Final Project Report

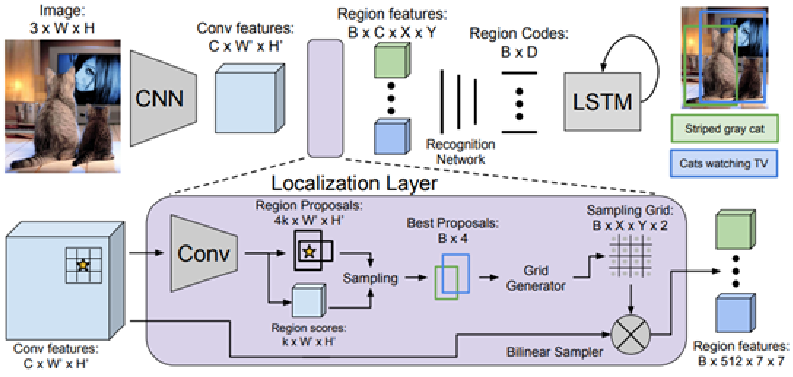
Project Overview

In this project, I developed a computer vision system to both localize and describe salient regions in images in natural language. This task generalizes object detection when the descriptions consist of a single word, and image captioning when one predicted region covers the full image. Specifically, I implemented the Fully Convolutional Localization Networks for Dense Captioning from the paper DenseCap. This model 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.

Approach

In the previous related works, the most popular and widely used approach is to split it into two separate tasks. However, one of the contributions for DenseCap is that it designs a unifying framework for this task.

Specifically, it has three major components, the first one is a convolutional network that converts the original image into convolution features, and the second part is a novel dense localization layer. It is inspired by the architecture in Faster RCNN, and what it does is that after another few convolutional layers, it use a region proposal layer to propose several regions from the images, and it would also generate a score for the regions. After sampling the best proposals, I use bilinear sampler to extract a fixed-size feature representation for each variably sized region proposal and generate the final region features. The third step is to use a recurrent network to generate the label sentences for each of the proposed regions.



Implementation Detail

Convolutional Network:
For this part, I used VGG16 pre-trained model to extract convolutional features from the input images.

Fully Convolutional Localization Layer:
I first feed the convolutional features into another two convolutional layers.
The next step is, for each element in the feature map, select k anchor boxes of different aspect ratios in the input image space.
For each of these, the localization layer will predict the offsets and confidence.
In the above process, I also used box sampling to subsample from all proposals to reduce computation complexity. The approach I took is that: I consider a region to be positive if it has an intersection over union (IoU) of at least 0.7 with some ground-truth region and negative if it has IoU less than 0.3 with all ground-truth regions.
The last step is to extract a fixed-size feature representation for each variably sized region proposal and generate the final region features. There are two approaches I consider, the RoI pooling layer and the bilinear interpolation sampling method. For RoI pooling layer, each region proposal is projected onto the grid of convolutional features and divided into a coarse grid aligned to pixel boundaries by rounding. Features are max-pooled within each grid cell, resulting in a grid of output features. For bilinear sampling, I interpolate the features to produce an output feature map, and after projecting the region proposal I compute a sampling grid associating each element with real-valued coordinates. Both approaches allows us to extract features for all sampled regions gives a tensor, forming the final output from the localization layer.

Recognition Network:
The feature maps I obtained earlier are passed through an MLP to compute representations corresponding to each region, and the results are used as the first state to an LSTM which is trained to predict each word of the caption. Specifically, I use LSTM architecture, and at each time step I 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.

Main Result

For training, due to the time constraint, I only complete 1000 epochs for RPN and 500 for RNN, which is much less than what the original paper did. The evaluation of the model consists of two parts, the localization accuracy and the caption generation accuracy, and thus I measure the mean Average Precision (AP) for both localization and language. For localization I use intersection over union (IoU) thresholds 0.3, 0.4, 0.5, 0.6, 0.7. For language I use METEOR score thresholds 0, 0.05, 0.1, 0.15, 0.2, 0.25. I measure the average precision across all pairwise settings of these thresholds and report the mean AP. I compared my results with the results reported in the original paper as a benchmark.

	Dense captioning (AP)	Language (METEOR)
Benchmark	5.39	0.305
RoI Pooling	2.18	0.217
Bilinear Interpolation	2.23	0.252

We can see that for localization part, my AP score is much lower than the original paper, possibly due to that RPN by nature requires much more training which I could not complete. It is also likely that for the localization network, I made some different design choice that is not so optimal, which results in the discrepancy in the result. If we compare the two sampling methods, the AP scores are quite close, and it is likely due to that the choice of sampling method would not affect the predicted region but only the representation of these regions. For the language part, I directly evaluate using the ground truth regions to eliminate the influence of localization, and the difference between METEOR score for two models are not so huge, which could suggest that the RNN part is doing fine. And the bilinear model does a better job on this task, which confirms the statement in the original paper.

Next Step

For the next step, I would experiment with another task, Image Retrieval using Regions and Captions, which means I would give the model an input of natural language query, and it would retrieve images and localize these queries in the images. I could generate several test queries by repeatedly sampling some random captions from some image and then expect the model to correct retrieve the source image for each query.
I would also try to continue to fine-tune the existing model. As mentioned earlier, I was only able to finish 1000 epochs for RPN and 500 epochs for RNN, and it is very likely that with more training, the performance can be improved.