.03	90ms	140ms	10ms	240ms
SSD	**5.62**		**29.23**	**10ms**	**15ms**	**10ms**	**35ms**

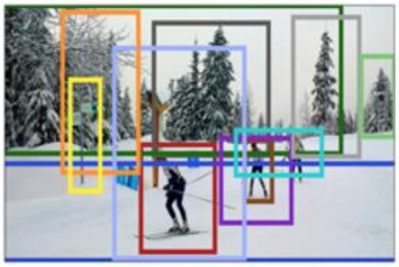
In table 3. the language metric is METEOR (high is good), our dense captioning metric is Average Precision (AP, high is good), and the test runtime performance for a 720 × 600 image with 300 proposals is given in milliseconds on a Titan X GPU (ms, low is good). EB, RPN, and GT correspond to EdgeBoxes [53], Region Proposal Network [37], and ground truth boxes respectively, used at test time. Numbers in GT columns (italic) serve as upper bounds assuming perfect localization.

Our model outperforms individual region description. Our final model performance is 5.62 AP. In particular, note that during the test run-time our model is almost 7 times faster than Faster-RCNN (using RPN). Our performance is quite a bit higher than that of the Faster R-CNN model, even though the region model evaluated is without region proposals. We attribute this improvement to the fact that our model can take advantage of visual information from multiple layers of prediction using multiple scales.

Qualitative results. We show example predictions of the dense captioning model in Figure 5. 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", "head of an elephant", and an "ear of an elephant"). The same is true for the computer example, where the white, computer monitor, laptop and books are correctly localized. Common failure cases include repeated detections (e.g. a computer monitor is described twice).



a white computer monitor. a shelf with books. a white box on the floor. a brown box on the wall. a red towel hanging on a rack. a man on a laptop. a blue chair with a red pillow. a book shelf with books. a white computer monitor. a computer monitor. a computer monitor. a bottle of water on the table. a small white computer mouse.



a person skiing. snow on the ground. a tree covered in snow. trees covered in snow. a person wearing a black jacket. a person skiing. a tree with snow. a tree with snow. a man skiing on a mountain. two people skiing. a tree with snow on it. a sign on the pole.

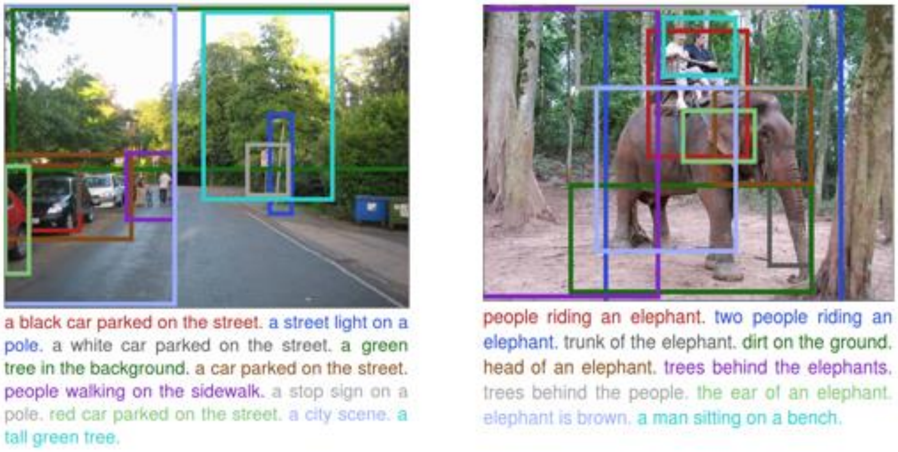

a black car parked on the street.
a street light on a pole.
a white car parked on the street.
a green tree in the background.
a car parked on the street.
people walking on the sidewalk.
a stop sign on a pole.
red car parked on the street.
a city scene.
a tall green tree.
people riding an elephant.
two people riding an elephant.
trunk of the elephant.
dirt on the ground.
head of an elephant.
trees behind the elephants.
trees behind the people.
the ear of an elephant.
elephant is brown.
a man sitting on a bench.

Figure.5 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.

7 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 use SSD architecture, which supports end-to-end training and efficient test-time performance. Our architecture is based on recent SSD models developed for image captioning. Our experiments in both generation and retrieval settings demonstrate the power and efficiency of our model with respect to baselines related to previous work, and qualitative experiments show visually pleasing results. In future work we would like to test our model on real time image frames in videos.

Acknowledgments

We are extremely grateful to Professor Sargur Srihari for elucidating all the necessary concepts related to Deep Learning. We also thank Dr. Ranganathan Setlur for providing computing resources on Nvidia Titan X GPU .We would also appreciate the efforts taken by the students of our class for proofreading the paper and furnishing critical reviews for the same.

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

[3] C. M. C. J. C. C. J. H. Bryan A. Plummer, Liwei Wang and S. Lazebni. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. 2015.

[4] 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.

[5] X. Chen and C. L. Zitnick. Learning a recurrent visual representation for image caption generation. arXiv preprint arXiv:1411.5654, 2014.

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

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

[8] 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.

[9] 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.

[10] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 2155–2162. IEEE, 2014.

[11] 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.

[12] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. Platt, et al. ´ From captions to visual concepts and back. arXiv preprint arXiv:1411.4952, 2014. 2 [13] 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. In ECCV. 2010.

[13] 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. In ECCV. 2010.

[14] R. Girshick. Fast r-cnn. arXiv preprint arXiv:1504.08083, 2015.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[16] A. Graves. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013.

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

[18] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. 2015.

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

[20] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. arXiv preprint arXiv:1506.02025, 2015.

[21] Y. Jia, M. Salzmann, and T. Darrell. Learning cross-modality similarity for multinomial data. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 2407–2414. IEEE, 2011.

[22] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. arXiv preprint arXiv:1412.2306, 2014.

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

[24] D. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[25] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539, 2014.

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

[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. In CVPR, 2011.

[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.

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

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Com- ´ mon objects in context. In Computer Vision–ECCV 2014, pages 740–755. Springer, 2014.

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

[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.

[33] T. Mikolov, M. Karafiat, L. Burget, J. Cernock ´ y, and S. Khu- ` danpur. Recurrent neural network based language model. In INTERSPEECH, 2010.

[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.

[35] qassemoquab. stnbhwd. https://github.com/ qassemoquab/stnbhwd, 2015. 4 [36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2015.

[37] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015.

[38] 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.

[39] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In ICLR, 2014.

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

[41] R. Socher and L. Fei-Fei. Connecting modalities: Semisupervised segmentation and annotation of images using unaligned text corpora. In CVPR, 2010.

[42] 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.

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

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. arXiv preprint arXiv:1409.4842, 2014.

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

[46] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. The new data and new challenges in multimedia research. arXiv preprint arXiv:1503.01817, 2015.

[47] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. arXiv preprint arXiv:1411.5726, 2014.

[48] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. arXiv preprint arXiv:1411.4555, 2014.

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

[50] 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. arXiv preprint arXiv:1502.03044, 2015.

[51] 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.

[52] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. arXiv preprint arXiv:1311.2901, 2013.

[53] C. L. Zitnick and P. Dollar. Edge boxes: Locating object ´ proposals from edges. In ECCV, 2014.

[54] Liu, Wei and Anguelov, Dragomir and Erhan, Dumitru and Szegedy, Christian and Reed, Scott and Fu, Cheng-Yang and Berg, Alexander C. SSD: MultiBox Detector ECCV, 2016.

[55] Johnson, Justin and Karpathy, Andrej and Fei-Fei, Li, DenseCap: Fully Convolutional Localization Networks for Dense Captioning Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016