Compressive Visual Question Answering

by

Li-chi Huang

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved August 2017 by the
Graduate Supervisory Committee:

Pavan Turaga, Chair
Yezhou Yang
Baoxin Li

ARIZONA STATE UNIVERSITY

December 2017

ABSTRACT

Compressive sensing theory allows to sense and reconstruct signals/images with lower sampling rate than Nyquist rate. Applications in resource constrained environment stand to benefit from this theory, opening up many possibilities for new applications at the same time. The traditional inference pipeline for computer vision sequence reconstructing the image from compressive measurements. However,the reconstruction process is a computationally expensive step that also provides poor results at high compression rate. There have been several successful attempts to perform inference tasks directly on compressive measurements such as activity recognition. In this thesis, I am interested to tackle a more challenging vision problem - Visual question answering (VQA) without reconstructing the compressive images. I investigate the feasibility of this problem with a series of experiments, and I evaluate proposed methods on a VQA dataset and discuss promising results and direction for future work.

ACKNOWLEDGMENTS

I would like to especially thanks my advisor Dr. Pavan Turaga for all his support and guidance through this research project. His deep insights in computer vision guide me through the process of my thesis work. I also want to especially thanks Dr. Kuldeep Kulkarni for all his support and valuable comments.

Also, I would like to thanks my committee members Dr. Baoxin Li and Dr. Yezhou Yang for their times, and provide their professional insight to examine my thesis work.

Finally, I want to thanks all lab members in our lab for their comments and being such a great co-worker.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Motivation

Compressive sensing has been a popular research topic recently for computational imaging research community, with many prototypes of compressive imagers have been developed such as Single-Pixel-Camera [35], compressive light-field imaging [26] and compressive lenseless imaging [16]. These imagers take projections of underlying signals to form measurements, resulting in smaller amount of data storage than traditional image acquisition methods. Because of its low requirement for data storage, compressive imager have an advantage in resource-constrained environments such as surveillance.

Traditional research in computer vision inference usually takes rectangular arrays of pixels as input. Therefore, if one would like to facilitate computer vision inference task with a compressive imager, one need to first reconstruct images using a reconstruction algorithm, then apply the computer vision algorithm to fulfill the purpose. Although compressive sensing theory allows nearly perfect reconstruction from compressive measurement, the reconstruction process is usually computationally expensive and reconstruction result is somewhat degraded at high compression rates. Therefore, the method that take compressive measurement as input is desirable for cut down the computational cost, and thus open up the possibility to perform computer vision inference task with novel compressive imagers. Previous works have successfully tackled several computer vision inference problems using compressive measurements. For example, Kulkarni and Turaga [20] provide a reconstruction-free solution for activity recognition. Lohit *et al.* [22] try to directly perform image classification on compressive measurement. I am interested in solving a much more complex inference problem called visual question answering, and explore the possibiity to extract such information from compressive measurements.

Visual question answering (VQA) is the task that answer question about an image. VQA is a task that involves natural language processing, question answering, object recognition and semantic interpretation. The complex nature of this task make it to be considered as an AI complete task [1]. This topic have been researched extensively in the natural language processing and computer vision research community, many successful attempts have been addressed and achieved solid results for VQA task. In this work, I would like to investigate the utility of compressive imagers for a complex computer vision tasks such as VQA.

1.2 Compressive Sensing

In this section, I will give an brief introduction to the theory and idea of compressive sensing. Compressive sensing is a novel data acquisition paradigm that goes beyond traditional sampling method following Nyquist's theorem, CS theory claims that perfect reconstruction is possible from a sampled signal with fewer samples than necessary amount compared to the NyquistShannon sampling theorem claims. To illustrate this mechanism, let's suppose original signal is x ,where $x \in \mathbb{R}^N$ forms a projection y ,where $y \in \mathbb{R}^M$ so that

$$y = \phi x \tag{1.1}$$

where ϕ is measurement matrix. According to CS theory, the possibility of recovery from measurement y given $M < N$ requires sparsity of x and incoherence of measurement matrix ϕ.

A real signal can usually be represented with an othornormal basis, the sparsity convey the idea that larger coefficients often be able to capture most information of the signal. Suppose we have a signal f, represented as expansion of basis in the following form,

$$f = \psi x \tag{1.2}$$

where ψ is sparse basis and The above equation indicate the possibility to approximate the original signal while discarding small coefficients. Assume one keep S largest coefficients and set the rest of coefficients to zero, it is called S-sparse. The goal is to approximate original signal with x_s, so that $f_s := \psi x_s$.Moreover, for x to be compressible, the sorted magnitude of x_i need to decrease rapidly, so that $||f - f_s||$ is negligible given $||f - f_s|| = ||x - x_s||$.

As we see in the equation 1.1 and equation 1.2, the choice of sensing matrix and sparse basis can be various. The property of incoherence limits the valid pair of matrices. The coherence [4] between sparse basis and sensing matrix is denoted as

$$\mu(\phi, \psi) = \sqrt{n} \cdot max|\langle \varphi_k, \psi_j \rangle| \tag{1.3}$$

the above equation basically measures the largest correlation value between any two elements in ϕ and ψ. The incoherence property in compressive sensing theory requires a low coherence pair to work. Random matrices are thus to be one of desirable choices to be sensing matrix since they are incoherent to any fixed basis ψ.

How much the signal can be compressed is the main concern of compressive sensing theory. The minimum number of measurements m allowed relative to n is subject to the following relation

$$m \geq C\mu^2(\phi, \psi)S \log n \tag{1.4}$$

where ψ is S-sparse and C is for some constant larger than zero. As seen in 1.4, coherence plays a crucial role in this theory, the smaller the coherence the fewer the number of measurements needed. As for recovery, one can solve a convex optimization problem of coefficient sequence x to minimize the l_1 norm. [4, 7]

To examine the robustness of compressive sensing, a key concept for sensing matrix design is called restricted isometry property(RIP) [6]. The definition of RIP is as following; the isometry constant δ_S for each S=1,2...S of matrix A such that

$$(1-\delta_S)||x||^2 \leq ||Ax||^2 \leq (1+\delta_S)||x||^2 \tag{1.5}$$

is valid for all S-sparse signal x. When sensing matrix A has RIP property, transformation A on x preserve the Euclidean length of S-sparse signals, indicating that S-sparse vectors x cannot be in the null space of A. If we wish to obtain x with sensing matrix A, all pairwise distance between x must be preserved in the measurement space. That is, for a constant δ_{2S} which is less than one,

$$(1-\delta_{2S})||x_1-x_2||^2 \leq ||Ax_1-Ax_2||^2 \leq (1+\delta_{2S})||x_1-x_2||^2 \tag{1.6}$$

Chapter 2

BACKGROUND

In this chapter, I will outline background for my work, including inference using compressive measurement and visual question answering.

2.1 Compressed Learning

Given the fact that compressive measurements are far fewer than original data dimensionality, conducting machine learning experiments in measurement domain is equivalent to a dimensionality reduction for machine learning problem. Generally, one has to reconstruct data from measurement domain before applying machine learning algorithm since standard algorithms usually take original data as input. Learning directly on measurement domain can avoid cost of recovery to data domain, and create a desirable property for this approach. Calderbank et al. [5] prove the performance of the SVM classifier in measurement domain is nearly the same as in the data domain, thus conclude compressed learning is a possible approach. There have been research activities in other machine learning task such as palmprint recogniton [14], face recognition [37] and image classification [22]. In a real-world setting, compressive sensing is particularly useful in resource constrained environments with limited computational power and storage. Problems like action recognition were addressed in [20] opening up the potential application in surveillance and monitoring.

2.2 Visual Question Answering

Visual Question Answering(VQA) is a complex inference task which concerns answering free-form questions based on visual information in images. Unlike traditional computer vision tasks such as object recognition and image classification that tend to be task specific, VQA is the challenge that require combination of computer vision, natural language processing and knowledge representation. VQA is considered as an "AI complete" task [1] that aims to push development of modern artificial intelligence. As advancement of natural language and computer vision research, the research on intersection of both fields draws more and more attention in the research community. Image captioning [36, 11], which generate description of images, is one of the successful attempts to integrate the field of computer vision and natural language processing. However, visual question answering is significantly more complex than image captioning since it require exact information requested in the question, thus needs deeper level of understanding of both images and language. Textual question answering is also a task that is closely related to VQA. Instead of textual information as input, images as input creates much higher dimensions to the problem than just text, and inference on image is much difficult than text, since text usually has a well-understood grammatical structure, while image structure is noisy. In the remaining of this section, I will describe previous works to tackle VQA task, and it will be structured in the following. First, I will introduce two basic components of VQA, which are image embedding and question embedding. Then, I will conduct a brief survey of previous works to interact with those two features. Generally, approaches for VQA can be categorized into three categories– joint embedding approaches, attention mechanism and external knowledge based.

2.2.1 Image Embedding

In traditional computer vision, SIFT [23] and HOG [10] features descriptor is often employed to extract features from images. These traditional methods are computationally efficient but fail to generalize the description of image, partly because they have less parameters in descriptor. Malinowski and Fritz [24], utilized image segmentation algorithms to find the objects in the images. However, this method along with SIFT and HOG feature is hand-crafted. In recent times, neural architectures have aimed to address this issue and provide more powerful representation.

Convolutional neural network is the neural network containing the convolutional layer. Instead of performing multiplication in the case of fully-connected layer, convolutional layer perform convolution in each filter to get local feature of the image. Convolutional neural network gained a lot of success in computer vision task recently, such as object recognition, activity recognition and event detection [40].Accordingly, CNN makes a natural choice for feature extraction techniques to represent the image feature. In practice, CNN have been trained on images classification on large database is used to be served as image feature. In term of architecture of CNN, Large CNN such as VGGNet [31], GoogleNet [34] and ResNet [13] are usually chosen given much powerful image representation than smaller network.

2.2.2 Question Embedding

Semantic representation of language is a long-standing problem in natural language processing. Traditional approaches such as Bag-of-words and Tf-Idf are used to represent sentences or documents. These approaches rely on word occurrence in the database of documents to model the documents, and still prevalent approaches in text mining field. Some of the early work in VQA adopted BOW [42] to represent text feature and shows impressive results for VQA task. However, the drawback for these approaches is its lack of semantic representation in words, since these model only count the frequency or the number of appearance of words in the documents. Semantic parser is a way to introduce semantic meaning to the word by parsing the word into the labeled parsed tree to capture the semantic relations between words.

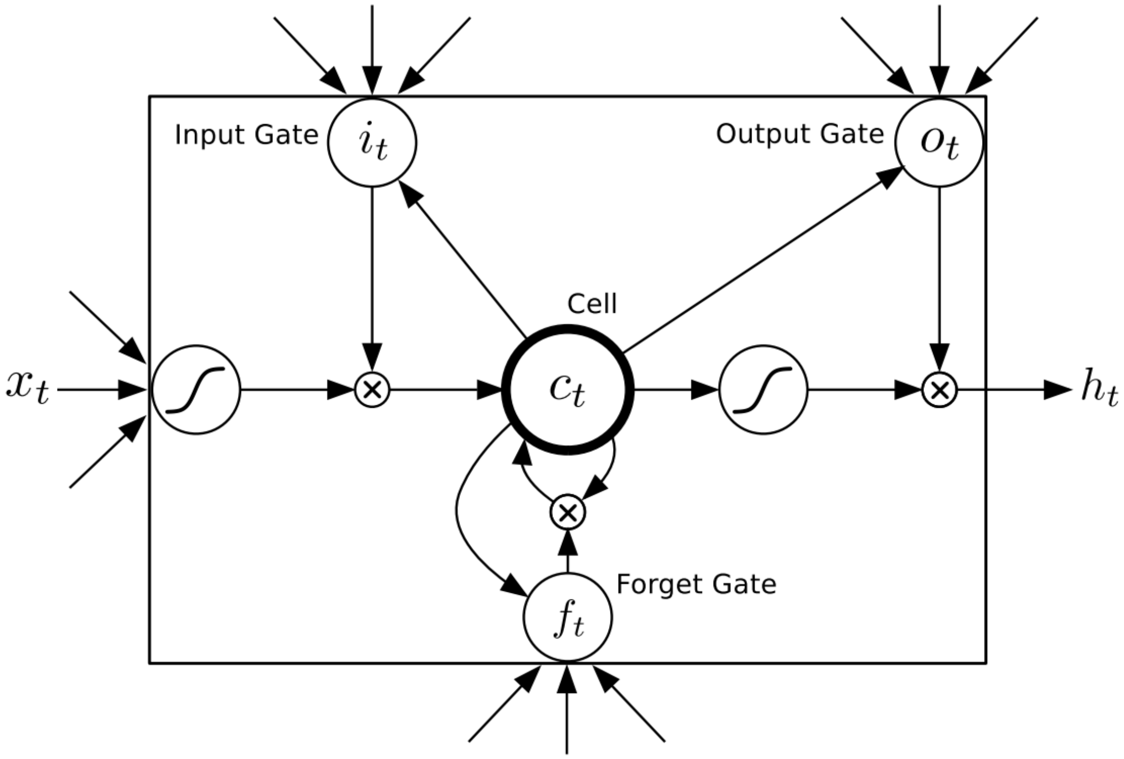

Figure 2.1: LSTM Memory Block

Socher *et al.* propose a method called recursive tensor network to represent sentences in vector space. However, the parsing method require parsing the entire document, make it computationally expensive and language dependent. Word embedding aims to address the problem by providing efficient word embedding approaches, word2vec [27] employs cbow and skip-gram architecture to learn the word embedding through fully-connected neural network. They shows that their approach outperform the N-grams model and recursive tensor network while eliminate the need for parsing whole documents in advance. Regarding sentence and document embedding, doc2vec [21] utilize similar architecture of word2vec to obtain document embedding. However, recurrent neural network is gaining popularity for NLP tasks due to its superior performance in modeling sequences of words.

Recurrent neural network is similar to multi-layer perceptron but its hidden layer have weight between each units at adjacent time step, making the unit have memory about the previous input, and thus make it suitable to model sequence. However, the typical recurrent neural network often suffer from gradient vanishing problem[15], which refer to the sensitivity of the network decaying exponentially through the recurrent units, thus making RNN difficult to train. The variants of RNN, LSTM[15] and GRU[9] help resolve the vanishing gradient issue with gate mechanism. Since LSTM and GRU are basic components in the VQA pipeline, I will give a brief introduction of them in the following paragraph, more detail description of LSTM will be presented in the chapter 2.

LSTM, short for Long-short-term memory, consists of memory cells to form recurrent neural network. A memory cell connects the output gate to another memory cell to pass memory to the network. The basic component for LSTM is the memory block, each block contains one cell and three gate– input gate, output gate and forget gate. The schematics of memory block is shown in Figure 2.1 [29]. The advantage of LSTM over traditional RNN is that their three gates allow information to be stored or discarded in long periods of time, tackling the problem of vanishing gradient. The idea of GRU(gate recurrent unit) is similar to LSTM, try to add the previous memory to the current memory. The memory block of GRU is the difference between LSTM and GRU, the block of GRU consists of reset and update gates as seen in Figure 2.2 [9]. Consequently, GRU cuts down multiplicative gates in LSTM to two, while preserve the advantage over LSTM of adding to the previous memory. The smaller amount of parameters than LSTM thereby make GRU more computationally efficient. More details of GRU performance can be seen in [9].

2.2.3 Related Works

Once we get the image feature from CNN and text feature from RNN, we try to merge these two embeddings to facilitate reasoning. I will outline some previous

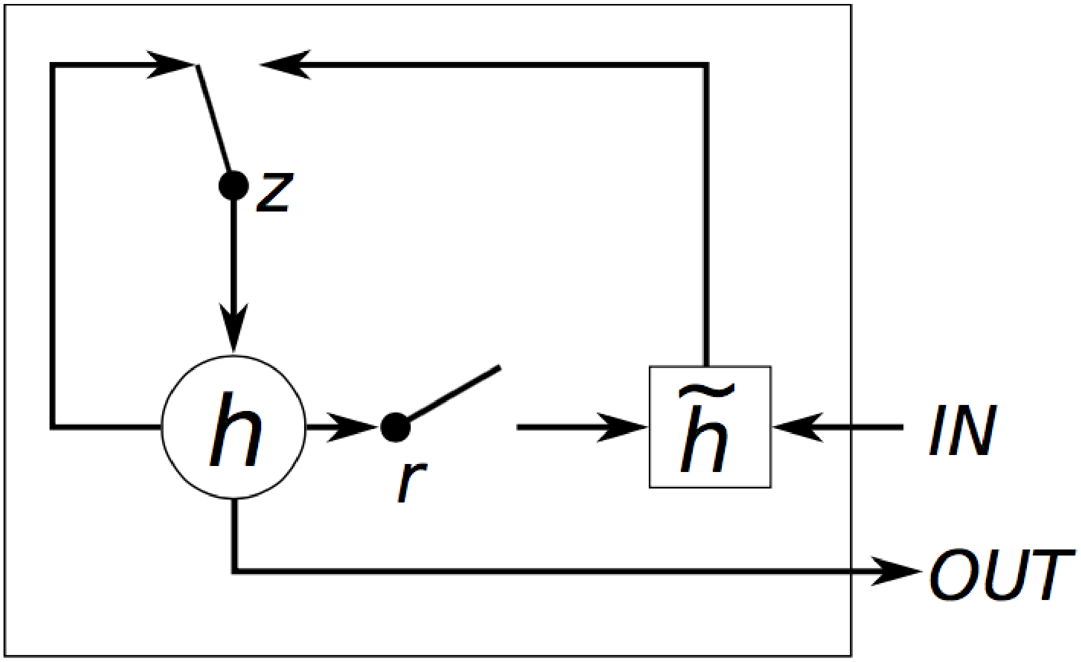

Figure 2.2: GRU Memory Block

works that use this concept to illustrate the idea. Malinowski *et al.* [25] utilize LSTM to generate answers directly for VQA task. Pre-trained CNN for image recognition is employed to produce image feature, while question feature is produced by concatenation of question word embedding and previous ground truth answer. Then it concatenates the image and question feature to serve as inputs to LSTM. At the answer generating phase, the predicted answer is concatenated with question word embedding to feed into LSTM. Noh *et al.* [28] proposes an approach that consist of classification network and parameter prediction network. Classification network include VGGNet, Dynamic parameter layer and fully-connected layer, it treats VQA as a classification task. Parameter prediction network consists of GRUs that feed with question word embeddings to generate the candidate weights. They employ parameter hashing to project the parameters in candidate weight to dynamic parameter layer, thereby allows question feature to project in the image feature. It proposes a new method that would interact with question and image feature, and outperform standard methods [25].

Bilinear pooling denotes the outer product of two vectors, which allow much richer representation power of two vector than element-wise multiplication or concatenation. However, the outer product of two embeddings ends up creating a large number of parameters in the network, requiring huge memory to train the network. [12, 17, 2] utilize approaches to reduce the number of parameters to fit the memory constraint. Fukui *et al.* [12] leverage count sketch projection to express count sketch of the outer product as a convolution of two count sketches, therefore avoiding computing outer product directly. Kim *et al.* propose a low-rank method that splits the weight matrix into two low-rank matrices. Two feature vectors then are computed by Hadamard product to represent joint embedding. Ben-younes *et al.* [2] utilize Tucker decomposition to decompose outer product into three factor matrices and a core tensor. They show that Fukui *et al.* [12] and Kim *et al.* [17] are special cases of their method, and also achieve state-of-the-art result at the time of writing.

Attention mechanism aims to incorporate local features into the reasoning process. Methods without attention mechanism use only the global feature to represent the visual information, while global feature may prone to introduce noisy information to the reasoning process. Inspired by the method used in image captioning, Attention mechanism tries to address this issue by extracting local features from image, and assigns different regions weight to allow reasoning over local image features. Xu and Saenko [39] proposes a method called ”spatial memory network”. They take a concatenation of word embedding as question vector, and extract image feature from Googlenet in dimension of $L \times M$, which preserve the local regions feature. The image and question feature then pass through a weight W_A to generate attention embedding. The evidence embedding take the image feature aims to recognize the semantic concepts such as objects. By performing element-wise multiplication of the evidence and attention embedding, spatial memory network learns the heatmap for the semantic concept, therefore benefiting the reasoning process. They employ multiple hops to deepen the reasoning process. Yang *et al.* [41] employ similar idea as [39] but use LSTM as question feature instead. They called the method ”SAN” that perform inference in multiple attention layers.

External knowledge based methods target to remedy the issue of insufficient knowledge in terms of answering the question. For example, to answer the question ”Why is the apple falling?”, one need to know the concept of gravity and apple. Wu *et al.* propose a joint embedding approach that combining the Doc2vec [21] to obtain the external knowledge.

Chapter 3

METHODS

In this chapter, I will detail my proposed frameworks for the VQA task. This chapter is outlined as following order, I will first introduce the image recognition model for image classification task, which will then be used to develop the VQA pipeline.

3.1 Compressive Measurement Recognition Model

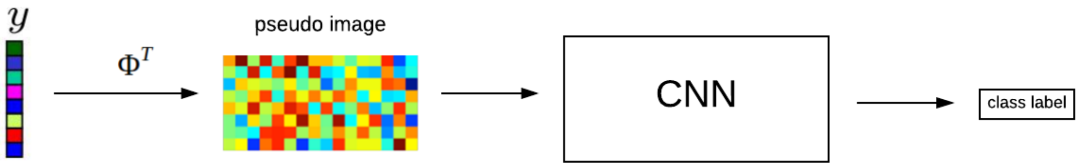

Figure 3.1: Pipeline for Compressive Recognition Model

Convolutional neural networks have achieved very good performance on many computer vision tasks such as classification and object recognition. Therefore, I would like to leverage CNN to perform image recognition using compressive measurements. However, compressive measurment is in dimension of $m \times 1$, which m denote as the number of measurements, while CNN operates on 2-dimensional array of pixels. Therefore, I use a linear projection inspired by [22], transpose of sensing matrix ϕ, to the compressive measurement transforming 1-D compressive measurements into 2-D ”pseudo image”. Pseudo images then feed to CNN to perform the image recognition task. I conduct the experiment on image classification to test the amount of information we could extract from the pseudo images. The overall pipeline for classification task can as seen in Figure 3.1. Moreover, I also use block-based linear projection to

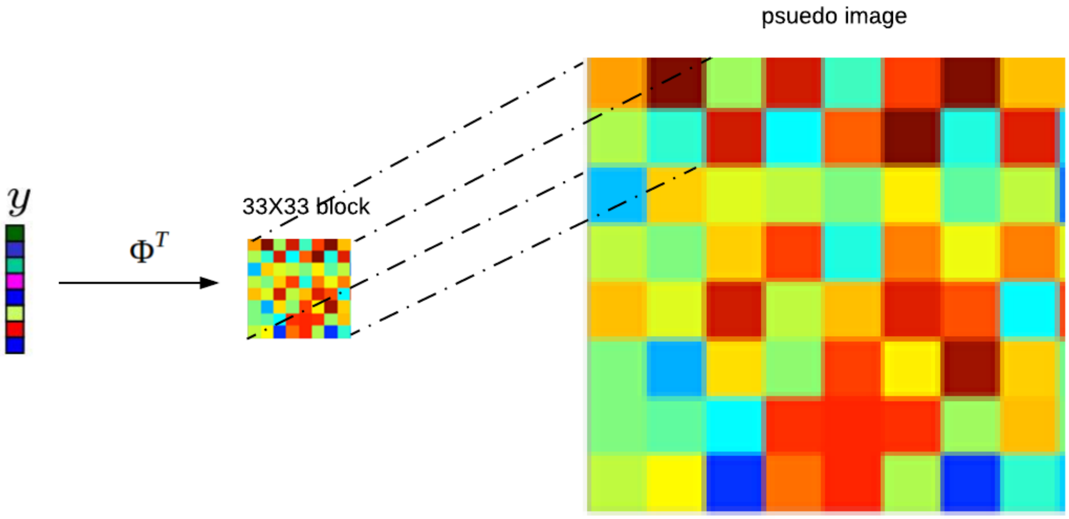

Figure 3.2: Generation of Block-based Pseudo Image

investigate the projection technique of compressive measurements. I set the block size at 33×33 pixels, and project compressive measurements for each block into block pseudo image separately, and then put them in order to form a pseudo image as in Figure 3.2. The CNN architecture I adopt for the classification task is Googlenet [34] and Resnet [13]. The classification experiment result will be discussed in next chapter.

3.2 VQA Pipeline

To tackle the VQA task, extraction of informative image feature is necessary. The pretrained networks such as VGGNet and ResNet are usually used to extract image features. I employ a trained googlenet on simulated Imagenet dataset mentioned in section 3.1, to generate the image embedding. Regarding the question feature, I feed the questions into a RNN [15] to extract the textual information in the questions. In this section, I would like to first describe the model to produce question embedding since it is the same for both methods, then I will describe two methods I adopted to

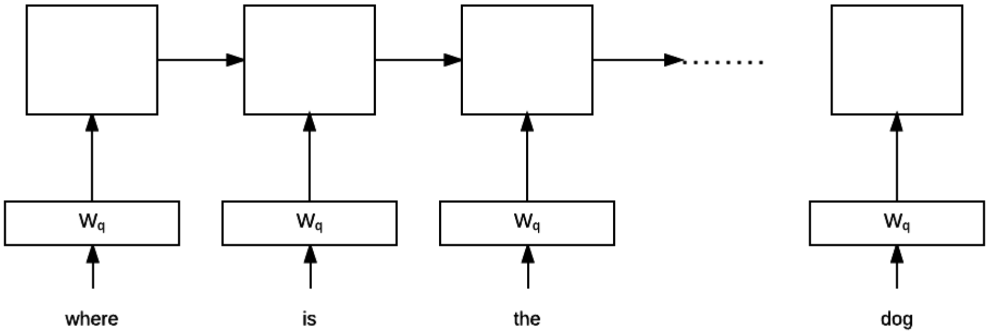

Figure 3.3: LSTM for Modeling Questions

combine image and textual features in order to tackle compressive VQA later.

3.2.1 Question Embedding

In section 2.2.2, I briefly introduce the idea of RNN and LSTM, and address the importance of LSTM in sequence modeling task; Because of this, it make sense to employ LSTM to model the question and generate textual representation for the VQA task. Assume the question with N words $q = [w_0, w_1, ..., w_N]$ is expressed as a sequence of word embeddings [1]. I project the word embedding to the vector space via weight matrix W_q to form the word embedding to represent each word in the question. That is, for the word in position t, the word embedding for w_t is $x_t = W_q w_t$. Then, LSTM units take the word embedding vector as a input to perform learning process, overall structure for question modeling is shown in 3.3.

As discussed in chapter 1, the essential building block of LSTM is a memory cell as shown in Figure 2.1. The memory cell consists of three gates to control the amount of information flow, a cell state to sustain memory and a hidden state as output to the next memory cell. The memory cell adopts sigmoid function σ for gates. Input gate i_t regulates the amount of updates in current cell state from current input x_t

and previous hidden state h_t as in 3.1.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{3.1}$$

Forget gate f_t decides the portion of information from previous cell state c_{t-1} memorize in current cell state as in 3.2.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{3.2}$$

Output gate controls the amount of information in the current cell state output to hidden state, given as:

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{3.3}$$

The update procedure for cell state and hidden state is controled by gates with $tanh$ activation layer in input and output gate.

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{3.4}$$

$$h_t = o_t tanh(c_t) \tag{3.5}$$

where $x_t = W_q w_t$. The final hidden state h_N is used as representation for the whole question $q = [q_0, q_1 .... q_N]$.

3.2.2 LSTM CNN: Element-wise Multiplication Feature Fusion

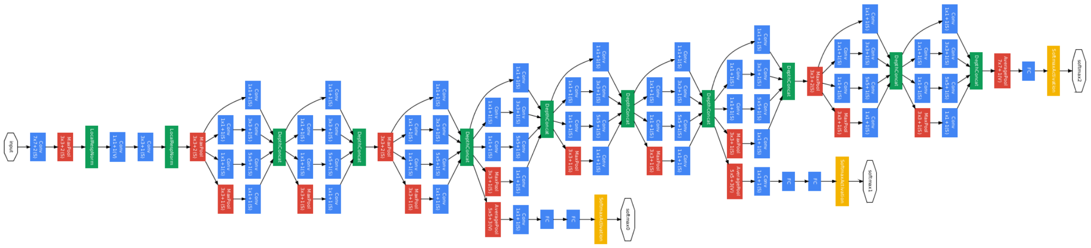

Figure 3.4: Illustration of Googlenet Architecture

Regarding image feature, I adopt CNN to generate image embedding. The CNN architecture I employ is Googlenet [34], and detailed architecture of googlenet is shown in Figure 3.4. To extract the image feature, trained image recognition model on compressive measurements is used as described in section 3.1. As discussed in the section 3.1, I take pseudo images as input data for the CNN network, then feed-forward the trained model to the last average pooling layer to yield image embedding for this method, which can be seen in Figure 3.1. The dimension of the extracted image feature is 1024.

A single layer perceptron is employed after image embedding and question embedding to project the embedding to a vector with dimension 1024. Then these two projected vectors perform element-wise multiplication to merge the two features. A softmax layer is employed after a single layer perceptron as merged feature with output dimension 1000. The output of the softmax layer generates the probability of possible answers for classification and generate the answer for the question. The overall pipeline for this method [1] is shown in Figure 3.5.

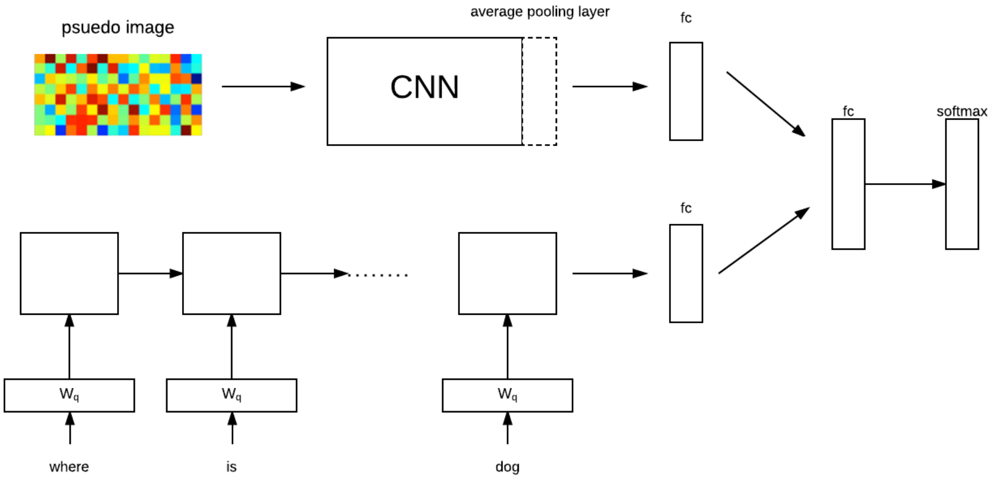

Figure 3.5: Overall Architecture of LSTM CNN Method

3.2.3 Stacked Attention Model

This method aims to leverage attention mechanism [41] in order to capture the local information in the image, and the relationship of these regions. First, I adopt the trained googlenet as before to generate the image feature. However, I feedforward the trained googlenet to the output of the last inception module, which is ”DepthConcat layer” as shown in the 3.4. Since the input of the CNN is cropped image in dimension of 243×243, the dimension of the extracted image feature is $1024 \times 7 \times 7$. This image feature is to create a feature that can represent 49 local regions on the pseudo image, each with 1024 dimensional representation.

Given the image feature from Googlenet and question embedding produced from LSTM, I employ attention layers taking these two vectors as input, and wish to perform inference by learning the weight on each region of image feature according to question feature, the overall architecture of the model is shown in Figure 3.6. The detailed information of the attention layers is as following. Assume the question embedding extracted from LSTM is v_q, and image feature is v_I, I tile the question

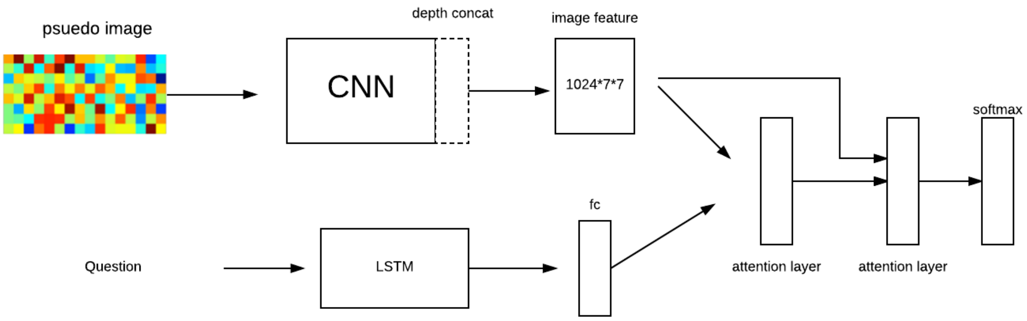

Figure 3.6: Architecture of Stacked Attention Model

vector by the number of image feature regions to the dimension of 1024×49, denoted as v_Q. Performing element-wise addition on image and question embedding followed by a *tanh* activation layer to merge these two embeddings:

$$W_A = tanh(W_I v_I + (W_Q v_Q + b_Q)) \tag{3.6}$$

where $W_I, W_Q \in R^{a \times d}$, a is output size of attention layer, and d is dimension of feature for each image regions. $b_Q \in R^a$ is a bias term for question feature. Suppose image and question feature $v_I, v_Q \in R^{d \times m}$, m is the number of image regions. Attention matrix is thus $W_A \in R^{a \times m}$.

Attention matrix is used to capture the question and image weight over the image regions. A softmax layer is employed after the attention matrix to generate the probability distribution over the image regions.

$$P_I = softmax(W_P W_A + b_P) \tag{3.7}$$

where $P_i \in R^m$, $W_P \in R^{m \times a}$ $b_P \in R^m$.

Probability distribution P_I then multiplies with image feature vector to obtain the weighted image feature vector. Then I combine the weighted sum of image features over the image regions $v_{Iw}$with the question embedding $v_q$to generate the output of the attention layer r.

$$r = v_{Iw} + v_q \tag{3.8}$$

where $v_{Iw} = \sum_i^m p_i v_i$,and v_i denote as image feature in i_{th} image region. $v_q, v_{Iw} \in R^I$, I is the dimension of image feature, thus the output of attention layer $r \in R^I$.

To perform further reasoning, I stack another attention layer to the first attention layer as shown in Figure 3.6. This operation intends to refine the attention process with attention layer taking output of the first attention layer r as input. Similar to the first attention layer, the detailed process is as following.

$$W_{A2} = tanh(W_{I2} v_I + (W_{Q2} v_Q + b_{Q2})) \tag{3.9}$$

$$P_{I2} = softmax(W_{P2} W_{A2} + b_{P2}) \tag{3.10}$$

$$r_2 = v_{Iw2} + r \tag{3.11}$$

where $v_{Iw2} = \sum_i^m p_{i2} v_i$. Note that in Eq. 3.11 combine refined weighted sum image vector v_{Iw2} and output vector r from previous attention layer, this allows further reasoning on the top of result from previous attention layer.

Finally, the output embedding r_2 is fed into a single layer perceptron to perform classification task and generate the answer.

$$W_a ns = softmax(W_f r_2 + b_f) \tag{3.12}$$

3.2.4 Reconstrction-based VQA Using ReconNet

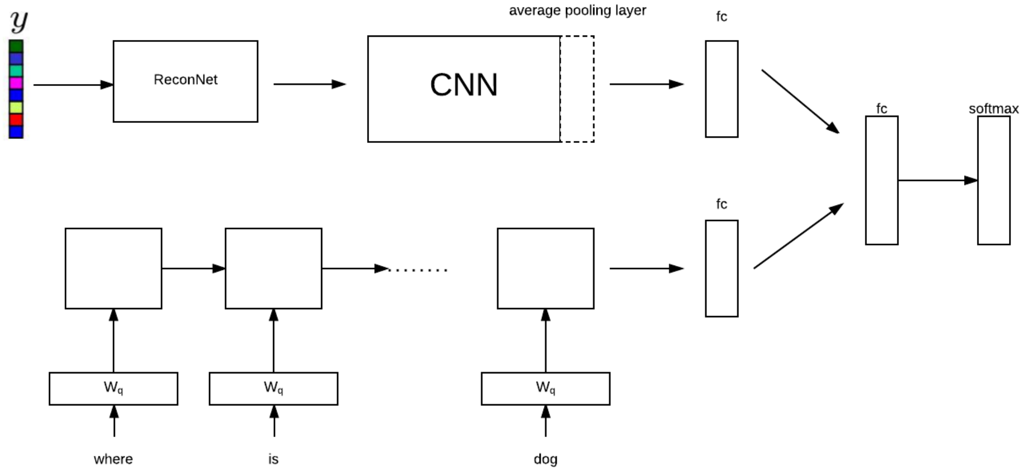

Figure 3.7: Architecture of ReconNet LSTM CNN

ReconNet [19] is a compressive sensing reconstruction approach using convolutional neural network to reconstruct compressive measurements to images. It shows potential advantage over iterative recontrction algorithm in terms of time complexity[19], and offer better quality of recontruction than traditional reconstruction algorithms. Therefore, this method serves the following purposes: First, this method examines the utility of reconstruction first before performing high-level inference task. [19] already shows the object tracking task in video with decent result compare to original video, I would like to examine the performance in the task with much higher dimensional problem like VQA. Second, the comparison of this method and the method without reconstruction can give us full picture of compressive VQA.

The framework for this method is as following : ReconNet is stacked to CNN as input and generate the image feature after reconstruction of compressive measurements. The rest of architecture perform element-wise multiplication between question and image embedding as decribed in section 3.2.2. The overall architecture is shown in

Figure 3.7. In similar fashion, ReconNet can be used to stack with stacked attention model as well, I will show all results in chapter 3.

Chapter 4

EXPERIMENTS AND RESULTS

In this chapter, I will discuss experiments in detail for the compressive image recognition task and compressive Visual Question Answering task. I will first describe the dataset and evaluation method for each task, then the experimental setup for training the model. Finally, I will present the results for these experiments and discuss my experimental outcomes.

4.1 Compressive Image Recognition

4.1.1 Compressive Imagenet Dataset

The dataset I adopt for the compressive image recognition task is ImageNet Large Scale Visual Recognition Challenge 2012 dataset (ILSVR2012), which has a large number of training examples and diversity to examine generalization of the model under scrutiny. This is a well-known image recognition dataset in the imagenet database [30]. The training set is comprised of 1.2 millions images with 1000 categories of object. The validation task consists of 50000 images. Similar to [22], I utilize Hadamard matrix as a measurement matrix to simulate compressive sensing measurements. Then I iterate this process for the whole imagenet dataset to serve as a simulated dataset for image recognition task.

4.1.2 Baselines and Evaluation Metrics

I evaluate the top-1 classification accuracy as evaluation metric for image recognition. To demonstrate the validity of my proposed framework, I compare the experimental results of my model with the model after reconstructing the compressive measurements using ReconNet [19]. Moreover, I will also compare my result with the pre-trained googlenet on original image for imagenet.

4.1.3 Experimental Setup

As mentioned earlier, I employ googlenet [34] and Resnet [13] to perform the image classification task. A Hadamard matrix is used as sensing matrix, the compression ratio of the sensing matrix is 0.25. A linear projection is posed to produce 256×256 pseudo images as input to the network. The batch size is 32, each batch augments the data by cropping images to 243×243 and using mirror reflections. For googlenet, I use stochastic gradient descent as optimizer with momentum 0.9. I adopt the step size decay policy to adjust the learning rate, learning rate decay by the factor of 0.8 for every 80000 iterations. For Resnet, I adopt the 50 layers Resnet, and stochastic gradient descent with momentum 0.9 is used. The batch size is 50. Learning rate policy is step size decay policy, learning rate decay by 0.96 for every 320000 iterations. Dropout [33]layer with 0.5 dropout ratio is used at the end of fully-connected layer to tackle overfitting.

4.1.4 Results and Discussion

projection	accuracy
$\phi^T \phi x$	48.7
block-based $\phi^T \phi x$	48.5
ReconNet + GoogleNet(no finetuned)	35.68
ReconNet + GoogleNet(finetuned)	64.1
uncompressed [3]	68.7

Table 4.1: Googlenet Image Recognition Result on Different Levels of Projection for Compressive Measurement

The image classification task results on data with different projection techniques for compressive measurements is presented in Table 4.1.4, where the image recognition results with linear transpose of measurement matrix denote as $\phi^T\phi x$, and block-based projection with size of 33×33 measurement matrix as described in section 3.1 denote as block-based $\phi^T\phi x$. The difference in performance of models using these two projection techniques seems insignificant(0.2%), which may imply the fact that projection of compressive measurement with small 32×32 block does not yield the pseudo image in better resolution than whole measurement projection in term of image classification task. Regarding the experimental result using Resnet-50, it achieve 48.5% accuracy with $\phi^T\phi x$ projection, which obtains similar performance as Googlenet's result.

As the baseline, the image recognition results after reconstruction using ReconNet also present in Table 4.1.4. The image recognition result using the pretrained Googlenet on original imagenet dataset, which is denoted as "no finetuned" in the Table 4.1.4, experience 33.02 % drop in accuracy compared to the pretrained googlenet model with original imagenet dataset. The reason why the pretrained model with ReconNet has such a big difference in performance may be because reconstructed images by ReconNet are not exact reconstructions from compressive measurements but rather a blurred version of original images. However, the accuracy rises to 64 % when I finetuned the model; it shows that ReconNet reconstructed images still contain a decent amount of information for image recognition task and validate my previous observation. All these trained models are used in the next section to compute image features for the VQA task, which I will discuss in detail in next section.

4.2 Compressive Visual Question Answering

4.2.1 VQA Dataset

VQA [1] is the dataset based on the images in Microsoft Common Objects in Context (MS COCO), which contains 83783 training images and 40504 validation images. They provide three questions for each image, so there are 248349 questions for the training set and 121512 questions for the validation set. Answers for questions are generated by humans (Amazon turker), 10 answers are provided for each question from unique workers. Answers are generally open-ended, types of answers are generally classified as "yes and no", "number" and "other" answers. I adopt the validation set to test the performance of my method.

To generate the simulated compressive measurements for images in VQA dataset, I used a random Gaussian matrix as a sensing matrix to project whole images in the dataset. As stated in section 3.1, the transpose of a sensing matrix and block-based projection is used to project compressive measurements to pseudo images.

4.2.2 Baselines and Evaluation Metrics

The evaluation metric for open-ended task in VQA dataset given a generated answer is as following:

$$accuracy = min(\frac{\text{\# of match to human provided answer}}{3}, 1) \tag{4.1}$$

this evaluation metric basically gives the answer full credit if there are at least three (out of ten) answers provided by workers match the generated answer. If the generated answer matches with less than three answers, it will get partial credit as shown in Eq. 4.1.

To examine the validity of our image feature, I generate simulated compressive

VQA dataset with Gaussian matrix as sensing matrix, then feed the compressive measurements as image feature directly into the architecture mentioned in section 3.2.2 to serve as baseline for compressive VQA task. Also, I will compare my methods to the baseline and method in [1], such as question feature only baseline, in order to validate the performance of my methods and compare to method using uncompressed images .

4.2.3 Experimental Setup

For image feature, I extract the image feature from trained GoogleNet on simulated imagenet dataset as discussed in 3.1. For LSTM CNN method, image feature with dimension 1024 is extracted as described in section 3.2.2. For stacked attention model, image feature with dimension $1024 \times 7 \times 7$ is obtained to represent 49 local region features as described in section 3.2.3.

For question feature, two layers of LSTM are stacked together and LSTM's dimension is 512 for cell and hidden state. I set dimension of word embedding for each word of the question to be 200.

I use the top 1000 most frequent answers as possible outputs that covers 82.67% of all answers, as the same in [1]. Regarding the optimizer, all models adopt Adam optimizer [18], with $\epsilon = 10^{-8}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ as values of configurations. The batch size is fixed to 500. The learning rate is set to 0.0003 initially, then decrease by factor of 88.6 every 5000 iterations. Dropout layer is employed to avoid overfitting.

4.2.4 Results and Discussion

I will present my experimental results for VQA task outlined in following: First, I will present the experimental results on each projection for compressive measurement using two methods. Then I will present all the experimental results together to discuss how different projections affect the experimental results.

method	All	Yes/No	Number	Other
VGG-net only [1]	28.13	64.01	0.42	3.77
deep LSTM [1]	50.39	78.41	34.68	30.03
LSTM + csm	47.95	78.34	32.45	29.10
SA	50.42	77.8	32.95	34.32
LSTM CNN	51.1	78.82	33.3	34.82
deep LSTM + norm VGG-net [1]	57.75	80.5	36.77	43.08

Table 4.2: Open-ended VQA Result for $\phi^T \phi x$ Dataset

Experimental results for $\phi^T \phi x$ is shown in Table 4.2. Three baselines are present in Table 4.2 that I will describe in details as following: “VGG-net only” denote as using only VGGNet feature to answer the question. “deep LSTM” refer to using only deep LSTM, which have 2 layers of hidden layers with 512 units in each layer, to answer the question without aid of images. “LSTM + csm” refer to using LSTM as question feature and compressive measurements as image feature. “deep LSTM+ norm VGGnet” refer to using deep LSTM as question feature and normalized VGGNet feedforward vector as image feature to perform inference. “SA” denote as stacked attention model as mentioned in section 3.2.3. We can see that LSTM CNN method experience 6.7% accuracy drop with the result using deep LSTM and normalized VGGnet feature reported in [1]. However,the LSTM CNN and SA methods are both outperform the baselines. In addition, the experimental result shows that “LSTM + csm” is worse than “deep LSTM” baseline, it may indicate that compreesive measurement itself is not a informative image feature, so one need to use the feature extractor like CNN to generate better image representation.

As mention in 3.1, I use block-based linear projection with the 32×32 block to

method	All	Yes/No	Number	Other
VGG-net only [1]	28.13	64.01	0.42	3.77
deep LSTM [1]	50.39	78.41	34.68	30.03
LSTM + csm	47.95	78.34	32.45	29.10
SA	52.06	78.38	32.73	37.21
LSTM CNN	52.98	79.5	33.03	38.15
deep LSTM + norm VGG-net [1]	57.75	80.5	36.77	43.08

Table 4.3: Open-ended VQA Result for Block-based $\phi^T\phi x$ Dataset

project the compressive measurements into pseudo images, the VQA results for this projection technique present in Table 4.3. Both LSTM CNN and stacked attention method outperform the baseline methods more than 1%, the LSTM CNN method have only 4.75% drop in term of accuracy and outperform the deep LSTM baseline for 2.6%.

Table 4.4, 4.5 shows experimental results for LSTM CNN method and stacked attention model, respectively. I also show the results from image recognition model utilizing ReconNet in these tables as comparisons of my methods. Experimental results from LSTM CNN method shown in Table 4.4, the performance for the block-based $\phi^T\phi x$ and RecoNet (no finetuned) model is quite the same, but the image recognition result have 13.02% difference as shown in Table 4.1.4. It may implies that image recognition results is not always positively correlated with the VQA results. Another example can validate this argument is that the block-based $\phi^T\phi x$ consistently outperform $\phi^T\phi x$ in VQA result while the image recognition result is nearly the same as shown in Table 4.1.4.

We can see that the LSTM CNN method consistently outperform the stacked attention model no matter which projection method is used. The reason for it may

projection	All	Yes/No	Number	Other
$\phi^T\phi x$	51.1	78.82	33.3	34.82
block-based $\phi^T\phi x$	52.98	79.5	33.03	38.15
ReconNet(no finetuned)	52.97	79.81	32.94	37.91
ReconNet(finetuned)	54.22	79.85	33.28	40.21
deep LSTM + norm VGG-net [1]	57.75	80.5	36.77	43.08
deep LSTM [1]	50.39	78.41	34.68	30.03

Table 4.4: Open-ended VQA Result for LSTM CNN Method

projection	All	Yes/No	Number	Other
$\phi^T\phi x$	50.42	77.8	32.95	34.32
block-based $\phi^T\phi x$	52.06	78.38	32.73	37.21
ReconNet(no finetuned)	52.14	78.4	32.56	37.40
ReconNet(finetuned)	53.15	78.38	32.71	39.40
deep LSTM + norm VGG-net [1]	57.75	80.5	36.77	43.08
deep LSTM [1]	50.39	78.41	34.68	30.03

Table 4.5: Open-ended VQA Result for Stacked Attention Model Method

be the image feature I extract from CNN is $1024 \times 7 \times 7$ creating coarse local regions representation, and thus fail to generate refine reasoning through attention mechanism to yield the better result than purely element-wise LSTM CNN method.

Regarding the time complexity for our models, the execution times for each models to answer a question for one image is present in Table 4.6 to compare efficiency of my proposed frameworks. We can see from the table that the method without reconstructing the compressive measurements is significantly faster than the method after reconstruction, it is nearly 3 order of magnitude difference in term of time complexity for my best model. The experimental results thus shows the advantage in time complexity using the method to inference without reconstruction.

method	time (s)
blocked-based + LSTM CNN	0.1592
blocked-based + SA	0.1612
$\phi^T \phi x$ + LSTM CNN	0.7185
$\phi^T \phi x$ + SA	0.7205
ReconNet + LSTM CNN	4.4995
ReconNet + SA	4.5015

Table 4.6: Execution Time for Each Model to Answer the Single Image with CPU

model	number of parameters
LSTM CNN	9193472
SA	14441514
ReconNet	22914

Table 4.7: Number of Parameters for Each Model

Table 4.7 shows the number of parameters for each model, where "SA" denote as stacked attention model as mentioned in section 3.2.3. The number of parameters for Stacked attention model is slightly larger than that of LSTM CNN method as shown in Table 4.7. By skipping the reconstruction process, Table 4.7 implies the amount of computational cost saves training the ReconNet given the number of parameters in ReconNet. In addition to the execution time experiment, experiments show that reconstruction process is relatively time consuming in testing phase, so we can avoid large amount of time cost by bypassing the step to reconstruct images from compressive measurements.

Chapter 5

CONCLUSION AND FUTURE WORK

In this thesis work, I propose an attempt to tackle VQA task using compressive sensing measurements. I also conduct a series of experiments to examine the feasibility of compressive VQA. Experimental results show that methods I propose outperform the language baselines. Moreover, our experimental results also achieve the similar performance of the result after reconstruction while bypassing the time consuming reconstruction process both in training phase and execution phase. Therefore, I think it is promising for future research to try to tackle this task. Moreover, I regard this work to explore the potential for compressive measurement to do complex task like VQA. The advantage of the reconstruction-free inference method in the resource constrained environment is obvious, I am excited to see more applications to come for complex inference task using compressive measurement.

5.1 Future Work

Regarding the future work, I think there is a few directions for future direction for this research. First, it is worthwhile to tackle compressive visual question answering to very low measurement rate for compressive measurements such as 0.01 and 0.001 to relax the requirement for storage space. It is worthwhile to investigate the utility of compressive measurement at very low measurement rate for complex computer vision task, and thereby will open up more possibilities in the resource constrained environment. Second, I think parameter hashing technique [8] may be promising for projection technique at image recognition task. As I encountered the overfitting issue to tackle image recognition task using compressive measurements, the hashing technique can be possible solution to overcome this issue since it significantly reduce the number of parameters. Also, it may be useful to employ it directly to the VQA task, as [28] use the method to predict the parameter in network.

REFERENCES

[1] Antol, S., A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick and D. Parikh, “VQA: Visual Question Answering”, in “International Conference on Computer Vision (ICCV)”, (2015).

[2] Ben-younes, H., R. Cadène, M. Cord and N. Thome, “MUTAN: multimodal tucker fusion for visual question answering”, CoRR **abs/1705.06676**, URL `http://arxiv.org/abs/1705.06676` (2017).

[3] BVLC, “Models accuracy on imagenet 2012 val”, URL `https://github.com/BVLC/caffe/wiki/Models-accuracy-on-ImageNet-2012-val` (2015).

[4] C, E., J. Romberg and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements”, Comm. Pure Appl. Math. **59**, 1207–1223 (2006).

[5] Calderbank, R., S. Jafarpour and R. Schapire, “Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain”, Tech. rep. (2009).

[6] Candes, E. J. and T. Tao, “Decoding by linear programming”, IEEE Trans. Inf. Theor. **51**, 12, 4203–4215, URL `http://dx.doi.org/10.1109/TIT.2005.858979` (2005).

[7] Candes, E. J. and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?”, IEEE Trans. Inf. Theor. **52**, 12, 5406–5425, URL `http://dx.doi.org/10.1109/TIT.2006.885507` (2006).

[8] Chen, W., J. Wilson, S. Tyree, K. Weinberger and Y. Chen, “Compressing neural networks with the hashing trick”, in “International Conference on Machine Learning”, pp. 2285–2294 (2015).

[9] Chung, J., Ç. Gülçehre, K. Cho and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling”, CoRR **abs/1412.3555**, URL `http://arxiv.org/abs/1412.3555` (2014).

[10] Dalal, N. and B. Triggs, “Histograms of oriented gradients for human detection”, in “Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 1 - Volume 01”, CVPR ’05, pp. 886–893 (IEEE Computer Society, Washington, DC, USA, 2005), URL `http://dx.doi.org/10.1109/CVPR.2005.177`.

[11] Donahue, J., L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description”, in “Proceedings of the IEEE conference on computer vision and pattern recognition”, pp. 2625–2634 (2015).

[12] Fukui, A., D. H. Park, D. Yang, A. Rohrbach, T. Darrell and M. Rohrbach, "Multimodal compact bilinear pooling for visual question answering and visual grounding", CoRR **abs/1606.01847**, URL `http://arxiv.org/abs/1606.01847` (2016).

[13] He, K., X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 770–778 (2016).

[14] Hennings-Yeomans, P. H., B. V. K. V. Kumar and M. Savvides, "Palmprint classification using multiple advanced correlation filters and palm-specific segmentation", IEEE Trans. Information Forensics and Security **2**, 3-2, 613–622, URL `https://doi.org/10.1109/TIFS.2007.902039` (2007).

[15] Hochreiter, S. and J. Schmidhuber, "Long short-term memory", Neural Comput. **9**, 8, 1735–1780, URL `http://dx.doi.org/10.1162/neco.1997.9.8.1735` (1997).

[16] Huang, G., H. Jiang, K. Matthews and P. Wilford, "Lensless imaging by compressive sensing", in "Image Processing (ICIP), 2013 20th IEEE International Conference on", pp. 2101–2105 (IEEE, 2013).

[17] Kim, J., K. W. On, J. Kim, J. Ha and B. Zhang, "Hadamard product for low-rank bilinear pooling", CoRR **abs/1610.04325**, URL `http://arxiv.org/abs/1610.04325` (2016).

[18] Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization", CoRR **abs/1412.6980**, URL `http://arxiv.org/abs/1412.6980` (2014).

[19] Kulkarni, K., S. Lohit, P. Turaga, R. Kerviche and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 449–458 (2016).

[20] Kulkarni, K. and P. Turaga, "Reconstruction-free action inference from compressive imagers", IEEE Transactions on Pattern Analysis and Machine Intelligence **38**, 4, 772–784 (2016).

[21] Le, Q. and T. Mikolov, "Distributed representations of sentences and documents", in "Proceedings of the 31st International Conference on Machine Learning (ICML-14)", pp. 1188–1196 (2014).

[22] Lohit, S., K. Kulkarni and P. Turaga, *Direct inference on compressive measurements using convolutional neural networks*, vol. 2016-August, pp. 1913–1917 (IEEE Computer Society, United States, 2016).

[23] Lowe, D. G., "Distinctive image features from scale-invariant keypoints", Int. J. Comput. Vision **60**, 2, 91–110, URL `http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94` (2004).

[24] Malinowski, M. and M. Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input", in "Advances in Neural Information Processing Systems", pp. 1682–1690 (2014).

[25] Malinowski, M., M. Rohrbach and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images", in "Proceedings of the IEEE international conference on computer vision", pp. 1–9 (2015).

[26] Marwah, K., G. Wetzstein, Y. Bando and R. Raskar, "Compressive Light Field Photography using Overcomplete Dictionaries and Optimized Projections", ACM Trans. Graph. (Proc. SIGGRAPH) **32**, 4, 1–11 (2013).

[27] Mikolov, T., K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space", CoRR **abs/1301.3781**, URL `http://arxiv.org/abs/1301.3781` (2013).

[28] Noh, H., P. Hongsuck Seo and B. Han, "Image question answering using convolutional neural network with dynamic parameter prediction", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 30–38 (2016).

[29] otoro, "Lstm", URL `http://blog.otoro.net/2015/05/14/long-short-term-memory/` (2015).

[30] Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge", International Journal of Computer Vision **115**, 3, 211–252 (2015).

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

[32] Socher, R., A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank", in "Proceedings of the conference on empirical methods in natural language processing (EMNLP)", vol. 1631, p. 1642 (Citeseer, 2013).

[33] Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", Journal of Machine Learning Research **15**, 1929–1958, URL `http://jmlr.org/papers/v15/srivastava14a.html` (2014).

[34] Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 1–9 (2015).

[35] Takhar, D., J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly and R. G. Baraniuk, "A new compressive imaging camera architecture using optical-domain compression", Proc. SPIE **6065**, 606509–606509–10, URL `http://dx.doi.org/10.1117/12.659602` (2006).

[36] Vinyals, O., A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator", in "Proceedings of the IEEE conference on computer vision and pattern recognition", pp. 3156–3164 (2015).

[37] Wright, J., A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, "Robust face recognition via sparse representation", IEEE Trans. Pattern Anal. Mach. Intell. **31**, 2, 210–227 (2009).

[38] Wu, Q., P. Wang, C. Shen, A. Dick and A. van den Hengel, "Ask me anything: Free-form visual question answering based on knowledge from external sources", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 4622–4630 (2016).

[39] Xu, H. and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering", in "European Conference on Computer Vision", pp. 451–466 (Springer, 2016).

[40] Xu, Z., Y. Yang and A. G. Hauptmann, "A discriminative cnn video representation for event detection", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 1798–1807 (2015).

[41] Yang, Z., X. He, J. Gao, L. Deng and A. Smola, "Stacked attention networks for image question answering", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 21–29 (2016).

[42] Zhou, B., Y. Tian, S. Sukhbaatar, A. Szlam and R. Fergus, "Simple baseline for visual question answering", CoRR **abs/1512.02167**, URL `http://arxiv.org/abs/1512.02167` (2015).