Large Scale Object Identification and Image Captioning using GPUs and Apache Spark

Submitted in partial fulfillment of the requirements
for the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Aseem Patni (12CS10008)

under the guidance of

Dr. Sourangshu Bhattacharya

Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur
Kharagpur 721 302
May, 2016

Certificate

This is to certify that the report titled Large Scale Object Identification and Image Captioning using GPUs and Apache Spark submitted by Aseem Patni (12CS10008) to the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur is a bonafide record of work carried out by him under my supervision and guidance. The report has fulfilled all the requirements as per the regulations of the Institute and, in my opinion, has reached the standard needed for submission.

Dr. Sourangshu Bhattacharya
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Contents

List of Figures

Chapter 1

Introduction

As we move towards more complete image understanding, having more precise and detailed object recognition becomes crucial. In this context, one cares not only about classifying images, but also about precisely estimating the class and location of objects contained within the images, a problem known as object detection.

The main advances in object detection were achieved thanks to improvements in object representations and machine learning models. A prominent example of a state-of-the-art detection system is the Deformable Part-based Model (DPM) (2). It builds on carefully designed representations and kinematically inspired part decompositions of objects, expressed as a graphical model. Using discriminative learning of graphical models allows for building high-precision part-based models for variety of object classes.

Manually engineered representations in conjunction with shallow discriminatively trained models have been among the best performing paradigms for the related problem of object classification as well (3). In the last years, however, Deep Neural Networks (DNNs) (4)have emerged as a powerful machine learning model.

DNNs exhibit major differences from traditional approaches for classification. First, they are deep architectures which have the capacity to learn more complex models than shallow ones [2]. This expressivity and robust training algorithms allow for learning powerful object representations without the need to hand design features. This has been empirically demonstrated on the challenging ImageNet classification task (5) across thousands of classes (6; 24)

In this work, we exploit the power of Deep Comvolution Neural Networks leveraging Apache Spark and GPUs for the problem of object detection, where we not only classify but also try to precisely localize objects. The problem we are address here is challenging, since we want to detect a potentially large number object instances with varying sizes in the same image using a limited amount of computing resources.

Another problem that we tried to tackle is Deep Visual-Semantic Alignments for Generating Image Descriptions. The task is to generates natural language descriptions of images and their regions. The approach we use leverages datasets of images and their sentence descriptions to learn about the inter-modal correspondences between language and visual data. This was originally proposed by Andrej Karpathy from Stanford University. The alignment model is based on a novel combination of Convolutional Neural Networks over image regions, bidirectional Recurrent Neural Networks over sentences, and a structured objective that aligns the two modalities through a multimodal embedding.

Then we also experimented with dense captioning, which requires a computer vision system to both localize and describe salient regions in images in natural language. The dense caption- ing 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 local- ization and description task jointly we used a Fully Con- volutional Localization Network (FCLN) architecture that processes an image with a single, efficient forward pass, re- quires 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 also did experiments on different hardware to study the effect of computational resource on training time, and thus the quality of model trained. The no of parameters that can be trained directly depends on the computational resource and memory available. And, the performance is directly correlated with the number of parameters learned.

Chapter 2

Related Work

2.1 Dense image annotations

Barnard et al. (19) and Socher et. al. (20) studied the multimodal correspondence between words and images to annotate segments of images. Several works [(21), (23), (24), (31)] studied the problem of holistic scene understanding in which the scene type, objects and their spatial support in the image is inferred. However, the focus of these works is on correctly labeling scenes, objects and regions with a fixed set of categories, while our focus is on richer and higher-level descriptions of regions and the implementation on GPUs and Apache Spark.

2.2 Neural networks in visual and language domains

Multiple approaches have been developed for representing im- ages and words in higher-level representations. On the im- age side, Convolutional Neural Networks (CNNs) [(27), (28)] have recently emerged as a powerful class of models for image classification and object detection Recurrent Neural Networks have been previously used in language modeling. [(25), (26)]

Chapter 3

Data Set

Object detection task similar in style to PASCAL VOC Challenge. There are 200 basic-level categories for this task which are fully annotated on the test data, i.e. bounding boxes for all categories in the image have been labeled. The categories were carefully chosen considering different factors such as object scale, level of image clutterness, average number of object instance, and several others. Some of the test images will contain none of the 200 categories.

3.1 PASCAL

The training data provided consists of a set of images; each image has an annotation file giving a bounding box and object class label for each object in one of the twenty classes present in the image. multiple objects from multiple classes may be present in the same image.

A subset of images are also annotated with pixel-wise segmentation of each object present, to support the segmentation competition.

20 classes. The train/val data has 11,530 images containing 27,450 ROI annotated objects and 6,929 segmentations.

3.2 IMAGENET

ImageNet is an image dataset organized according to the WordNet hierarchy. Each meaningful concept in WordNet, possibly described by multiple words or word phrases, is called a ”synonym set” or ”synset”. There are more than 100,000 synsets in WordNet, majority of them are nouns (80,000+). ImageNet provides on average

1000 images to illustrate each synset. Images of each concept are quality-controlled and human-annotated. In its completion, it is expected that ImageNet will offer tens of millions of cleanly sorted images for most of the concepts in the WordNet hierarchy.

3.2.1 Statistice

- Total number of non-empty synsets: 21841
- Total number of images: 14,197,122
- Number of images with bounding box annotations: 1,034,908
- Number of synsets with SIFT features: 1000
- Number of images with SIFT features: 1.2 million

Comparative scale			
		PASCAL VOC 2012	ILSVRC 2014
Number of object classes		20	200
Training	Num images	5717	456567
	Num objects	13609	478807
Validation	Num images	5823	20121
	Num objects	13841	55502
Testing	Num images	10991	40152
	Num objects	—	—

Chapter 4

Hardware Specifications

The following systems have been used for all the computational purposes.

4.1 Apache Spark Cluster

The Spark Cluster consists of 17 nodes with the following specifications.

Processor	Intel Corporation Xeon E7 v2/Xeon E5 v2/Core i7
Clock Speed	2.30 GHz
Cores	12
Cache	15MB (L3 Cache)
RAM	128GB

4.2 GPU Server

The GPU server has the following specifications:

GPU	NVIDIA Corporation GK110GL [Tesla K20m]
Processor	Intel Corporation Xeon E5/Core i7
Clock Speed	2.30 GHz
Cores	12
Cache	15MB (L3 Cache)
RAM	64GB

4.3 Laptop

We couldn't get the required permissions to install important tools on the server, so we had to rely on my laptop also. The laptop has the following specifications:

Model	Samsung NP550P5C-S02IN
GPU	NVIDIA GeForce GT 650M
Processor	Intel Core i7-3610QM
Clock Speed	2.30 GHz
Cache	6MB (L3 Cache)
RAM	8GB

Chapter 5

Standard Competitions

5.1 PASCAL Visual Object Classes Challenge

The main goal of this challenge is to recognize objects from a number of visual object classes in realistic scenes (i.e. not pre-segmented objects). It is fundamentally a supervised learning learning problem in that a training set of labelled images is provided. The twenty object classes that have been selected are:

- Person: person
- Animal: bird, cat, cow, dog, horse, sheep
- Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train
- Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor

There are three main object recognition competitions: classification, detection, and segmentation, a competition on action classification, and a competition on large scale recognition run by ImageNet. In addition there is a ”taster” competition on person layout.

5.1.1 Classification Competitions

For each of the twenty classes, predicting presence/absence of an example of that class in the test image.

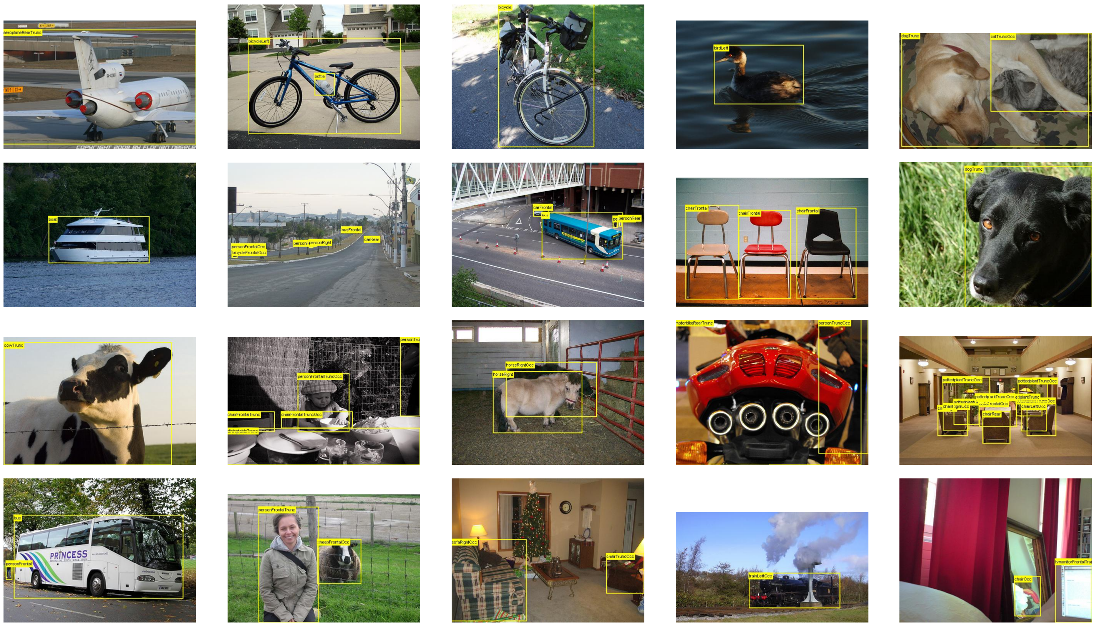


Figure 5.1: 20 classes

5.1.2 Detection Competitions

Predicting the bounding box and label of each object from the twenty target classes in the test image.

5.1.3 Segmentation Competitions

Generating pixel-wise segmentations giving the class of the object visible at each pixel, or ”background” otherwise.

5.1.4 Action Classification Competition

Predicting the action(s) being performed by a person in a still image.

5.2 ImageNet Large Scale Visual Recognition Challenge

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and image classification at large scale. One high level motivation is to allow researchers to compare progress in detection across a wider variety of objects – taking advantage of the quite expensive labeling effort. Another motivation is to measure the progress of computer vision for large scale image indexing for retrieval and annotation.

The ImageNet Large Scale Visual Recognition Challenge is a benchmark in object category classi- fication and detection on hundreds of object categories and millions of images. The challenge has been run annually from 2010 to present, attracting participation from more than fifty institutions

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has been running annually for five years (since 2010) and has become the standard benchmark for large-scale object recognition.1 ILSVRC follows in the footsteps of the PASCAL VOC challenge (Everingham et al., 2012), established in 2005, which set the precedent for standardized evaluation of recognition algorithms in the form of yearly competitions. As in PASCAL VOC, ILSVRC consists of two components:

1. a publically available dataset, and

2. an annual competition and corresponding workshop.

The dataset allows for the development and comparison of categorical object recognition algorithms, and the competition and workshop provide a way to track the progress and discuss the lessons learned from the most successful and innovative entries each year.

The publically released dataset contains a set of manually annotated training images. A set of test images is also released, with the manual annotations withheld.2 Participants train their algorithms using the training images and then automatically annotate the test images. These predicted annotations are submitted to the evaluation server. Results of the evaluation are revealed at the end of the competition period and authors are invited to share insights at the workshop held at the International Conference on Computer Vision (ICCV) or European Conference on Computer Vision (ECCV) in alternate years.

5.2.1 Workshop

Every year of the challenge there is a corresponding workshop at one of the premier computer vision conferences. The purpose of the workshop is to present the methods and results of the challenge. Challenge participants with the most successful and innovative entries are invited to present.

Chapter 6

Methodologies

6.1 SIFT and LBT + Stochastic SVM

Local Binary Patterns and Scale Invariant Feature Transform. ILSVRC2010. The first year the challenge consisted of just the classification task. The winning entry from NEC team (Lin et al., 2011) used SIFT (Lowe, 2004) and LBP (Ahonen et al., 2006) features with two nonlinear coding representations (Zhou et al., 2010; Wang et al., 2010) and a stochastic SVM. The honorable mention XRCE team (Perronnin et al., 2010) used an improved Fisher vector representation (Perronnin and Dance, 2007) along with PCA dimensionality reduction and data compression followed by a linear SVM. Fisher vectorbased methods have evolved over five years of the challenge and continued performing strongly in every ILSVRC from 2010 to 2014.

6.2 Histogram intersection kernel SVM

ILSVRC2011. The winning classification entry in 2011 was the 2010 runner-up team XRCE, applying highdimensional image signatures (Perronnin et al., 2010) with compression using product quantization (Sanchez and Perronnin, 2011) and one-vs-all linear SVMs. The single-object localization competition was held for the first time, with two brave entries. The winner was the UvA team using a selective search approach to generate class-independent object hypothesis regions (van de Sande et al., 2011b), followed by dense sampling and vector quantization of several color SIFT features (van de Sande et al., 2010), pooling with spatial pyramid matching (Lazebnik et al., 2006), and classifying with a histogram intersection kernel SVM

(Maji and Malik, 2009) trained on a GPU (van de Sande et al., 2011a).

6.3 Large-scale deep neural network

ILSVRC2012. This was a turning point for large-scale object recognition, when large-scale deep neural networks entered the scene. The undisputed winner of both the classification and localization tasks in 2012 was the SuperVision team. They trained a large, deep convolutional neural network on RGB values, with 60 million parameters using an efficient GPU implementation and a novel hidden-unit dropout trick (Krizhevsky et al., 2012; Hinton et al., 2012). The second place in image classification went to the ISI team, which used Fisher vectors (Sanchez and Perronnin, 2011) and a streamlined version of Graphical Gaussian Vectors (Harada and Kuniyoshi, 2012), along with linear classifiers using Passive-Aggressive (PA) algorithm (Crammer et al., 2006). The second place in single-object localization went to the VGG, with an image classification system including dense SIFT features and color statistics (Lowe, 2004), a Fisher vector representation (Sanchez and Perronnin, 2011), and a linear SVM classifier, plus additional insights from (Arandjelovic and Zisserman, 2012; Sanchez et al., 2012). Both ISI and VGG used (Felzenszwalb et al., 2010) for object localization; SuperVision used a regression model trained to predict bounding box locations. Despite the weaker detection model, SuperVision handily won the object localization task. A detailed analysis and comparison of the SuperVision and VGG submissions on the single-object localization task can be found in (Russakovsky et al., 2013) The influence of the success of the SuperVision model can be clearly seen in ILSVRC2013 and ILSVRC2014.

6.4 Deep CNN averaged together

ILSVRC2013. There were 24 teams participating in the ILSVRC2013 competition, compared to 21 in the previous three years combined. Following the success of the deep learning-based method in 2012, the vast majority of entries in 2013 used deep convolutional neural networks in their submission. The winner of the classification task was Clarifai, with several large deep convolutional networks averaged together. The network architectures were chosen using the visualization technique of (Zeiler and Fergus, 2013), and they were trained on the GPU following (Zeiler et al., 2011) using the dropout technique (Krizhevsky et al., 2012).

ILSVRC2014. 2014 attracted the most submissions, with 36 teams submitting

123 entries compared to just 24 teams in 2013 a 1.5x increase in participation.9 As in 2013 almost all teams used convolutional neural networks as the basis for their submission. Significant progress has been made in just one year: image classification error was almost halved since ILSVRC2013 and object detection mean average precision almost doubled compared to ILSVRC2013.

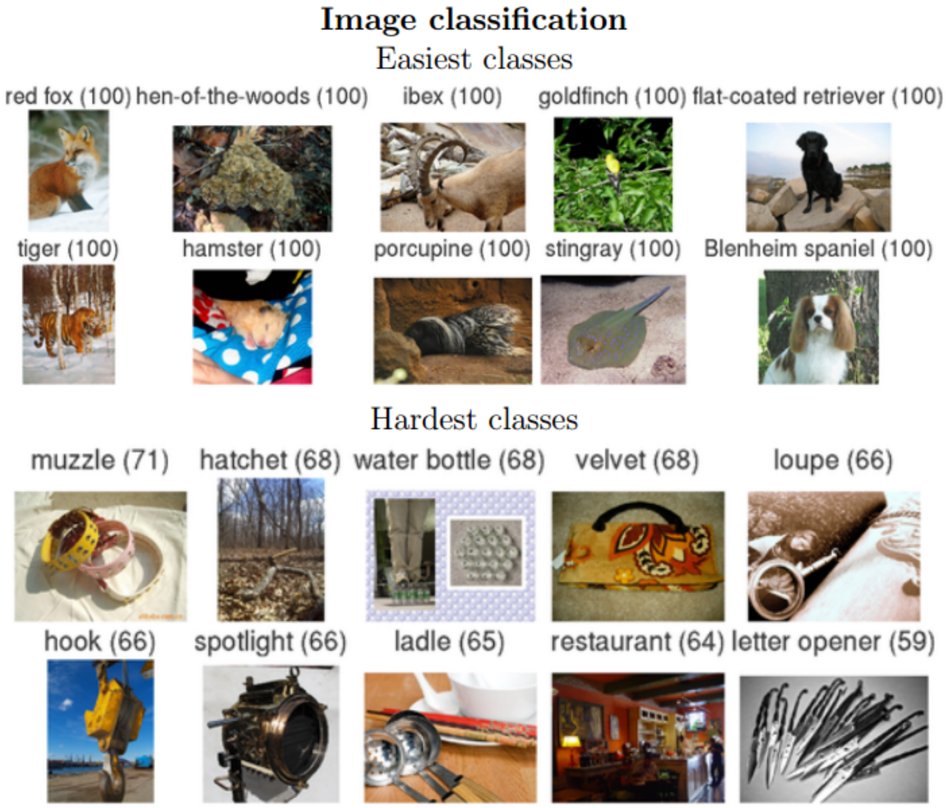
The winning image classification with provided data team was GoogLeNet, which explored an improved convolutional neural network architecture combining the multi-scale idea with intuitions gained from the Hebbian principle. Additional dimension reduction layers allowed them to increase both the depth and the width

6.5 CNN + RNN + FCNN

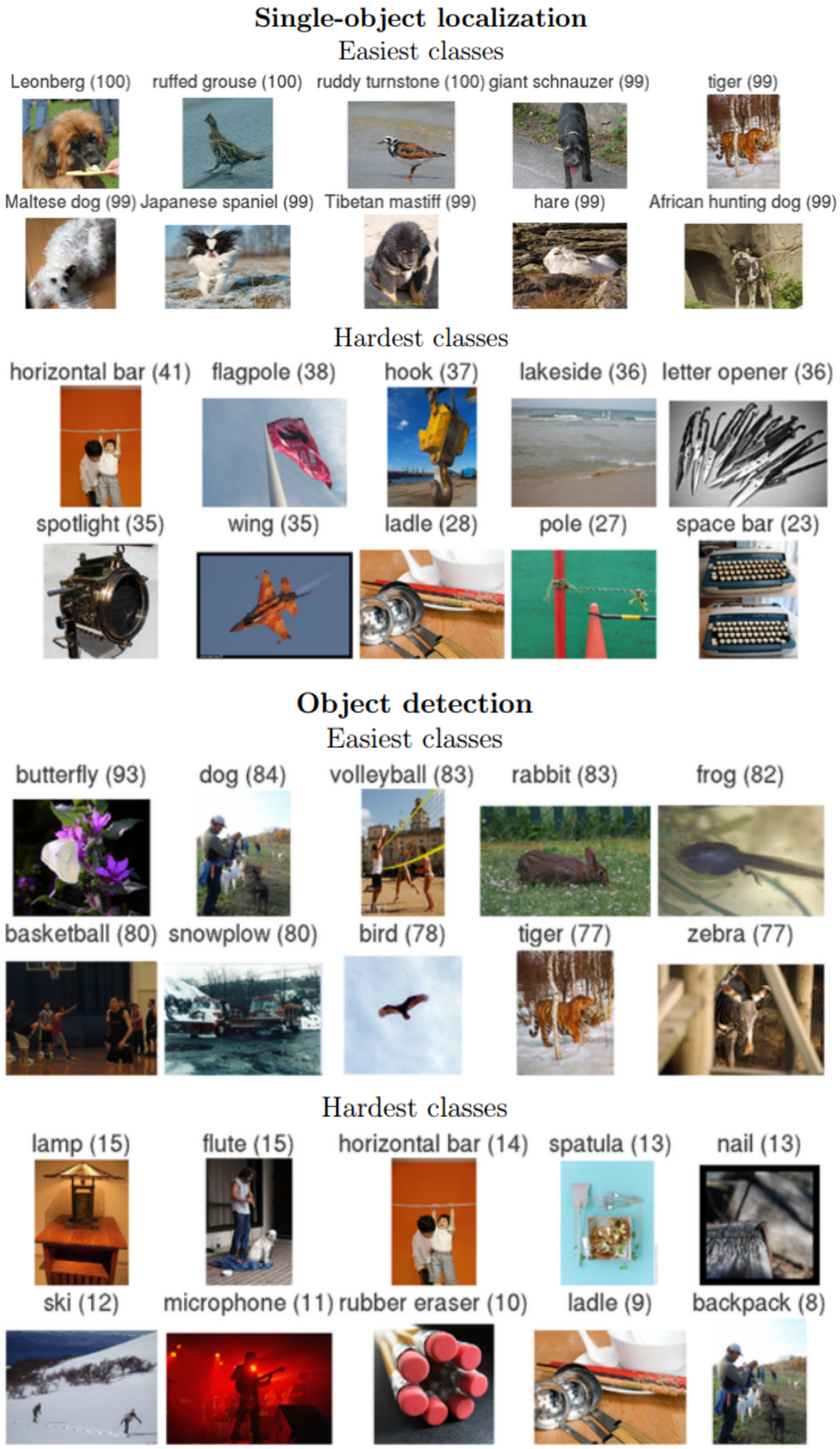
The goal is to design an architecture that jointly localizes regions of interest and then describes each with natural language. The primary challenge is to develop a model that supports end-to-end training with a single step of optimization, and both efficient and effective inference. We used the architecture proposed by Andrej Kerpathy et. al. (14) which draws on architectural elements present in recent work on object detection, image captioning and soft spatial attention to simultaneously address these design constraints. More details about the approach and architecture are provided in later sections

Chapter 7

Statistical Significance



For each object category, we take the best performance of any entry submitted to ILSVRC2012-2014 (including entries using additional training data). Given these optimistic results we show the easiest and harder classes for each task. The numbers in parentheses indicate classification and localization accuracy. For image classification the 10 easiest classes are randomly selected from among 121 object classes with 100% accuracy.


Single-object localization
Easiest classes
Leonberg (100)
ruffed grouse (100)
ruddy turnstone (100)
giant schnauzer (99)
tiger (99)
Maltese dog (99)
Japanese spaniel (99)
Tibetan mastiff (99)
hare (99)
African hunting dog (99)
Hardest classes
horizontal bar (41)
flagpole (38)
hook (37)
lakeside (36)
letter opener (36)
spotlight (35)
wing (35)
ladle (28)
pole (27)
space bar (23)
Object detection
Easiest classes
butterfly (93)
dog (84)
volleyball (83)
rabbit (83)
frog (82)
basketball (80)
snowplow (80)
bird (78)
tiger (77)
zebra (77)
Hardest classes
lamp (15)
flute (15)
horizontal bar (14)
spatula (13)
nail (13)
ski (12)
microphone (11)
rubber eraser (10)
ladle (9)
backpack (8)

Chapter 8

Approach for this work

8.1 Classification with Deep Convolutional Neural Networks

8.1.1 Architecture

The architecture of our network is summarized in Figure. It contains eight learned layers five convolutional and three fully-connected.(11)

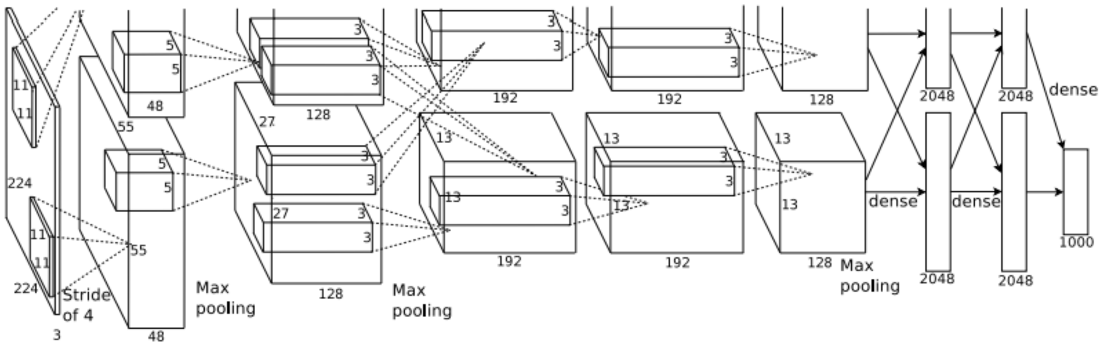


Figure 8.1: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The networks input is 150,528-dimensional, and the number of neurons in the networks remaining layers is given by 253,440186,62464,89664,89643,264 409640961000.

8.1.2 ReLU Nonlinearity

The standard way to model a neurons output f as a function of its input x is with

$$f(x) = tanh(x) \qquad \text{or} \qquad f(x) = (1 + e^x)^1$$

. In terms of training time with gradient descent, these saturating nonlinearities are much slower than the non-saturating nonlinearity

$$f(x) = max(0, x)$$

.

8.1.3 Training on Multiple GPUs

A single GTX 580 GPU has only 3GB of memory, which limits the maximum size of the networks that can be trained on it. It turns out that 1.2 million training examples are enough to train networks which are too big to fit on one GPU

The parallelization scheme that we employ essentially puts half of the kernels (or neurons) on each GPU, with one additional trick: the GPUs communicate only in certain layers.

8.1.4 Overlapping Pooling

Pooling layers in CNNs summarize the outputs of neighboring groups of neurons in the same kernel map. Traditionally, the neighborhoods summarized by adjacent pooling units do not overlap (e.g., [17, 11, 4]). To be more precise, a pooling layer can be thought of as consisting of a grid of pooling units spaced s pixels apart, each summarizing a neighborhood of size z z centered at the location of the pooling unit. If we set s = z, we obtain traditional local pooling as commonly employed in CNNs. If we set s ¡ z, we obtain overlapping pooling. This is what we use throughout our network, with s = 2 and z = 3. This scheme reduces the top-1 and top-5 error rates by 0.4% and 0.3%, respectively, as compared with the non-overlapping scheme s = 2, z = 2, which produces output of equivalent dimensions. We generally observe during training that models with overlapping pooling find it slightly more difficult to overfit.

8.1.5 Overall Architecture

Now we are ready to describe the overall architecture of our CNN. As depicted in Figure, the net contains eight layers with weights; the first five are convolutional and the remaining three are fullyconnected. The output of the last fully-connected layer is fed to a 1000-way softmax which produces a distribution over the 1000 class labels. The network maximizes the multinomial logistic regression objective, which is equivalent to maximizing the average across training cases of the log-probability of the correct label under the prediction distribution.

8.2 GoogleNet

8.2.1 Architecture Details

The main idea of the Inception architecture is to consider how an optimal local sparse structure of a convolutional vision network can be approximated and covered by readily available dense components. Note that assuming translation invariance means that our network will be built from convolutional building blocks. All we need is to find the optimal local construction and to repeat it spatially

In order to avoid patch-alignment issues, current incarnations of the Inception architecture are restricted to filter sizes 11, 33 and 55; this decision was based more on convenience rather than necessity. It also means that the suggested architecture is a combination of all those layers with their output filter banks concatenated into a single output vector forming the input of the next stage.

Additionally, since pooling operations have been essential for the success of current convolutional networks, it suggests that adding an alternative parallel pooling path in each such stage should have additional beneficial effect, too

One big problem with the above modules, at least in this nave form, is that even a modest number of 55 convolutions can be prohibitively expensive on top of a convolutional layer with a large number of filters. This problem becomes even more pronounced once pooling units are added to the mix: the number of output filters equals to the number of filters in the previous stage. The merging of output of the pooling layer with outputs of the convolutional layers would lead to an inevitable increase in the number of outputs from stage to stage. While this architecture might cover the optimal sparse structure, it would do it very inef- ficiently, leading to a computational blow up within a few stages.

In general, an Inception network is a network consisting of modules of the above type stacked upon each other, with occasional max-pooling layers with stride 2 to halve the resolution of the grid. For technical reasons (memory efficiency during training), it seemed beneficial to start using Inception modules only at higher layers while keeping the lower layers in traditional convolutional fashion. This is not strictly necessary, simply reflecting some infrastructural inefficiencies in our current implementation.

8.3 Captioning using CNN + RNN + FCLL

The aim 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. We used the architecture proposed by Andrej Kerpathy et. al. (14) which draws on architectural elements present in recent work on object detection, image captioning and soft spatial attention to simultaneously address these design constraints.

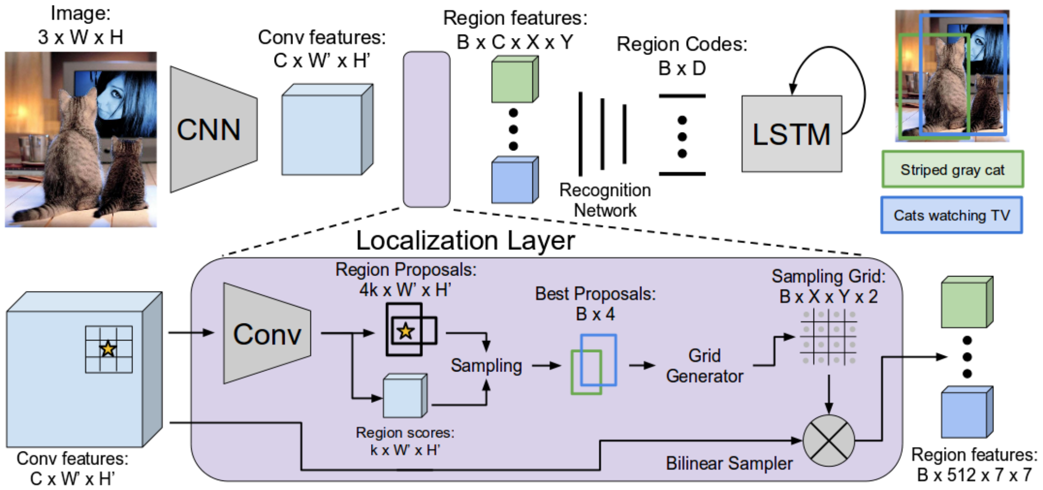


Figure 8.2: Model overview. An input image is first processed a CNN. The Localization Layer proposes regions and smoothly extracts a batch of corresponding activations using bilinear interpolation. These regions are processed with a fully-connected recognition network and described with an RNN language model. The model is trained end-to-end with gradient descent.

8.3.1 Model Architecture

8.3.1.1 Convolutional Network

We use the VGG-16 architecture (15) for its state-of-the-art performance (16). It consists of 13 layers of 3x3 convolutions interspersed with 5 layers of 2x2 max pooling. We remove the final pooling layer, so an input image of shape $3xWxH$ gives rise to a tensor of features of shape $CxW'xH'$ where C = 512, W'= (W/16), and H'= (H/16). 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.

8.3.1.2 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. The approach is based on that of Faster R-CNN, but we replace their ToI pooling mechanism with bilinear interpolation, allowing the 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, but we leave these extensions to future work

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

- **Region Coordinates** : A matrix of shape $Bx4$ giving bounding box coordinates for each output region.
- **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.
- **Region Features**: A tensor of shape $BxCxXxY$ giving features for output regions; is represented by an XxY grid of C -dimensional features.

Convolutional Anchors The localization layer predicts region proposals by regressing offsets from a set of translation-invariant anchors. In particular, we project each point in the $W'xH'$ grid of input features back into the WxH image plane, and con-sider 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 scalars regressing from the anchor to the predicted box coordinates. These are computed by passing the input feature map through a $3x3$ convolution with 256 filters, a rectified linear nonlinearity, and a $1x1$ convolution with $5k$ filters. This results in a tensor of shape $5kxW'xH'$ containing scores and offsets for all anchors.

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 Sampling approach proposed by (17) is followed.

8.3.1.3 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 BxD and passed to the RNN language mode

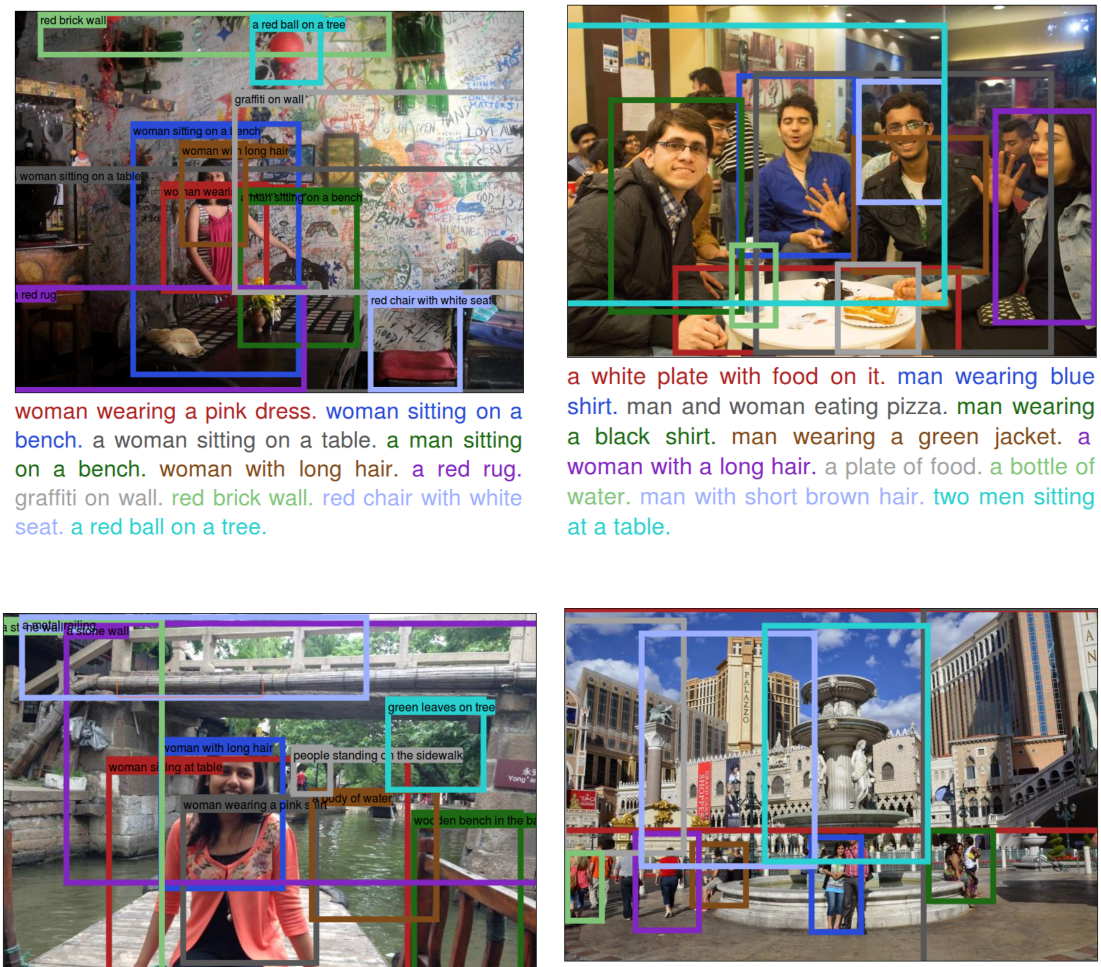
In addition, we allow the recognition network one more chance to refine the confidence and position of each proposal region. It outputs a final scalar confidence of each proposed region and four scalars encoding a final spatial offset to be applied to the region proposal. These two outputs are computed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

8.3.1.4 RNN Language Model

We use the region codes to condition an RNN language model (30), (31), (32)]

8.4 Towards efficient deep CNN implementation on Apache Spark

Apache Spark is an open source cluster computing framework originally developed in the AMPLab at University of California, Berkeley. It is a fast and general engine for large-scale data processing.



woman wearing a pink dress. woman sitting on a bench. a woman sitting on a table. a man sitting on a bench. woman with long hair. a red rug. graffiti on wall. red brick wall. red chair with white seat. a red ball on a tree.

a white plate with food on it. man wearing blue shirt. man and woman eating pizza. man wearing a black shirt. man wearing a green jacket. a woman with a long hair. a plate of food. a bottle of water. man with short brown hair. two men sitting at a table.

woman sitting at table. woman with long hair. woman wearing a pink shirt. wooden bench in the background. a body of water. a stone wall. people standing on the sidewalk. a stone wall. a metal railing. green leaves on tree.

a large building with many windows. man wearing blue shirt. a large building in the background. a person on a skateboard. a man wearing a black shirt. a man in a red shirt. a large building with many windows. a man wearing a red shirt. a large building with many windows. a large stone building.

Figure 8.3: Sample output for Dense Captioning

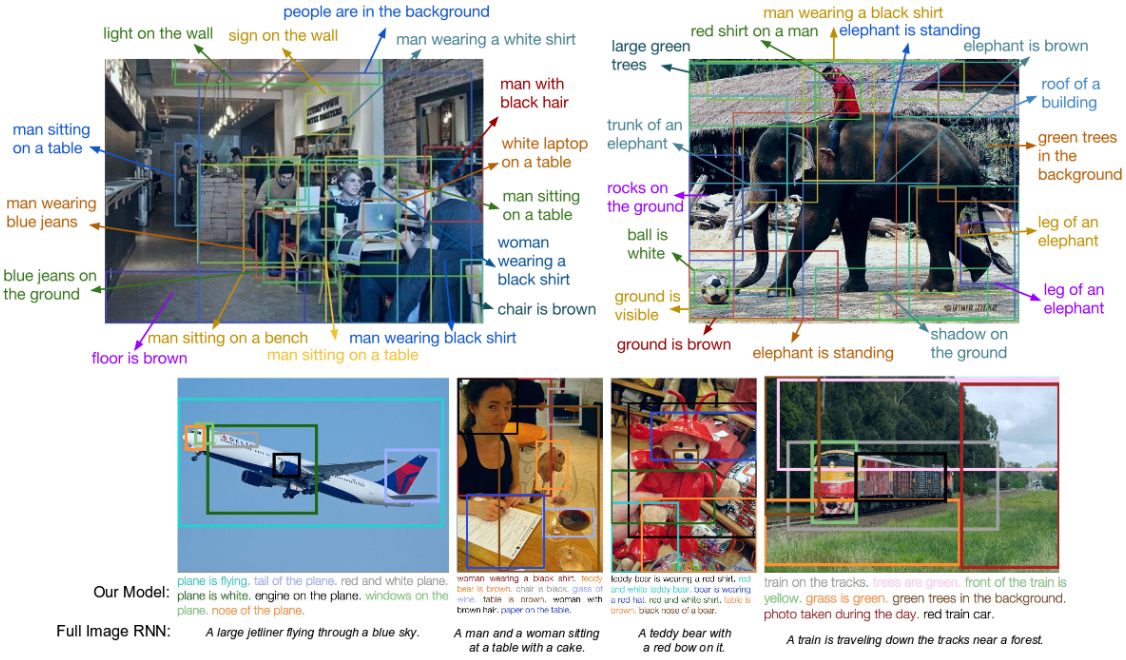

people are in the background
light on the wall
sign on the wall
man wearing a white shirt
man with black hair
white laptop on a table
man sitting on a table
woman wearing a black shirt
chair is brown
man sitting on a table
man wearing blue jeans
blue jeans on the ground
floor is brown
man sitting on a bench
man sitting on a table
man wearing black shirt
man wearing a black shirt
red shirt on a man
elephant is standing
elephant is brown
large green trees
roof of a building
trunk of an elephant
green trees in the background
rocks on the ground
ball is white
leg of an elephant
ground is visible
leg of an elephant
ground is brown
elephant is standing
shadow on the ground
Our Model:
plane is flying. tail of the plane. red and white plane. plane is white. engine on the plane. windows on the plane. nose of the plane.
woman wearing a black shirt. teddy bear is brown. chair is black. glass of wine. table is brown. woman with brown hair. paper on the table.
teddy bear is wearing a red shirt. red and white teddy bear. bear is wearing a red hat. red and white shirt. table is brown. black nose of a bear.
train on the tracks. trees are green. front of the train is yellow. grass is green. green trees in the background. photo taken during the day. red train car.
Full Image RNN:
A large jetliner flying through a blue sky.
A man and a woman sitting at a table with a cake.
A teddy bear with a red bow on it.
A train is traveling down the tracks near a forest.

Figure 8.4: 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.

8.4.1 Core Idea

The Google scientist, Jeffrey Dean promotes one way to large scale data DeepLearning training with distributed platform, named DistBelief. The key idea is model replica, each one takes the same current model parameters, but get the different data shards to train; then each model replica update the gradient to central parameter server.

We intend to splits the train data into different data shards, each one will be trained by the model replica. After all model replica finish the current epoch train, the update gradient will be reduced to update totally; then each model replica will start the next epoch train with new parameter until convergence or get to some stop conditions. The model replica can train the data with different way based on gradient update; eg, mini-batch gradient descent, Conjugate gradient, or L-BFGS.(CG always win the best result). We are using OpenDL for this implementation.

8.4.2 Third Party Software

- The Spark light cluster computing platform. Now we just use the latest version, 1.5.2 just released recently.
- The Mallet, java based machine learning package of UMASS. We use this one mainly for mathematical algorithm, eg, conjugate gradient, L-BFGS.
- JBlas, library of Linear Algebra for Java, refer to http://mikiobraun.github.io/jblas/. It has been used mainly for matrix computation optimization.

Chapter 9

Additional Experiments

9.1 A Neural Algorithm of Artistic Style

I implemented this paper on mixing artist content and style, recently published by Google Research.(13) The system uses neural representations to separate and recombine content and style of arbitrary images, providing a neural algorithm for the creation of artistic images.

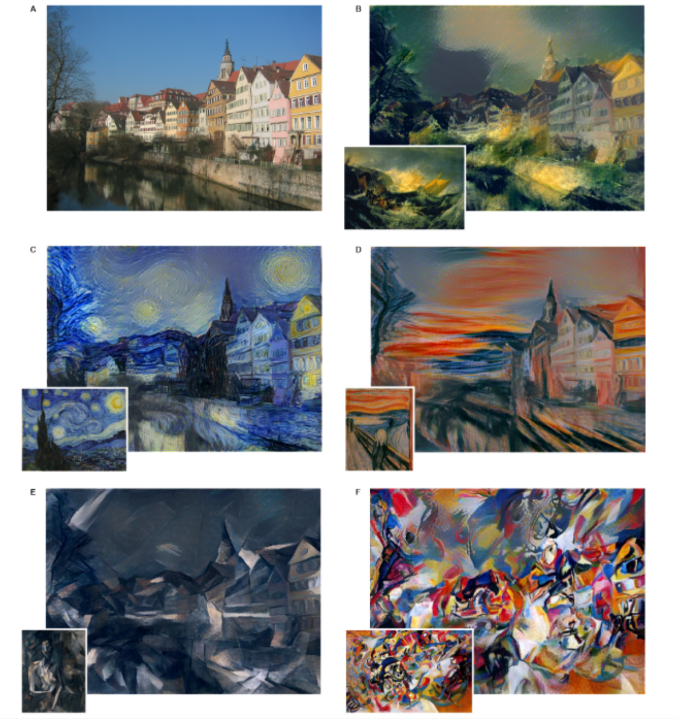


Figure 9.1: A Sample output after mixing different Artistic Styles

Chapter 10

Future Work

We will continue with efficient implementation of deep learning algorithms on Apache Spark for object identification in images. We are also working with the Spark community to integrate CNN and RNN algorithms with the core Spark API. We might also consider going with distributed GPU setup. Then the next step would be to evaluate the performance & compare the performance of the system with other state-of-the-art systems & approaches.

Also, non rectangular bounding boxed for regions in image can also be used for better identification and captioning.

The code for the work will be released soon.

Bibliography

[1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion.* Addison-Wesley, Reading, Massachusetts, 1993.

[2] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. *Zur Elektrodynamik bewegter Körper.* (German) [*Object detection with discriminatively trained part-based models.*]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):16271645, 2010.

[3] Jorge Sanchez and Florent Perronnin. *High-dimensional signature compression for large-scale image classification.* In Computer Vision and Pattern Recognition, 2011

[4] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. *Discriminatively trained deformable part models, release 5.* http://people.cs.uchicago.edu/ rbg/latent-release5/.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. *Imagenet classification with deep convolutional neural networks..* In Computer Vision and Pattern Recognition, 2009.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. *ImageNet: A Large-Scale Hierarchical Image Database..* In Advances in Neural Information Processing Systems 25, 2012.

[7] Quoc V Le, MarcAurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S Corrado, Jeff Dean, and Andrew Y Ng. *Building high-level features using large scale unsupervised learning..* In International Conference on Machine Learning, 2012.

[8] http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html

[9] http://image-net.org/challenges/LSVRC/

[10] http://arxiv.org/pdf/1409.0575v3.pdf

[11] http://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection.pdf

[12] http://spark.apache.org/

[13] http://arxiv.org/pdf/1508.06576v2.pdf

[14] https://arxiv.org/pdf/1511.07571v1.pdf

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

[16] 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 142, April 2015

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: To- wards real-time object detection with region proposal net- works. arXiv preprint arXiv:1506.01497 , 2015.

[18] 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

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

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

[21] L.-J. Li, R. Socher, and L. Fei-Fei. Towards total scene un- derstanding: Classification, annotation and segmentation in an automatic framework. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on , pages 20362043. IEEE, 2009.

[22] L.-J. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In ICCV, 2007.

[23] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In Computer Vision, 2009 IEEE 12th International Conference on, pages 18. IEEE, 2009

[24] S. Fidler, A. Sharma, and R. Urtasun. A sentence is worth a thousand pixels. In CVPR, 2013.

[25] T. Mikolov, M. Karafi at, L. Burget, J. Cernock'y, and S. Khudanpur. Recurrent neural network based language model. In INTERSPEECH, 2010.

[26] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From im- age descriptions to visual denotations: New similarity met- rics for semantic inference over event descriptions. TACL, 2014.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient- based learning applied to document recognition. Proceedings of the IEEE, 86(11):22782324, 1998

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

[29] Deep Visual-Semantic Alignments for Generating Image Descriptions Andrej Karpathy, Li Fei-Fei; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3128-3137

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

[31] T. Mikolov, M. Karafi at, L. Burget, J. Cernock'y, and S. Khu- danpur. Recurrent neural network based language model. In INTERSPEECH, 2010

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