VISUAL QEUSTION ANSWERING WITH DEEP LEARNING

TU FENGZHI 1155058610

ZHENG ZHIXUAN 1155058600

Supervised by Prof. LYU Rung Tsong Michael

TABLE OF CONTENTS

Table of Contents

TABLE OF CONTENTS

Introduction

1. Overview

In March 2016, a computer program named AlphaGo (AlphaGo) which is developed by Google DeepMind beat Lee Sedol in playing the board game Go and it was the first time a computer program beat human 9-dan professional in game Go because Go is one of most difficult games that have a large search space. AlphaGo uses algorithms in deep learning to “learn” from the previous game record both from top human players and computer to computer records. The success of AlphaGo makes deep learning become a popular topic in computer science and attracts more and more researcher dive into the deep learning area.

Recent years, more and more researchers in both university and industry devote them into the field of deep learning and the number of published articles about deep learning algorithms have exploded. Both the universities and companies are pushing forward to speed up the developments and research of this area since the deep learning can solve a lot of problems better than classic machine learning algorithm. Deep Neural

INTRODUCTION

Network are widely used nowadays in our daily life such as recommendation systems, automatic translation, auto-drive car, etc.

Among all the popular deep learning problems, we are interested in the problem that joining image and text based on the classic question-answering problem. This problem combines computer vision problems such as image caption and object detection and some text process problems such as natural language processing.

The objective of our project is to use the deep learning method to solve Visual Question Answering problem. Visual Question Answering is a new dataset which include both open-end questions and multiple choice questions about images, it allows us to rise a question to an image, and the deep learning model will answer the question based on the image. Visual Question Answering can be regarded as a promotion of classic text based Question Answering problem. From the previous works, we can find that the most common method to solve this Visual Question Answering problem is to use Convolutional Neural Network to extract visual features from images first. Then using the Long Short-Term Memory (a kind of Recurrent Neural Network) to process the embedded natural language question vectors.

2. Statement of Purpose

The task of our project is to come up with a new model about Visual Question Answering to enhance the accuracy of previous models since there are many limitations in previous models.

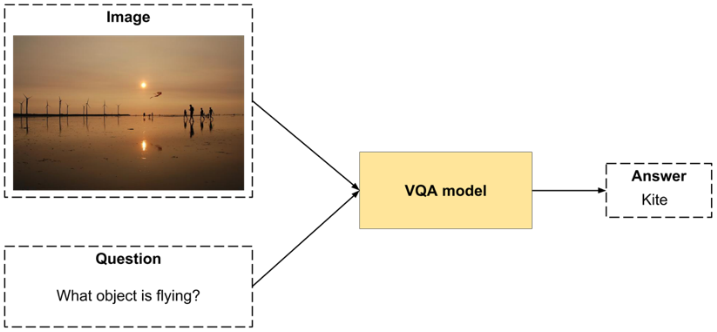


Figure 1 Example of Visual Question Answering

Based on the current work in Visual Question Answering area, the basic objectives of our project are:

- Study and explore the methods and theories used in text-based Question Answering problem.

- Study and explore the methods of extracting the features from images.

- Study and explore the methods of word embedding and find the best one that most suitable to our project.

- Build a model which can answer questions based on the input image.

- Come up with some models that maybe can enhance the accuracy of the Visual Question Answering and verify their feasibility.

The advance objectives of our project:

- Try different architectures and parameters to increase model's accuracy.

- Implement a model that have a high accuracy in number-relate and logic-relate questions.

- Extend the language of the model to Chinese, since the main language we use now is English and there exists huge difference semantics between the English word and Chinese word and we should use different way to process the Chinese text.

Study of Technologies

1. TensorFlow

1.1 Introduction to TensorFlow

Figure 2 Logo of TensorFlow (TensorFlow)

In our project, we use a famous deep learning platform TensorFlow (TensorFlow), which support various of APIs of deep learning algorithms such as CNN (Convolutional Neural Network), RNN (Recurrent Neural Network) and LSTM (Long Short-Term Memory) which are the most popular deep learning algorithm models in the fields of image processing, speech process and natural language processing. TensorFlow is an open source library for machine learning which is developed by Google brain team and it is based on DistBelief and it is named TensorFlow because of its methodology.

There are two basic but important objects in TensorFlow, one is the Tensor, another is the data flow graph. The basic principle of TensorFlow is to do operation using Tensor. Tensor is a N-dimension array and Flow means the calculation is based on the data flow graph. And the process of "TensorFlow" is the flow of tensor from a node of date flow graph to another node of graph. TensorFlow is a system that transfers the complex data structures to the artificial neural network and analysis and process the input data.

1.2 Tensor

Tensor is a linear form function that can be used to represent the relationship of geometric vectors, scalars and other tensors. The relationship can be inner product, outer product and linear mapping and Cartesian product.

The following figure is an example of a tensor. As we can see, the coordinate of a tensor is in a N-Dimension space and contains n^r vectors.

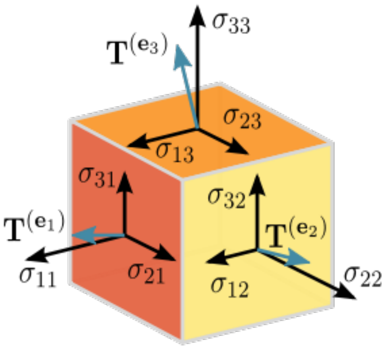


Figure 3 Tensor (htt2)

1.3 Data Flow Graph

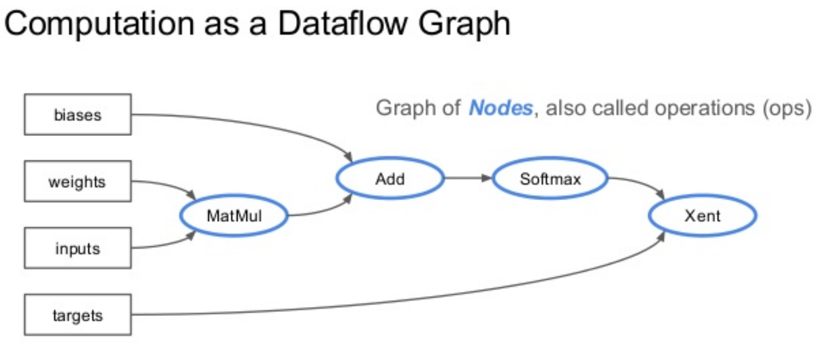

Computation as a Dataflow Graph
Graph of *Nodes*, also called operations (ops)
biases
weights
inputs
targets
MatMul
Add
Softmax
Xent

Figure 4 Data Flow Graph (htt3)

The data flow graph is the computation model using directed graph, nodes and edges. In this graph, nodes not only represent the mathematics operations, but also represent the operation that read/write the global variable, endpoints for feeding data and endpoints for pushing out result. Edges represent the communication between nodes and indicates the relationship between nodes by transiting tensor, multidimensional data arrays, between nodes. When running the TensorFlow program, the nodes will be assigned to computational devices like CPU and GPU, and each node can run in parallel asynchronously and execute instructions when data of all its incoming edges is really.

1.4 Single-Device and Multi-Device

TensorFlow not only supports a single machine mode but also supports a distributed system.

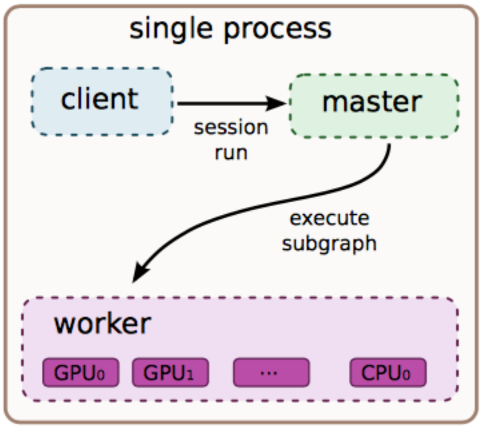


Figure 5 Single Machine (htt4)

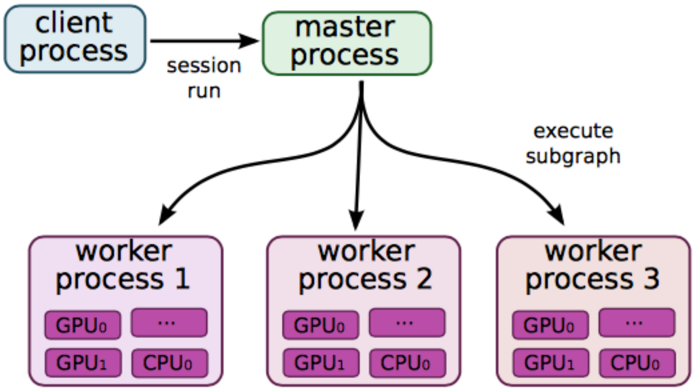


Figure 6 Distributed System (htt4)

Thus, when the work load is too heavy for single machine, we can use a distributed system to reduce the runtime of the operations.

2. Natural Language Processing

2.1 Traditional Methods

The ability for computers to communicating with human in natural language is what human are keeping on chasing. To let computers to be able to communicate with human in natural language, computers need to be able to understand the meaning of natural language and able to express the ideas in natural language. The first one is called Natural Language Understanding and the second one is called Natural Language Generating. Therefore, Natural Language Processing basically contains two parts which are Natural Language Understanding and Natural Language Generation. Historically, scientists did more research on Natural Language Understand and less on Natural Language Generation. However, situation is changed now.

Achieving the communication between human and computers is natural language is very difficult because of:

- Difficulty to determine the boundary of words.

In speaking, there is not boundary between word and word. Therefore, computer need find out the best combination of words based on the context.

- Ambiguity of words.

One word may have different meanings. The computer need to find out the most suitable meaning based on the context.

- Ambiguity of grammar.

Sentences may have different parsing trees because of ambiguous grammar. Therefore, computer need to determine which parse tree is the best based on the context.

When neural network is not introduced in this field, conceptual dependency theory is widely used in Natural Language Processing.

Conceptual dependency theory is first introduced in 1969. In this model, two sentences with the same meaning will have the same representation even though the words in these two sentences are not the same. Four basic representational tokens are used in conceptual dependency theory model which are real word objects with some attributes, real world actions with some attributes, times, locations.

Conceptual dependency representing
“John ate the egg”(Schank and Rieger 1974).

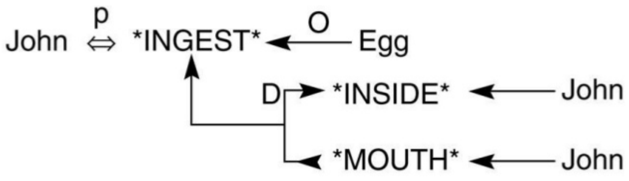


Figure 7 Example of Conceptual Dependency Theory

After machine learning is introduced, A revolution occurred in this field. At the beginning, scientist uses machine learning to produce something like decision thing to represent the natural language which is like previous traditional approach. Later, scientists began to focus on develop representation algorithm, like word embedding, on statistical model.

2.2 Word Embedding

Word embedding is a set of techniques that maps words or phrases to vectors of real number. We will use word2vec model developed by Google

Inc. (Mikolov, Sutskever and Chen) to process the natural language text of train questions and answers.

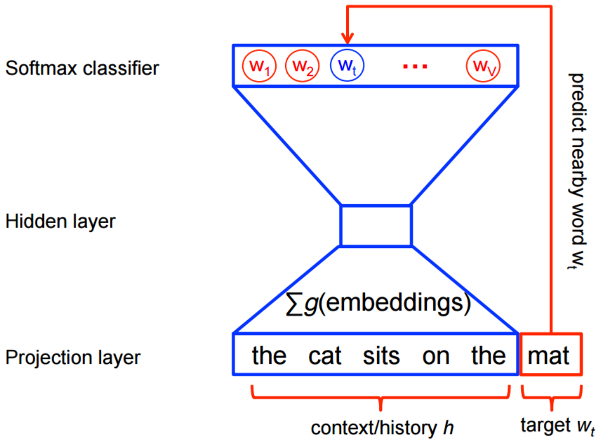


Figure 8 Model of Word Embedding

Word2vec [3] learning how words are used in input text by using a shallow neural network. Word2vec output a matrix and each column of this matrix represents a word in input text and provides a numerical description on how the word is used in input text. If the input train text is large enough, the vector that represent two words with similar meaning will have small vector distance. For instance, the word “woman” and word “lady” may have a small vector distance. Most application of word2vec judge the similarity of two words using cosine similarity. Empirically, Word2vec has very nice performance when the input text is very large, consistent and without ambiguity.

To map the words or phrases to numerical vector, A technique called “skip-gram with negative sample” is used by Word2vec. This method can be roughly divided into 5 steps.

First, take a word in the input text as target and a few words that are close to this target as context.

Second, represent each word in the input text by a random numerical vector. After this step, we can get the numerical vector of the target and context.

Third, put the vector of target and context together and shorten the distance of these vectors.

Fourth, pick up the words randomly from the input text outside the context, and enlarge the distance between the vectors of these words and the vector of the target. By doing this, we can make our target be further away from the words which is rarely used in context.

As keep on applying this process on the targets and their contexts, the vectors of words which are always used together will be pulling closer and closer which the vectors of words which are rarely used together will be pushing further and further.

2. Image Processing

In image science, image processing represents a various of technologies that used to analysis and process the images such that the images meet the visual or mentality request or some other technical requirements. It is an application of signal processing in the domain of images. Nowadays, most of images are stored in digital format, thus image processing most refers to the digital image processing. Besides, some methods based on the optical theories still holds an importance position.

Image processing is sub-class of signal processing and has a deep connection with computer science and artificial intelligence. The traditional ways in process the one-dimension signals can be used in image processing such as quantization and noise elimination. But the images are still two-dimension signals which have their own special methods.

STUDY OF TECHNOLOGIES

2.1 Feature Detection

When the neural network is not widely used, scientists use feature detection to gather information from images, deciding whether every single pixel belongs to an image feature. Therefore, the result of feature detection is usually a collection of sets of pixels and the sets contains isolated points, continuous lines or continuous regions.

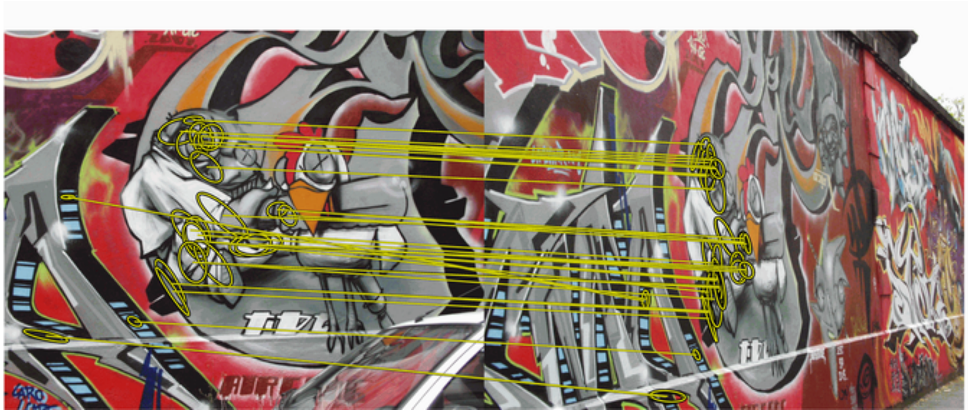


Figure 9 Feature Detection

Nowadays, there is not universal and accurate definition of features. Features are usually determined by the problems or the applications. Features can be the interest parts in a digital image and they are the starting point of many computational images analysis algorithm. Whether an algorithm will be successful or not depends on which features this algorithm uses. Therefore, features must be repeatable which means the features extracted on different images in same scenario should always be the same.

Feature detection is a very fundamental algorithm in image processing, which means it is always the first processing on the images. Feature detection checks every single pixel and determine whether this pixel represents a feature. If feature detection is a part of a larger algorithm, then feature detection usually only check the feature regions of images. As the preprocessing of feature detection, the input images are usually smoothed by using Gaussian kernel.

Many different image processing algorithms use feature detection as its starting point. Therefore, many different detection algorithms are developed and the types of features, the computational complexity and repeatability are varied from each other.

2.1.1 Edge detection

Edge detection is a fundamental problem in computer vision and image processing and aims to find the pixel at which image brightness changes a lot in the images. The sharp change in pixels usually refers to important change of feature. These changes include discontinuities in depth, discontinuities in surface orientation, changes in material properties and variations in scene illumination.

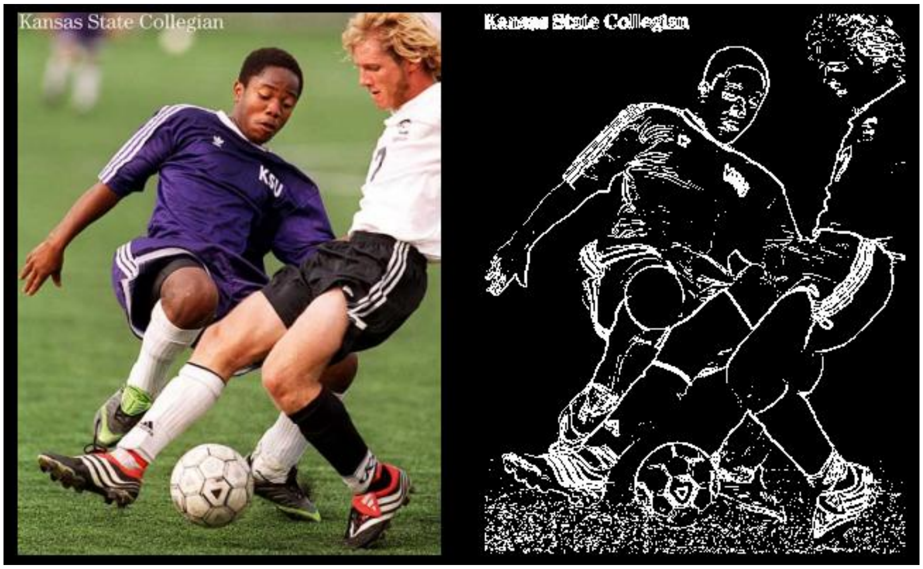


Figure 10 Sample of Edge Detection

After applying feature detection, the size of data is significantly reduced, some information which is considered as not important is removed and the important structural features of image are remained.

There are many different methods to perform edge detection, most of them can be categorize into two part, search-based and zero-based. The search-based method detects edges by computing the gradient of images and find out the local directional maximum of gradient. The zero-crossing based methods search for the zero-crossing of the Laplacian of images.

2.2 The Shortcoming of Traditional Methods

Although some traditional method in image processing can achieve a better performance, there are still some shortcomings in these ways. As the rapidly development of the computation speed of computers, deep learning algorithms are wildly used in image processing, such as CNN (Convolutional Neural Network). The results of experiments show that deep learning can perform better than the traditional methods in image processing.

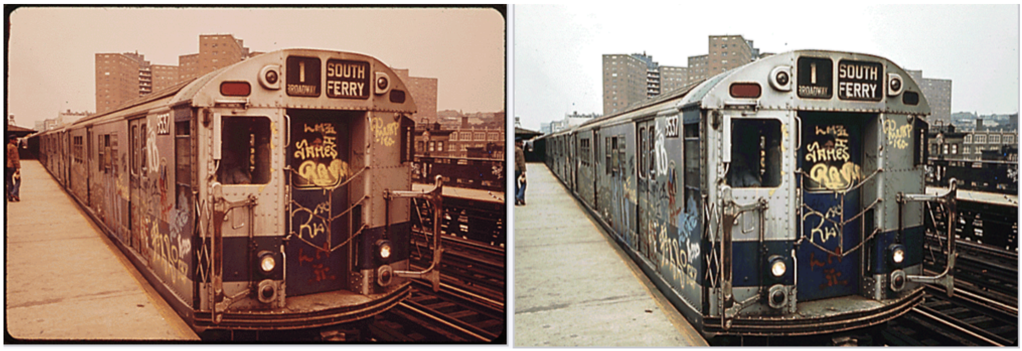


Figure 11 Enhance of Image Using Optical Theory (htt5)

In our project, we should extract the features of an image and feed the features into the neural network. According to the previous experiments, CNN (Convolutional Neural Network) has an excellent performance in extracting the features of images. And we will introduce the Convolutional Neural Network in the next chapter.

2.3 Convolutional Neural Network

In machine learning, convolutional neural network, inspired by animal vision system, is a type of feed-forward neural network that is generally used for image processing because the architecture of convolutional neural network is very suitable for this kind of tasks. For example, the ImageNet (Krizhevsky, Sutskever and Hinton) model has an excellent performance on image classification.

A very important difference between standard artificial neural network and convolutional neural network is that the neurons in the layers of convolutional neural networks is arranged into three dimensions instead of two dimensions in artificial neural network. In addition, differing from the neurons in artificial neural network, the neurons in convolutional neural network only connects to a small region of neurons instead of all neurons in previous layer. (O'Shea and Nash)

2.3.1 Overall architecture

Convolutional neural networks are composed of three kinds of layers, which are fully-connected layers, pooling layers and convolutional layers.

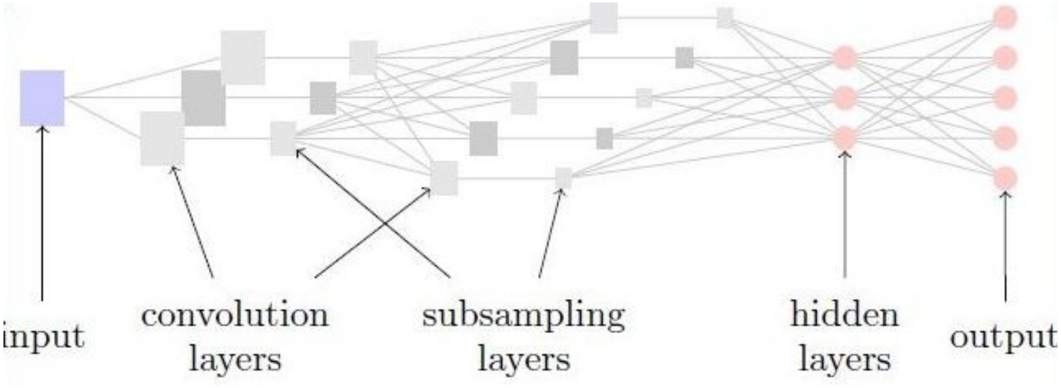


Figure 12 A Simple Convolutional Neural Network with Five Layer

As an example, the architecture shown above can be decomposed into several parts.

- Same as traditional artificial neural network, the input layer of Convolutional Neural Network holds the input data.

- Every neuron in the convolutional layers generate its own output by calculating the scalar sum of the products of weights and inputs from the neuron which it connected.

- The pooling layers apply down-sampling to reduce the dimension of input from previous layer. Therefore, the number of neurons, such as, neurons in preceding convolutional layer, and the number of parameters, such as, weights hold by neurons in preceding convolution layer, will be reduced. Therefore, the total model complexity is reduced due to pooling layers.
- The full-layers have the same functionality as in artificial neural network which is calculating the class scores from the activations, which will be used for classification.

According to this small sample, convolutional neural networks can transform the input using several technologies, like convolution, down-sampling layer by layer to produce class score for classification or other purposes.

2.3.2 Convolution Layer

Convolution layer is the most important layer of convolutional neural network and the parameter in this kind of layer is the learnable kernels.

These kernels are commonly very small in terms of dimension comparing with input. In convolutional layer, when the input is ready, the input convolves with the small kernel to create a 2D activation map.

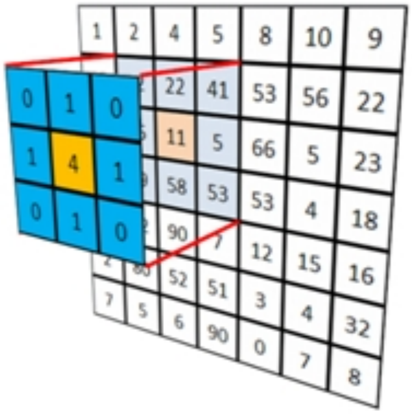


Figure 13 Gaussian Kernel

As we scan the input, the convolution result will be calculated using the kernels for each portion of input. Kernels will learn whether a specific feature has been sawn at a certain position of the input from the network and these processes are usually called activations. Each kernel has its own activation map based on the input, and these activation maps will be stack together to form a 3D-array as output of convolutional layers.

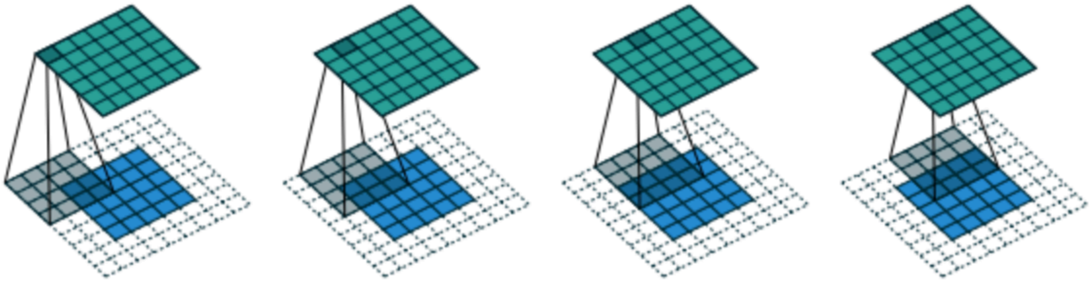


Figure 14 Convolution Process of An Image

As we all known, artificial neural networks which are been trained for image processing are always too large due to fully-connected manner of standard artificial neural network, therefore this kind of artificial neural networks cannot be efficient enough for training. To solve this problem, neurons in convolutional layers only connect to a small region of neurons in previous layer. The size of this region is called receptive field size.

For instance, if the input image is RBG image with size 100 * 100 * 3 and receptive field size is set to 5 * 5, each neuron in convolution layers should only maintain 75 weights comparing with 30000 weights for neurons in artificial neural networks.

Convolution layers also can optimize their output to reduce the model complexity significantly. There are three basic hyper-parameters for optimization which are the depth, the strike, and the size of zero-padding.

Depth is the third dimension of the output and it controls the number of neurons connect to the same region of input. Reducing the depth can significantly reduce the neurons in convolutional layers.

Stride controls the height and weight of output. If the stride is set to 1, the receptive fields will be heavily overlapped and the output will be very large. So increasing the stride can reduce the overlapping of receptive fields and the size of output.

Zero-padding is the process that padding on the border of the input. Therefore, the number of zeros that zero-padding applies can further control the size of output.

It is important to understand these three hyper-parameters and we can calculate the size of output based on the formula $(N - F + 2 * Z) / (S + 1)$ where N is the size of input, F is receptive field size, Z is the size of zeros padding on the border and S is the stride. If the result is not an integer, the stride is set incorrectly because the neurons cannot fit in this configuration on the input.

2.3.3 Pooling layers

Pool layers is used to reduce the size of data and therefore reduce the parameters (weights) of each neuron and model complexity. The pool layers take activation maps as input and reduce their size by using Max function. Since the pool layers use Max function, this kind of pool layers are usually referred to max-pooling layers. For instance, if a kernel with size 2 * 2 is applied with stride 2 to the input, the height and width will all be reduced to the half of their counterpart of input. Therefore, the size of output is reduced to only 25% of the size of input with the depth unchanged.

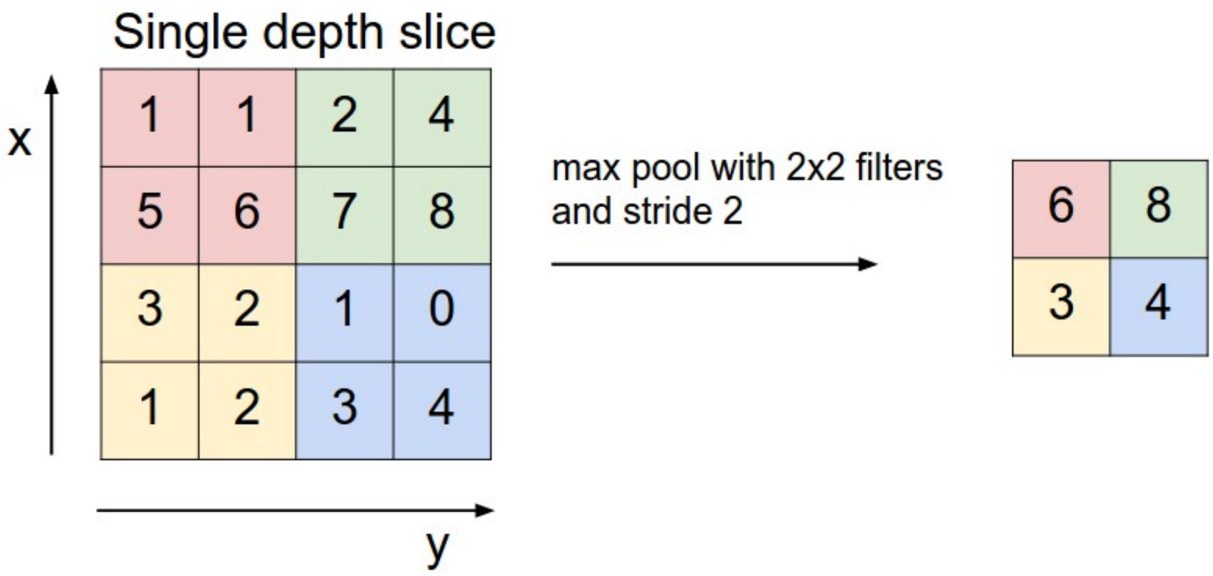


Figure 15 Pooling Layer

As the pooling layers remove information from input, there are only two ways to implement max-pooling. The first one is already described in example above. The kernel is set to 2 * 2 and stride is set to 2. Another one is called overlapping-pooling, where the kernel is set to 3 * 3 and stride is set to 2. In general, the size of kernel cannot exceed 3, because it will remove too much information.

2.3.4 Fully-connected Layers

each neuron in fully-connected layer connects to all the neurons in previous layer and preceding layer. The functionality of fully-connected layers in convolutional neural networks is the same as theirs in artificial neural network.

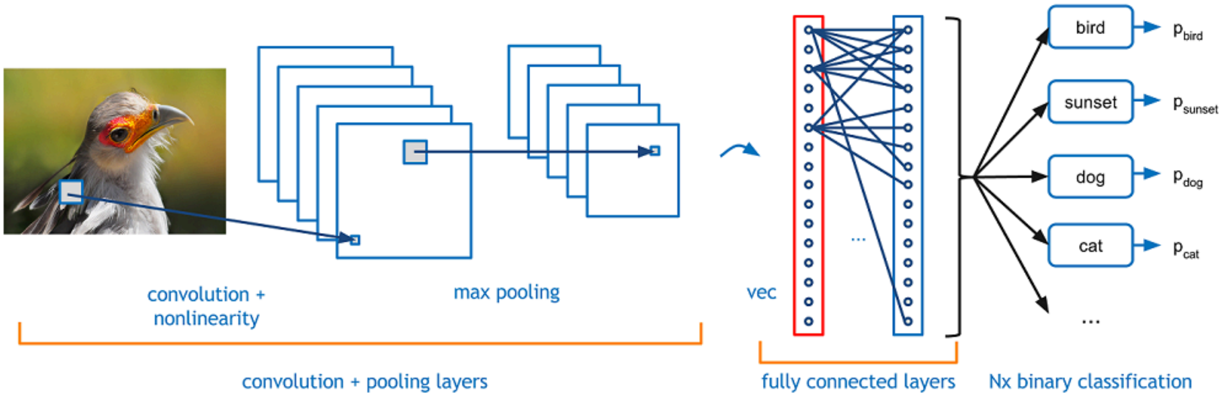


Figure 16 Fully Connected Layers In CNN

2.3.5 Softmax Layer (Output Layer)

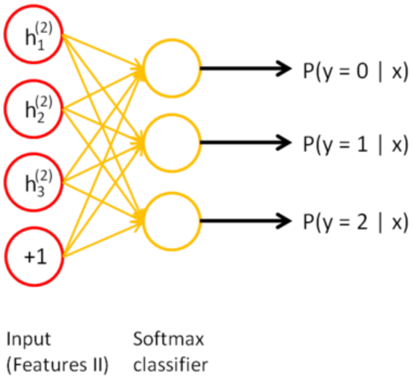


Figure 17 Softmax Layers

Softmax function is an extension of logistic function. The logistic regression is to solve the binary problem which are labeled {0, 1} .

The hypothesis function of logistic regression is:

$$h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)}$$

The cost function of logistic regression is:

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)} \log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right)\Big]$$

The train process is to minimize the cost function $J(\theta)$. After we obtain the optimal parameter, we can use the logistic function to predict the result of input data. But the logistic function can only be used in binary problem, when the labels of data is larger than two, we need to use k logistic classifiers and it is fussy. In this case, we can use the softmax function to solve the problem.

The hypothesis function of softmax function is:

$$h_\theta\left(x^{(i)}\right) = \begin{bmatrix} p\left(y^{(i)} = 1 \middle| x^{(i)}; \theta\right) \\ p\left(y^{(i)} = 2 \middle| x^{(i)}; \theta\right) \\ \vdots \\ p\left(y^{(i)} = k \middle| x^{(i)}; \theta\right) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} \theta_1^T x^{(i)} \\ \theta_2^T x^{(i)} \\ \vdots \\ \theta_k^T x^{(i)} \end{bmatrix}$$

The cost function of logistic regression is:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)} = j\} log \frac{\theta_j^T x^{(i)}}{\sum_{l=1}^{k} \theta_l^T x^{(i)}}\right]$$

The train process of the softmax function is like to the logistic function, and finally we can use the function to predict the result:

$$P(y = j|x) = \frac{e^{x^T w_j}}{\sum_{k=1}^{K} e^{x^T w_k}}$$

3. Long Short-Term Memory (Recurrent Neural Network)

3.1 Recurrent Neural Network

When human beings read articles, they comprehend the whole article by understanding each word based on the meaning of previous words since a single word doesn't have any meaning.

In traditional Neural Network, there have no connection between the nodes in the same layer. That means traditional Neural Network cannot preserve the relation between the original input data and this is a major defect of traditional full connected neural network.

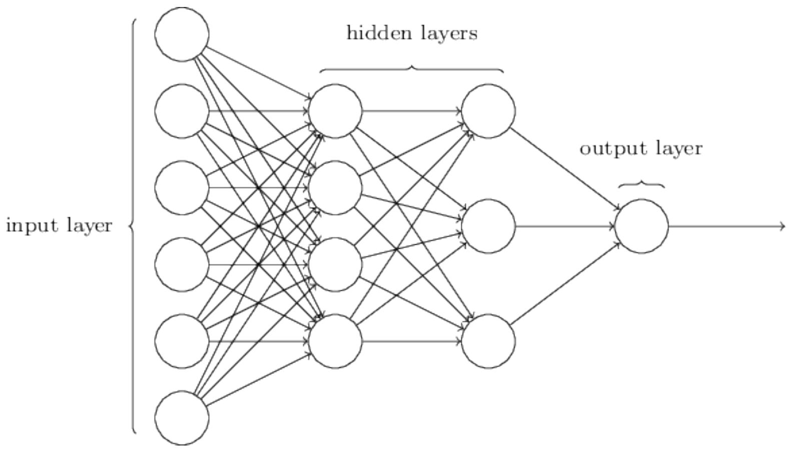


Figure 18 Structure of Traditional Neural Network

Recurrent Neural Network is designed to preserve the relationship between the node and its previous node. There are connections between nodes in the same layer of Recurrent Neural Network and this property makes Recurrent Neural Network becomes applicable to solve the problems like handwriting and natural language processing.

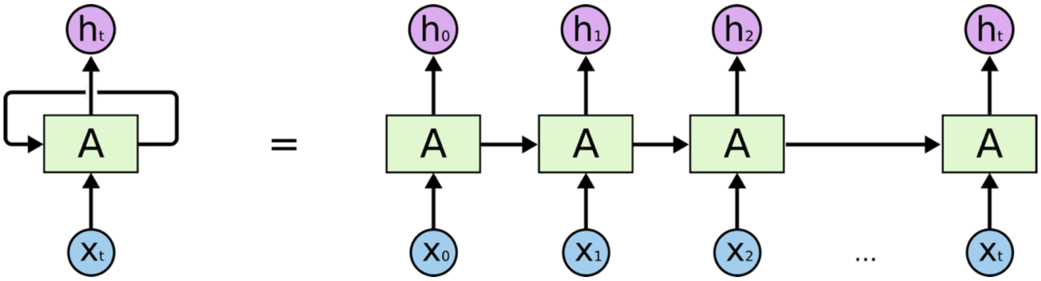


Figure 19 A Single Layer of Recurrent Neural Network (Understanding LSTM Networks)

In the above figure, A is a single layer in a Recurrent Neural Network while it has input parameter X and output value h and the value of information can pass from the previous nodes to next nodes in the same layer, that is from Xt to Xt-1.

Recurrent Neural Network are widely used in recent years since it has been proved to have an excellent performance in solving a various of problems like speech recognition, language translation and so on.

3.2 Long Short-Term Memory

Although Recurrent Neural Network have a good performance in some area, but it still has some shortcomings. Sometimes, we only need to loop back to just a few steps, for example, when we want to predict the next word the user may input, we may only look up the previous several words that has been inputted. In the above case, Recurrent Neural Network can have an excellent performance. But if we want to solve the Question Answering problem, looking up only several previous words doesn't work anymore. We need the content from further back. In practice, Recurrent Neural Network doesn't seem to solve this "Long-Term dependencies" problem well.

Long Short-Term Memory is explicitly designed to solve the “Long-Term dependencies” problem (Understanding LSTM Networks) and it is kind of Recurrent Neural Network which proposed in 1997.

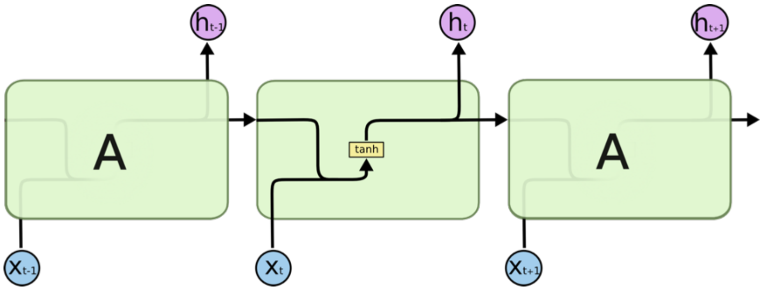


Figure 20 The Structure of a Single Node of Standard Recurrent Neural Network (Understanding LSTM Networks)

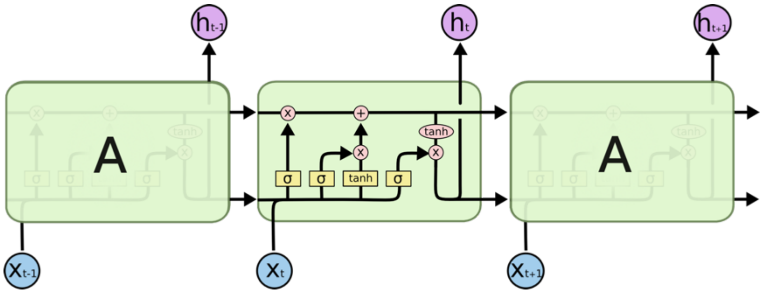


Figure 21 The Structure of a Single Layer of Long Short-Term Memory (Understanding LSTM Networks)

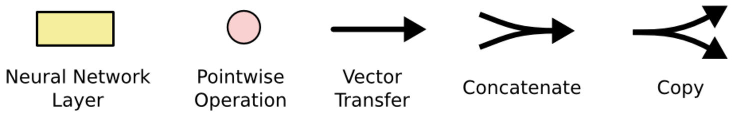


Figure 22 The Meaning of Symbols (Understanding LSTM Networks)

Long Short-Term Memory have the same chain structure like Recurrent Neural Network, but they have a different repeat module in a single node. In General, a standard Recurrent Neural network only has single neural network layer while Long Short-Term Memory has four.

Methodology

In general, an image can be described as a row * column * 3-dimension matrix which in a high dimension and text can be represented as a vector which in a low dimension. What we need to is to reduce the dimension of images or increase the dimension of text so that these two items can be combined and feed to neural network.

In the first, we consult to the structure of Question-Answering model based on text, then extent the text-base to image-base structure.

1. Text-Based QA Model

1.1 Word Embedding

First, we perform the word embedding to convert word strings into vectors. The method to perform the word embedding is call Skip-gram (Mikolov, Sutskever and Chen). First, we will traverse all the words in the training set and get a unique word dictionary vector. Second, we can replace words in all the sentence in the train set with the index of the word in the dictionary vector. After converting sentence in the train set into vectors, we can use Long Short-Term Memory (Understanding LSTM Networks) (LSTM) to capture more information about the sentence.

1.2 End-to-End Memory Network (Sukhbaatar, Szlam and Weston)

After converting the words to vectors, end-to-end memory network is used to solve question answering problem, this network is a kind of recurrent neural network where the recurrence reads from a large data before it produces the result.

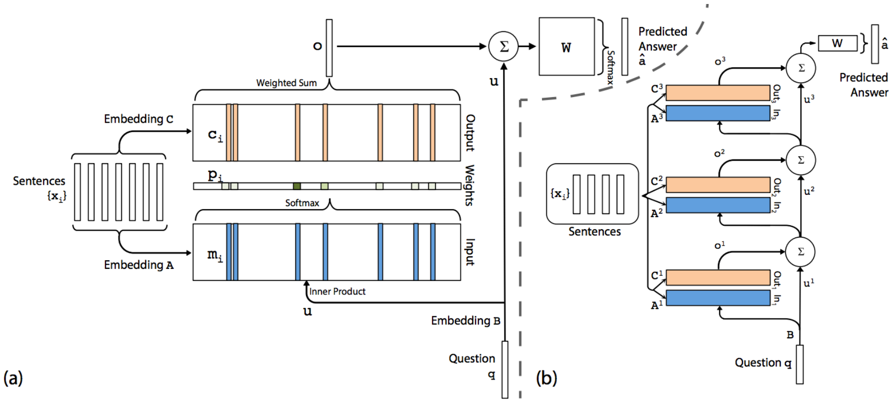


Figure 23 (a) A Single Layer Version (b) A Three Layer Version (Sukhbaatar, Szlam and Weston)

End-To-End Memory Network can divide into the following parts:

- Input representation:

In this stage, the model maintains several embedding matrices with the same dimension. We use a certain embedding matrix to convert the input vectors to memory vectors and convert the query to an internal state. After that, we compute the match, which is a possibility, of the memory vectors and the internal state by using SoftMax function on the dot product of these vectors.

- Output representation:

In this stage, we use the output vectors, which are calculated for each input vector by using another embedding matrix, to compute to the response vector of the memory. The response vector is simply the sum of the products of each output vector and its corresponding possibility which is computed in last stage.

- Find out the prediction:

In the single layer case, the prediction of the network is to sum all response vector and the internal state and then pass this sum through the weight matrix. Finally, we can use Softmax function to get the result.

2. From Text to Image

2.1 Extract Image Features.

In our application, we used the VGG model (K. Simonyan) to extract the feature from training set.

VGG model contains a stack of convolutional layers with kernel size of 3 * 3. The convolutional stride is 1 and the zero-padding size is also 1. For pooling layers, VGG model contains five max-pooling layers with 2 * 2 kernel and stride equal to 2. Following the convolutional layers, there are three Fully-connected layers. The first and second fully-connected layer contains 4096 neurons each while the third layer has only 1000 neurons because this layer is used to perform ILSVRC classification. Finally, we add a soft-max layer as final layer.

Instead of training our own VGG model, we use the VGG provided Keras (VGG16 model for Keras). After deploy the model in our application, we extract the feature from the images in MSCOCO dataset. In our program for extracting feature, we set parameter batch size to 10, which means the program will process 10 images in one turn. In each turn, we extract feature of these ten images, which are vectors, using the VGG model. After the features are produced by the VGG model, we store the features into a file as the input of LSTM.

2.2 Feed the Image Features to LSTM

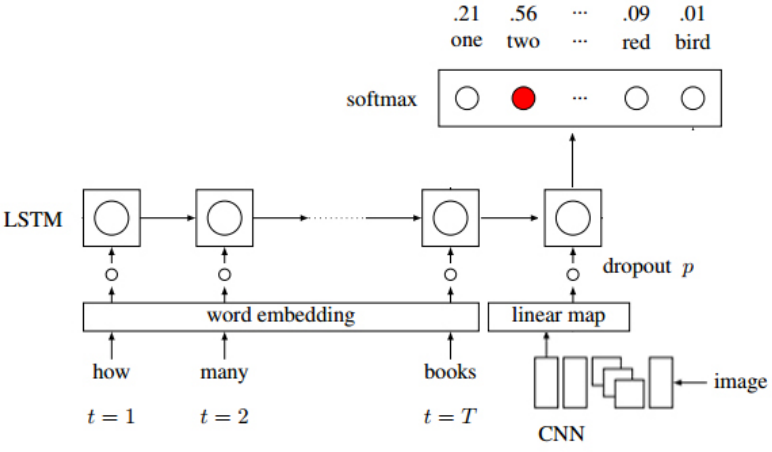


Figure 24 Whole Model of Visual Question Answering

After using Convolutional Neural Network to extract the features from the image train set, we feed these features with the text vectors to the Long Short-Term Memory to train the model.

3. Hypothetical Model

To improve the accuracy of number-relative question, we decide to use some ways to process the image such that we can count all objects in an image and feed the result into the neural network as image features.

3.1 Object Counting Model

To improve the performance of our model for visual question answering, we build a sub-model for objection detection which may have benefit to our model.

This object detection is referred to the paper (Girshick), which describe a kind of neural network called Fast Region-based Convolutional Network. Comparing with other neural network, this neural network has faster training speed and more accurate detection.

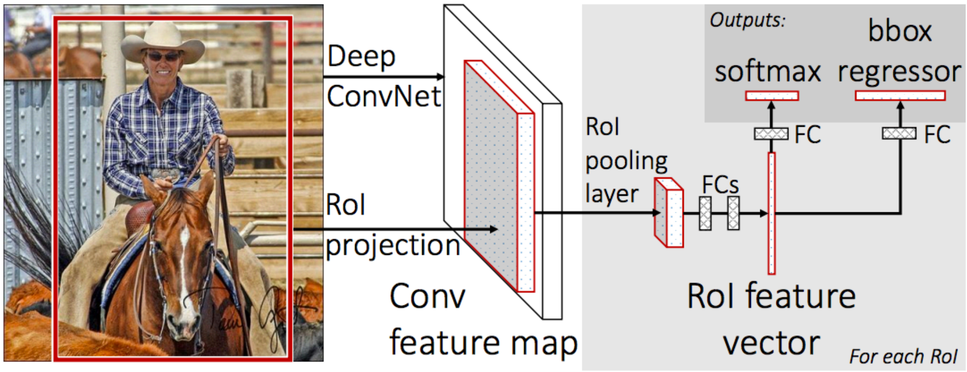


Figure 25 Fast R-CNN Architecture (DIY Deep Learning for Vision: a Hands-On Tutorial with Caffe)

Fast R-CNN (Girshick) uses the whole image and its object proposals as input. First, Fast R-CNN passes the input through several convolutional layers and max-pooling layer to generate the convolutional feature map of this image. After that, the convolutional feature map is passed to a region of interest (RoI) layer to generate a fixed-length feature vector for each proposal. Finally, these feature vectors are passed through two separate fully-connected layer. For the first fully-connect layer which perform softmax function, this layer produces a numerical number for each proposal which indicate the possibility that this object is in this image. For the second fully-connected layer, this layer output four numbers for each proposal which indicate the location of the object in this image.

In our model, as the accuracy of answering question like predicting number of objects is very low, I think Fast R-CNN is very suitable to our model. As mentioned earlies, the output of Fast R-CNN can be considered as a set of regions of interest (rectangles) indicating objects and the corresponding label for each object in these regions of interest. Therefore, by reviewing the output of Fast R-CNN, we can find out the numbers of different objects in input image.

For example, the input image is shown below.

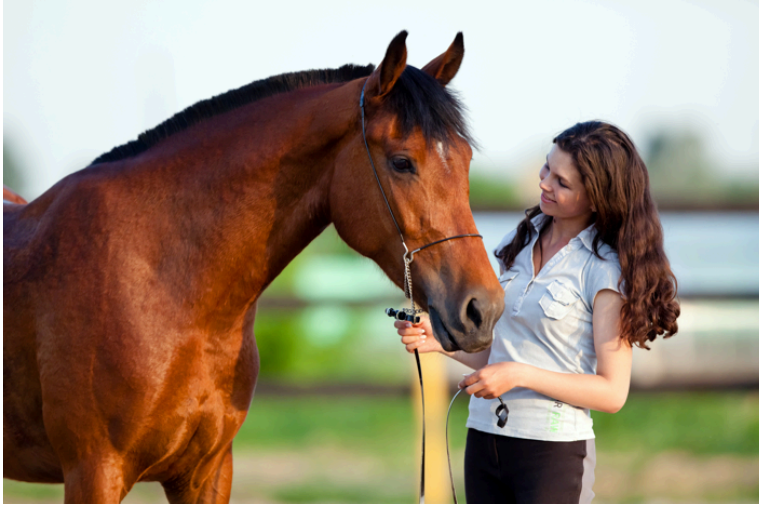


Figure 26 Input Image for Fast R-CNN

In this image, there are a female and a horse. After the processing of Fast R-CNN, we can toughly consider the output as image below.

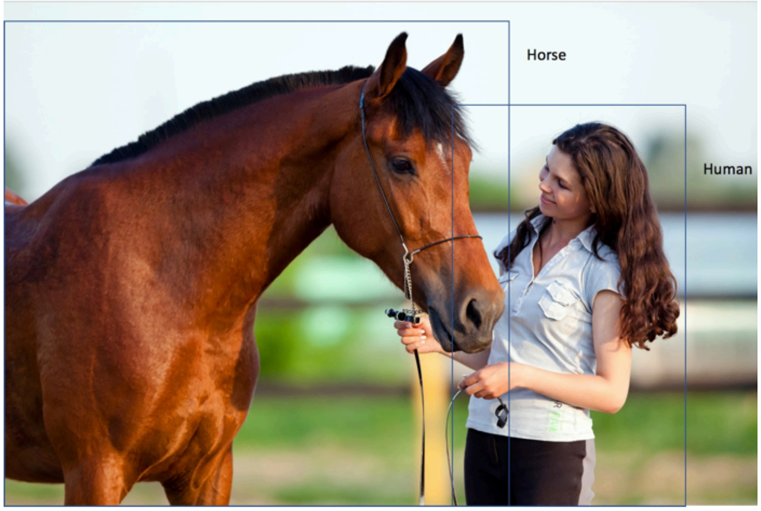


Figure 27 Output of Fast R-CNN

In the output of Fast R-CNN, there are two regions of interest (rectangles) and a region of interest contains a female while the other region of interest contains a horse.

Therefore, by pushing these kinds of information to the LSTM for question answering. It is possible that the model can answer the question about counting, like "How many people are there in this image" or "Is there any horse in this image", more accurate. Therefore, this model can help us improve the accuracy of our visual question answering model.

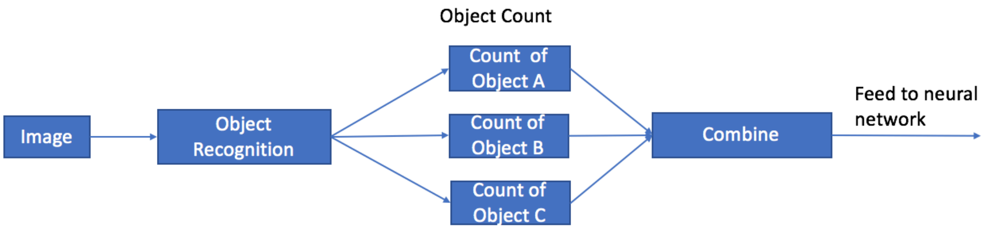


Figure 28 Count Model

In this model, we want to use object recognition to count each object in an image and together with the image features to enhance the accuracy of model.

3.2 Image Caption Model

3.2.1 NIC Model (Show and Tell: A Neural Image Caption Generator)

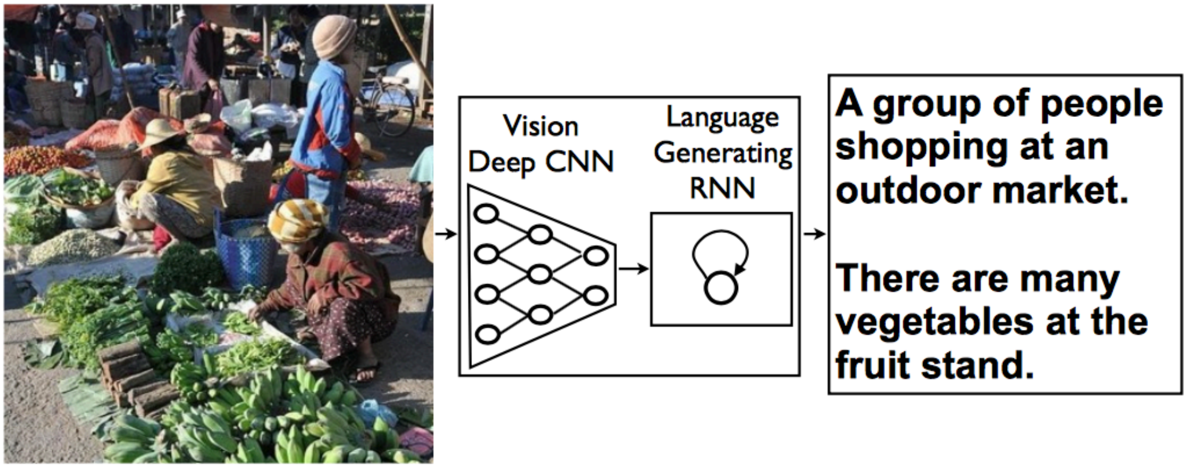
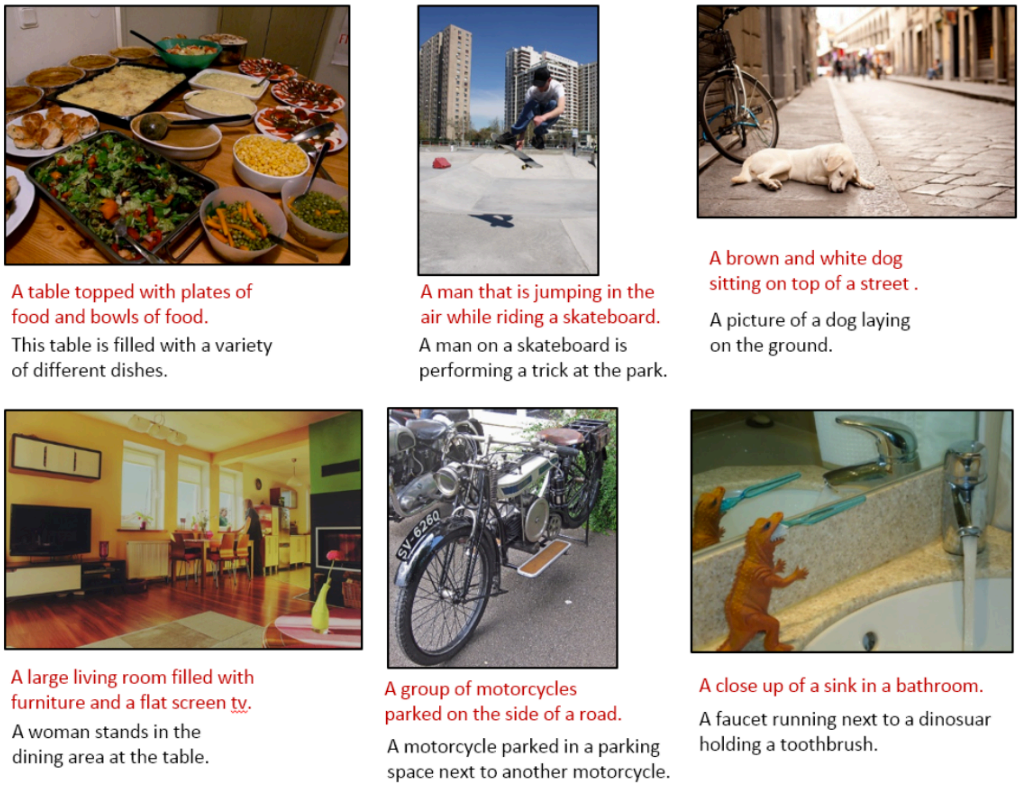

Vision Deep CNN
Language Generating RNN
A group of people shopping at an outdoor market.
There are many vegetables at the fruit stand.

Figure 29 Sample Output

The score of this model is as following:

Metric	BLEU-4	MeTEOR	cider
NIC	27.7	23.7	85.5
Random	4.6	9.0	5.1
Neighbor	9.9	15.7	36.5
Human	21.7	25.2	85.4

3.2.2 Recurrent Visual Representation (Mind's Eye: A Recurrent Visual Representation for Image Caption Generation)



A table topped with plates of food and bowls of food.
This table is filled with a variety of different dishes.

A man that is jumping in the air while riding a skateboard.
A man on a skateboard is performing a trick at the park.

A brown and white dog sitting on top of a street .
A picture of a dog laying on the ground.

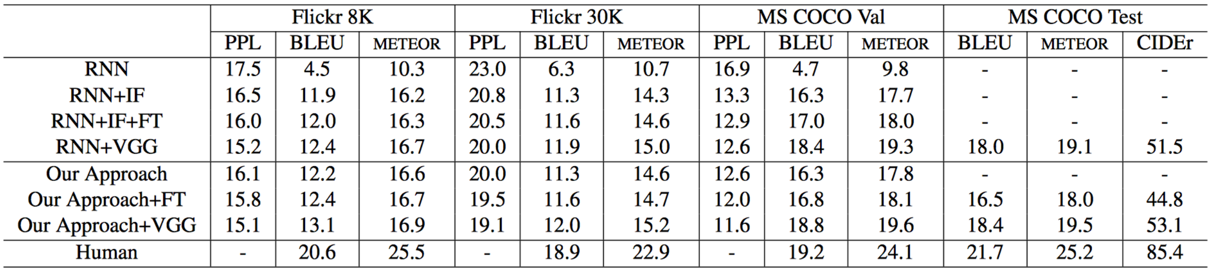
A large living room filled with furniture and a flat screen tv.
A woman stands in the dining area at the table.

A group of motorcycles parked on the side of a road.
A motorcycle parked in a parking space next to another motorcycle.

A close up of a sink in a bathroom.
A faucet running next to a dinosuar holding a toothbrush.

Figure 30 Sample Result

The overall score of this model:



	Flickr 8K			Flickr 30K			MS COCO Val			MS COCO Test		
	PPL	BLEU	METEOR	PPL	BLEU	METEOR	PPL	BLEU	METEOR	BLEU	METEOR	CIDEr
RNN	17.5	4.5	10.3	23.0	6.3	10.7	16.9	4.7	9.8	-	-	-
RNN+IF	16.5	11.9	16.2	20.8	11.3	14.3	13.3	16.3	17.7	-	-	-
RNN+IF+FT	16.0	12.0	16.3	20.5	11.6	14.6	12.9	17.0	18.0	-	-	-
RNN+VGG	15.2	12.4	16.7	20.0	11.9	15.0	12.6	18.4	19.3	18.0	19.1	51.5
Our Approach	16.1	12.2	16.6	20.0	11.3	14.6	12.6	16.3	17.8	-	-	-
Our Approach+FT	15.8	12.4	16.7	19.5	11.6	14.7	12.0	16.8	18.1	16.5	18.0	44.8
Our Approach+VGG	15.1	13.1	16.9	19.1	12.0	15.2	11.6	18.8	19.6	18.4	19.5	53.1
Human	-	20.6	25.5	-	18.9	22.9	-	19.2	24.1	21.7	25.2	85.4

3.2.3 Overall Structure

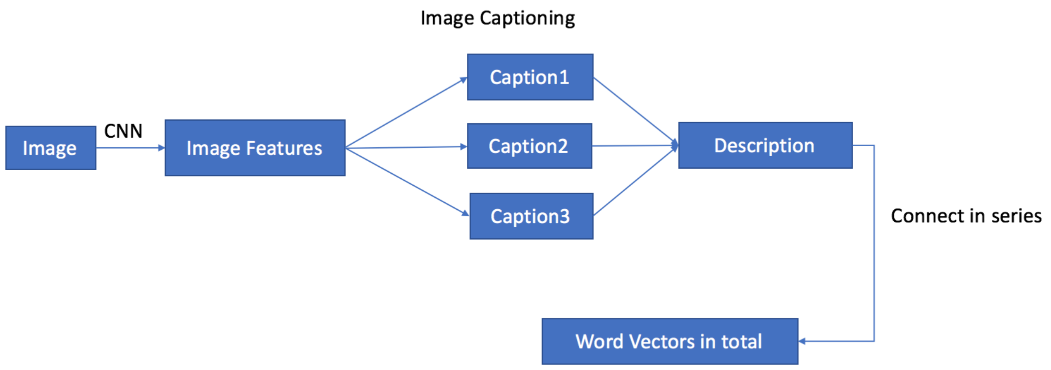


Figure 31 Caption Model

In our first model, we decide to using several different ways to do generate image captions and combine them into a sentence which represent the image. And in this case, we convert an image into sentence so that the Visual Question Answering problem is converted to a Text-Based Question Answering problem.

But this model still has many defects. First, current models in image caption field cannot describe the content in image well and too many information will lose in this process.

4. Dataset

Since we need a large amount of data to train a model, and the data set should contain image, question and answer pairs.

The training and testing dataset we used in our project is provided by the organizers of Visual Question Answering challenge which is sponsored by MS COCO website. This dataset contains three subsets which are train set, validation set and test set. Inside the dataset, there are two types of questions and first one is Open-End questions and another one is Multiple-Choice questions. The detail is as following (16ht):

Real Images:

- 82783 MS COCO training images
- 40504 MS COCO validation images
- 81434 MS COCO testing images

- 248349 questions for training
- 121,512 questions for validation
- 244,302 questions for testing (3 per image)
- 2483490 answers for training
- 1,215,120 answers for validation (10 per question)

Abstract Images:

- 20000 training images
- 10,000 validation images
- 20,000 MS COCO testing images
- 60000 questions for training
- 30000 questions for validation
- 60000 questions for testing (3 per image)
- 600000 answers for training
- 300000 answers for validation (10 per question)

Implementation Detail

The overall architecture of our model is like:

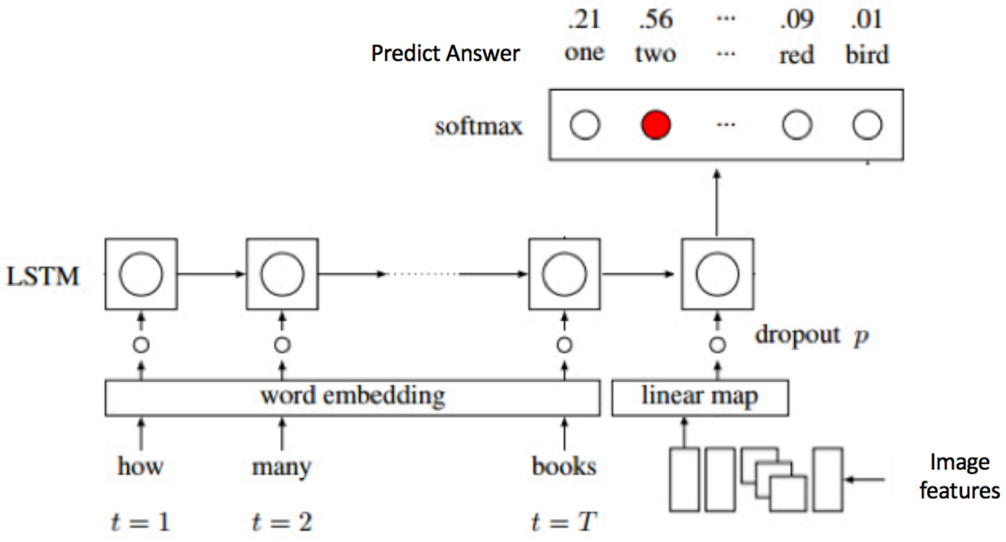


Figure 32 Overall Architecture

In the first, we used a preprocess program to extract the train images' features and using HDF5 binary data format to store it. Then we used the word embedding to convert the words to vectors and combining the question vectors and the image features, and last we feed the combined data to the LSTM (Long Short-Term Memory) to train the model.

IMPLEMENTATION DETAIL

1. Data Loader

Since the training images and the questions are separated in different files and in the format of JSON.

The image file name is like COCO_train2014_[image id].jpg. For example, there is an image whose name is COCO_train2014_000000348957.jpg, then the image is 348957.

In the train question file, the format is as following and it contains an image id and a question id.

```
{
    "question": "How many windows can you see?",
    "image_id": 434410,
    "question_id": 4344102
}
```

The format of answer is as following, and it contains a question id and several choices while each choice has its own answer id.

```
{
        "answer": "kettles",

        "answer_confidence": "yes",

        "answer_id": 6

}
```

Above all, the image id is not and question id may not continuously. In this case, we have a function to load the image, question and answer and binding them together.

2. Extract Image Features

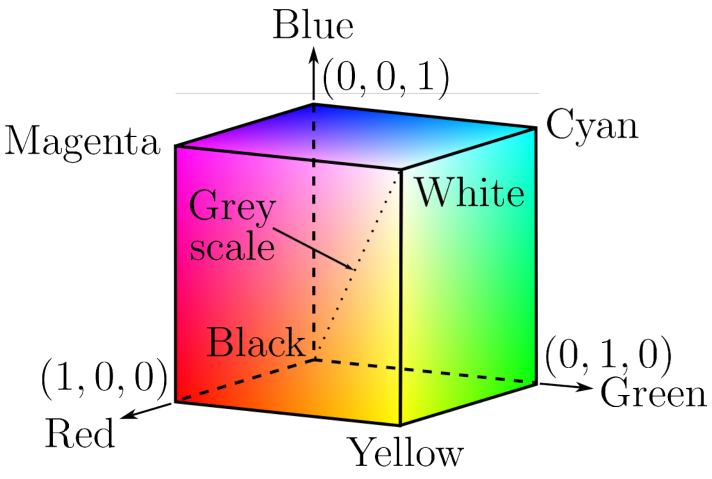


Figure 33 RGB Channels

IMPLEMENTATION DETAIL

After we can load the train data, we need to using Convolutional Neural Network to extract the feature from the images. The VGG-16 model accepts R, G, B channels that between 0 and 1. Then for each image we split it to R channel, G channel and B channel respectively. And then resize it to 224 * 224 dimension.

For example, in the following image, we first separate it into its three color channel, and then resize its dimension to (224, 224).

Figure 34 Train Image

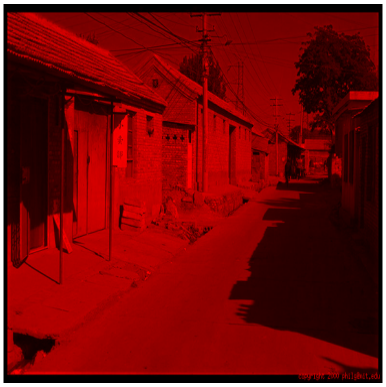


Figure 35 R Channel

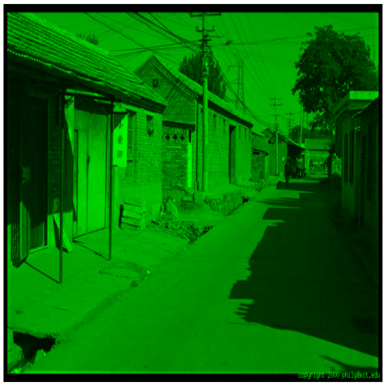


Figure 36 G Channel

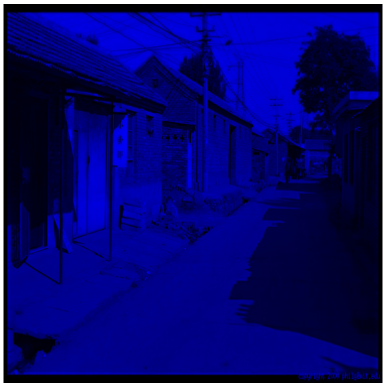


Figure 37 B Channel

```
def load_image_array(image_file):
     img = misc.imread(image_file)
     # GRAYSCALE
     if len(img.shape) == 2:
          img_new = np.ndarray( (img.shape[0], img.shape[1],
3), dtype = 'float32')
          img_new[:,:,0] = img
          img_new[:,:,1] = img
          img_new[:,:,2] = img
          img = img_new

     img_resized = misc.imresize(img, (224, 224))
     return (img_resized/255.0).astype('float32')
```

After preprocessing the train images files, we feed them into the VGG-16 Model to extract their features.

1.1 VGG-16 Model

The preprocess program use the VGG-16 model as the basic model, the structure is like following.

```
layer {
  name: "conv1_1"
  type: "Convolution"
  bottom: "data"
  top: "conv1_1a"
  convolution_param {
    num_output: 64
    pad: 1
    kernel_size: 3
  }
}
layer {
  name: "relu1_1"
  type: "ReLU"
  bottom: "conv1_1a"
  top: "conv1_1"
}
.
.
.
layer {
  name: "drop7"
  type: "Dropout"
  bottom: "fc7"
  top: "fc7"
  dropout_param {
    dropout_ratio: 0.5
  }
}
layer {
  name: "fc8"
  type: "InnerProduct"
  bottom: "fc7"
  top: "fc8"
  inner_product_param {
    num_output: 1000
  }
}
layer {
  name: "prob"
  type: "Softmax"
  bottom: "fc8"
  top: "prob"
}
```

The rough structure can describe as following:

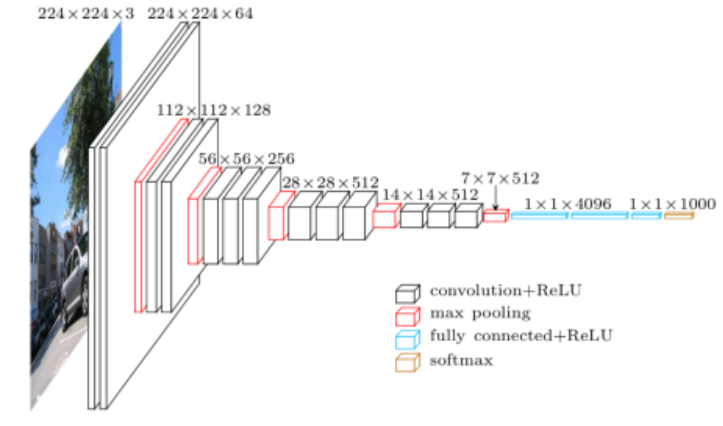
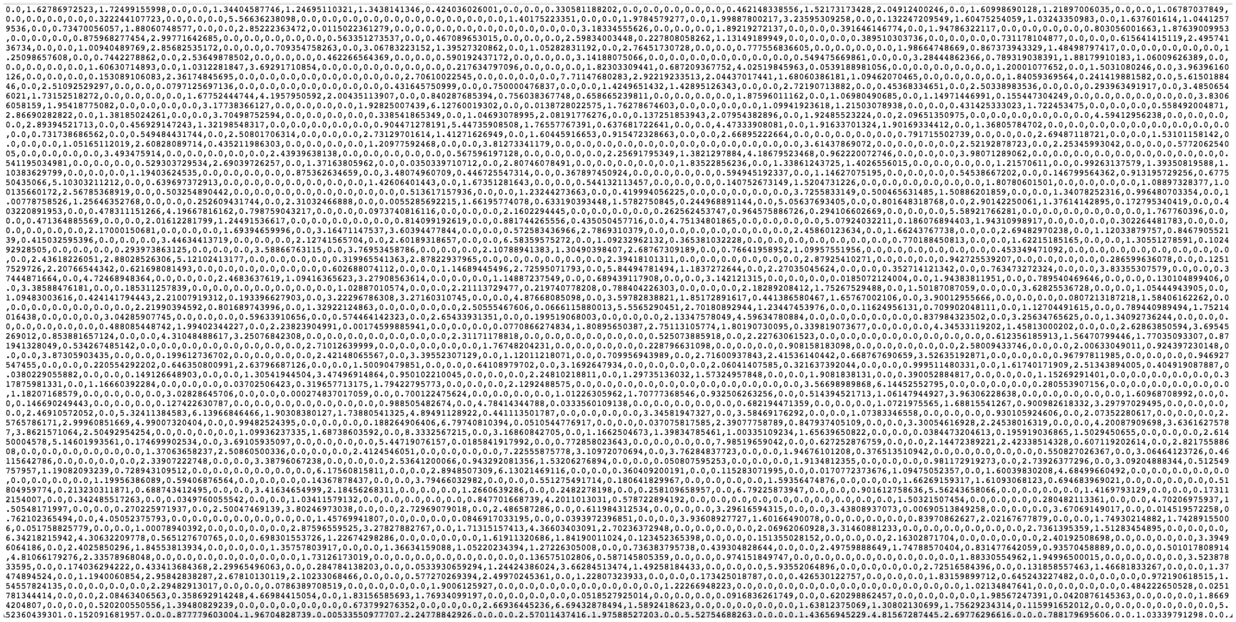


Figure 38 The Structure of VGG-16 Model (htt6)

1.2 Features Format

The feature of an image is a vector whose shape if 4096 * 1.



```
0.0,1.62786972523,1.72499155998,0.0,0.0,1.34404587746,1.24695110321,1.3438141346,0.424036026001,0.0,0.0,0.330581188202,0.0,0.0,0.0,0.0,0.0,0.462148338556,1.52173173428,2.04912400246,0.0,1.60998690128,1.21897006035,0.0,0.0,1.06787037849,
0.0,0.0,0.0,0.0,0.322244107723,0.0,0.0,0.0,5.56636238098,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.40175223351,0.0,0.0,1.9784579277,0.0,1.99887800217,3.23595309258,0.0,0.132247209549,1.60475254059,1.03243350983,0.0,1.637601614,1.0441257
9536,0.0,0.73470056057,1.88060748577,0.0,0.0,2.85222363472,0.0115022361279,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.18334555626,0.0,0.0,0.0,1.89219272137,0.0,0.0,0.391646146774,0.0,1.94786322117,0.0,0.0,0.0,0.0,0.803056001663,1.8763900953
,0.0,0.0,0.0,0.875968277454,2.99771642685,0.0,0.0,0.0,0.0,0.0,0.563351273537,0.0,0.467089653015,0.0,0.0,0.0,2.59834003448,0.227808058262,1.13149189949,0.0,0.0,0.0,0.389510303736,0.0,0.0,0.0,0.731178104877,0.0,0.0,0.615641415119,2.495741
36734,0.0,0.0,1.00940489769,2.85682535172,0.0,0.0,0.709354758263,0.0,3.06783223152,1.39527320862,0.0,1.05282831192,0.0,2.76451730728,0.0,0.0,0.777556836605,0.0,0.0,0.0,0.0,1.98664748669,0.867373943329,1.48498797417,0.0,0.0,0.0,0.0,0.0,0.0,1
.25098657608,0.0,0.7442278862,0.0,2.53649878502,0.0,0.0,0.0,0.462266564369,0.0,0.0,0.590192437172,0.0,0.0,0.0,3.14188075066,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.549475669861,0.0,0.0,3.28444862366,0.789319038391,1.88179910183,1.06009626389,0.0,
0.0,0.0,0.0,0.0,1.60630714893,0.0,1.0312281847,3.69291710854,0.0,0.0,0.0,0.0,0.0,0.0,0.217634797096,0.0,0.0,0.0,0.0,1.82303309441,0.687209367752,4.02519845963,0.0539188981056,0.0,0.0,0.0,0.0,0.0,1.20001077652,0.0,1.5031080246,0.0,3.96396160
126,0.0,0.0,0.0,0.0,0.153089106083,2.36174845695,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.70610022545,0.0,0.0,0.0,0.0,7.71147680283,2.92219233513,2.04437017441,1.68060386181,1.09462070465,0.0,0.0,0.0,0.0,1.84059369564,0.241419881582,0.0,5.61501884
46,0.0,2.51092529297,0.0,0.0,0.0,0.0797125697136,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.431645750999,0.0,0.750000476837,0.0,0.0,1.4249651432,1.42895126343,0.0,0.0,2.72190713882,0.0,0.45368334651,0.0,2.50338983536,0.0,0.0,0.293963491917,0.0,3.4850654
6021,1.73152518272,0.0,0.0,0.0,0.0,1.67752444744,4.1957950592,2.00435113907,0.0,0.840287685394,0.756038367748,0.658665239811,0.0,0.0,0.0,0.0,0.0,1.87596011162,0.0,1.06980490685,0.0,1.14971446991,0.155447304249,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.8306
6058159,1.95418775082,0.0,0.0,0.0,0.0,3.17738366127,0.0,0.0,0.0,0.0,1.92825007439,6.12760019302,0.0,0.0138728022575,1.76278674603,0.0,0.0,0.0,0.0,1.09941923618,1.21503078938,0.0,0.0,0.431425333023,1.722453475,0.0,0.0,0.0,0.0,0.558492004871,
2.86690282822,0.0,1.38185024261,0.0,0.0,3.70498752594,0.0,0.0,0.0,0.0,0.0,0.338541865349,0.0,1.04693078995,2.08191776276,0.0,0.137251853943,2.07954382896,0.0,1.92485523224,0.0,2.09651350975,0.0,0.0,0.0,0.0,0.0,0.0,4.59412956238,0.0,0.0,0.0,0.0,
0.0,2.89394521713,0.0,0.456929147243,1.32198548317,0.0,0.0,0.0,0.0,0.0,0.904471278191,5.44735908508,1.76557767391,0.637681722641,0.0,0.0,4.47333908081,0.0,1.91633701324,1.90169334412,0.0,1.36805784702,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
,0.0,0.731738686562,0.0,0.0,0.549484431744,0.0,2.50801706314,0.0,0.0,0.0,2.73129701614,1.41271626949,0.0,1.60445916653,0.915472328663,0.0,0.0,2.66895222664,0.0,0.0,0.0,0.0,0.0,0.0,0.791715502739,0.0,0.0,0.0,0.0,2.69487118721,0.0,0.0,1.53101158142,0
.0,0.0,0.0,1.05165112019,2.60828089714,0.435211986303,0.0,0.0,0.0,0.0,0.0,1.20977592468,0.0,0.0,3.81273341179,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.61437869072,0.0,0.0,0.0,0.0,0.0,2.52192878723,0.0,2.25345993042,0.0,0.0,0.0,0.5772062540
05,0.0,0.0,0.0,0.0,0.0,3.493475914,0.0,0.0,0.0,0.0,0.0,0.0,2.43939638138,0.0,0.0,0.0,0.0,0.0,0.0,0.567596197128,0.0,0.0,0.0,0.0,0.0,2.25691795349,1.3821297884,4.18679523468,0.962220072746,0.0,0.0,0.0,0.0,3.98071289062,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
541195034981,0.0,0.0,0.0,0.529303729534,2.69039726257,0.0,1.37163805962,0.0,0.0350339710712,0.0,2.80746078491,0.0,0.0,0.0,0.0,0.0,1.83522856236,0.0,1.33861243725,1.4026556015,0.0,0.0,0.0,0.0,1.21570611,0.0,0.99263137579,1.39350819588,1.
10383629799,0.0,0.0,0.0,1.19403624535,0.0,0.0,0.0,0.0,0.875362634659,0.0,3.48074960709,0.446725547314,0.0,0.367897450924,0.0,0.0,0.0,0.0,0.594945192337,0.0,1.14627075195,0.0,0.0,0.0,0.54538667202,0.0,0.146799564362,0.913195729256,0.6775
50435066,5.10303211212,0.0,0.0,0.639697372913,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.42606401443,0.0,1.67351281643,0.0,0.0,0.0,0.544132113457,0.0,0.0,0.0,0.0,0.140752673149,1.5204731226,0.0,0.0,0.0,0.0,0.0,0.0,1.80780601501,0.0,0.0,0.0,0.0,0.0,1.08897328377,1.0
0135660172,2.56785368919,0.0,0.0,0.503254890442,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.513617157936,0.0,0.0,1.23244273663,0.0,0.419994056225,0.0,0.0,0.0,0.0,0.0,3.7255833149,0.500465631485,1.50886201859,0.0,0.0,1.34078252316,0.996480703354,0.0,1
.00778758526,1.25646352768,0.0,0.0,0.252609431744,0.0,2.31032466888,0.0,0.0055285692215,1.66195774078,0.633190393448,1.5782750845,0.244968891144,0.0,5.05637693405,0.0,0.801648318768,0.0,2.90142250061,1.37614142895,0.172795340419,0.0,0.4
03220891953,0.0,0.478311151266,4.19667816162,0.798759043217,0.0,0.0,0.0973740816116,0.0,0.0,0.0,2.1602294445,0.0,0.0,0.0,0.0,0.0,0.262562453747,0.964575886726,0.294106602669,0.0,0.0,0.0,5.58921766281,0.0,0.0,0.0,0.0,0.0,0.0,1.767760396,0.0,
0.0,0.471364885569,0.0,0.0,2.01612281799,1.24491536617,0.0,0.0,0.0,0.0,0.0,0.814099192619,0.0,0.881744265556,0.435050457716,0.0,4.75134801865,0.0,0.0,0.0,0.0,5.07924032211,0.186076894403,1.94310998917,0.0,0.0,0.0,0.302264481783,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,2.17000150681,0.0,0.0,0.0,1.69394659996,0.0,3.16471147537,3.60394477844,0.0,0.0,0.572583436966,2.7869310379,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.45860123634,0.0,1.66243767738,0.0,0.0,2.69482970238,0.0,1.12033879757,0.8467905521
39,0.415032595396,0.0,0.0,0.0,3.44634413719,0.0,0.0,0.0,2.12741565704,0.0,2.60189318657,0.0,0.0,6.58359575272,0.0,1.09232962132,0.365381032228,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.770188450813,0.0,0.0,1.62215185165,0.0,0.0,1.30551278591,0.1024
92928505,0.0,0.0,0.0,0.293973863125,0.0,0.0,0.0,3.58866763115,0.0,3.76953458786,0.0,0.0,0.0,2.10788941383,1.30490398407,2.68767309189,0.0,0.76641958952,1.09957551956,0.0,0.0,0.0,0.0,0.0,0.0,0.453349471092,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
,0.0,2.43618226051,2.88028526306,5.12102413177,0.0,0.0,0.0,0.0,0.319965541363,2.87822937965,0.0,0.0,0.0,0.0,0.0,0.0,2.39418101311,0.0,0.0,0.0,0.0,0.0,0.0,2.87925410271,0.0,0.0,0.0,0.0,0.942725539207,0.0,0.0,0.0,0.0,0.0,0.286599636078,0.0,0.1251
7529726,2.20766544342,0.621698081493,0.0,0.0,0.0,0.0,0.0,0.0,0.602688074112,0.0,0.0,1.14689445496,2.72595071793,0.0,5.84494781494,1.1837272644,0.0,2.27035045624,0.0,0.0,0.352714121342,0.0,0.763473272324,0.0,0.0,3.83355307579,0.0,0.0,0.0,0.3
7444871664,0.0,4.72468948364,0.0,0.0,0.0,2.4683637619,1.09416365623,3.27908563614,0.0,0.0,0.0,1.14887237549,0.0,0.689439117908,0.0,0.0,3.142121315,0.0,0.0,0.0,0.0185072124004,0.0,1.94383811951,0.0,0.789540469646,0.0,0.0,0.130104899406,0
.0,3.38588476181,0.0,0.185311257839,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.02887010574,0.0,0.0,2.21113729477,0.219740778208,0.788404226303,0.0,0.0,0.0,2.18289208412,1.75267529488,0.0,1.50187087059,0.0,0.0,3.62825536728,0.0,0.0,1.05444943905,0.0,
1.09483003616,0.424141794443,2.21007919312,0.193396627903,0.0,3.22296786308,3.27160310745,0.0,0.0,4.87668085098,0.0,3.59782838821,1.85172891617,0.441386580467,1.65767002106,0.0,3.90012955666,0.0,0.0,0.0,0.0807213187218,1.58406162262,0.0
,0.0,0.0,0.0,0.0,0.0,0.0,2.21990394592,0.801689743996,0.0,1.32922124863,0.0,0.0,0.0,0.0,2.50555467606,0.0666115880013,5.5565290451,2.70180892944,1.23447453976,0.0,0.11624956131,0.709902048111,0.0,1.12704491615,0.0,0.789440989494,1.75214
016438,0.0,0.0,0.0,0.0,3.04285907745,0.0,0.0,0.0,0.0,0.59633910656,0.0,0.574464142323,0.0,2.65433931351,0.0,0.199519068003,0.0,0.0,0.0,0.0,2.13347578049,4.59634780884,0.0,0.0,0.0,0.0,0.0,0.0,0.837984323502,0.0,3.25634765625,0.0,1.34092736244,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.488085448742,1.99402344227,0.0,2.23823904991,0.00174599885941,0.0,0.0,0.0,0.0,0.0770866274834,1.80895650387,2.75113105774,1.80190730095,0.339819073677,0.0,0.0,0.0,4.34533119202,1.45813000202,0.0,0.0,2.62863850594,3.69545
269012,0.853881657124,0.0,0.0,4.31048488617,3.25076842308,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.31171178818,0.0,0.0,0.0,0.0,0.0,0.0,0.525073885918,0.0,2.22763061523,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.612356185913,1.56470799446,1.77035093307,0.87
1941328049,0.534267485142,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,2.71012639999,0.0,0.0,0.0,0.0,0.0,0.0,1.76748204231,0.0,0.0,0.0,0.0,0.0,0.228796631098,0.0,0.0,0.0,0.0,0.908158183098,0.0,0.0,0.0,0.0,0.0,0.0,2.58009433746,0.0,0.0,0.0,2.00633049011,0.924397230148,0
.0,0.0,3.87305903435,0.0,0.0,0.0,0.199612736702,0.0,0.0,0.0,0.0,0.0,2.42148065567,0.0,3.39552307129,0.0,1.12011218071,0.0,0.709956943989,0.0,2.71600937843,2.41536140442,0.668767690659,3.52635192871,0.0,0.0,0.0,0.0,0.96797811985,0.0,0.0,0.0,0.0,0.946927
547455,0.0,0.0,0.0,0.220554292202,0.646350800991,2.63796687126,0.0,0.0,1.50090479851,0.0,0.0,0.64108979702,0.0,3.1692647934,0.0,0.0,0.0,0.0,0.0,2.06041407585,0.321637392044,0.0,0.0,0.0,0.999511480331,0.0,1.61740171909,2.51343894005,0.404919087887,0
.0380229055882,0.0,0.0,0.0,0.149126648903,0.0,0.0,0.0,1.30541944504,3.47496914864,0.950102210045,0.0,0.0,0.0,2.24810218811,0.0,1.29735136032,1.57324957848,0.0,0.0,1.9081838131,0.0,0.0,0.390052884817,0.0,0.0,0.0,0.0,1.15269291401,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.3
17875981331,0.0,1.16660392284,0.0,0.0,0.0,0.0,0.03702506423,0.319657713175,1.79422795773,0.0,0.0,0.0,0.0,2.1292488575,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.56698989868,6.14452552795,0.0,0.0,0.0,0.0,0.0,0.280553907156,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
,1.18207168579,0.0,0.0,0.0,0.0,3.02828645706,0.0,0.0,0.000274837017059,0.0,0.700122475624,0.0,0.0,0.0,0.0,1.01226305962,1.70777368546,0.932506263256,0.0,0.514394521713,1.06147944927,3.96306228638,0.0,0.0,0.0,0.0,0.0,0.0,1.60968708992,0.0,0.
0,0.146690249443,0.0,0.0,0.0,0.0,0.127422630787,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.988505482674,0.0,4.7841434478 8,0.0333560109138,0.0,0.0,0.0,0.0,0.0,0.682194471359,0.0,0.0,0.0,1.0721975565,1.68815541267,0.900982618332,3.29797029495,0.0,0.0,0.0,0.0,
0.0,2.46910572052,0.0,5.32411384583,6.13966846466,1.90308380127,1.73880541325,4.89491128922,0.441113501787,0.0,0.0,0.0,0.0,0.0,3.34581947327,0.0,3.58469176292,0.0,0.0,1.07383346558,0.0,0.0,0.0,0.0,0.930105924606,0.0,2.07352280617,0.0,0.0,0.0,0.0,2.
5765786171,2.99960851669,4.99007320404,0.0,0.99482524395,0.0,0.0,0.0,0.188264906406,6.79740810394,0.0510544776917,0.0,0.0,0.037075817585,2.39077758789,0.847937405109,0.0,0.0,3.30054616928,2.24538016319,0.0,0.0,4.20087909698,3.6361627578
7,3.8621571064,2.50492954254,0.0,0.0,0.0,0.0,1.09936237335,1.68738603592,0.0,8.3332567215,0.0,3.16860842705,0.0,1.1662504673,1.39834785461,1.00335109234,1.65639650822,0.0,0.0,0.0384473204613,0.195919036865,1.5029450655,0.0,0.0,0.0,2.614
50004578,5.14601993561,0.174699902534,0.0,3.69105935097,0.0,0.0,0.0,0.0,5.44719076157,0.0185841917992,0.0,0.772858023643,0.0,0.0,0.0,0.0,7.98519659042,0.0,0.627252876759,0.0,0.0,2.14472389221,2.42338514328,0.607119202614,0.0,2.821755886
08,0.0,0.0,0.0,0.0,0.0,0.0,1.37063658237,2.50860500336,0.0,0.0,0.0,2.4124546051,0.0,0.0,0.0,0.0,7.22555875778,3.10972070694,0.0,3.76284837723,0.0,0.0,1.94676101208,0.376513510942,0.0,0.0,0.0,0.0,0.0,0.550827026367,0.0,3.06464123726,0.46
115642786,0.0,0.0,0.0,0.0,2.33907222748,0.0,0.0,3.38796067238,0.0,0.0,0.0,2.53641200066,0.943292081356,1.53206276894,0.0,0.0,0.050807595253,0.0,0.0,0.0,1.9134812355,0.0,0.0,0.0,0.981172919273,0.0,2.73926377296,0.0,3.09204888344,0.512549
757957,1.19082093239,0.728943109512,0.0,0.0,0.0,0.0,0.0,0.0,6.17560815811,0.0,0.0,2.8948507309,6.13021469116,0.0,0.0,0.360409200191,0.0,0.115283071995,0.0,0.0170772373676,1.09475052357,0.0,1.60039830208,4.68499660492,0.0,0.0,0.0,0.0,0.0,0.0
,0.0,0.0,0.0,0.0,1.1995638608 9,0.59406876564,0.0,0.0,0.0,0.14367878437,0.0,0.0,3.79466032982,0.0,0.0,0.551275491714,0.180641829967,0.0,0.0,0.0,0.0,1.59356474876,0.0,0.0,0.0,1.66269159317,1.61093068123,0.694683969021,0.0,0.0,0.0,0.0,0.0,0.51
804959774,0.213230311871,0.688743412495,0.0,0.0,3.41634654999,2.18456268311,0.0,0.0,0.0,1.2660639286,0.0,0.2482278198,0.0,0.258109658957,0.0,6.79225873947,0.0,0.0,0.901612758636,5.56243658066,0.0,0.0,0.0,0.0,1.4169793129,0.0,0.0,0.17311
2154007,0.0,0.342485517263,0.0,0.0349760055542,0.0,0.0,1.03411579132,0.0,0.0,0.0,0.0,0.0,0.0,0.847701668739,4.2011013031,0.578722894192,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.50321507454,0.0,0.0,0.0,0.280482113361,0.0,0.0,4.70206975937,1
.50548171997,0.0,0.0,0.270225971937,0.0,2.50047469139,3.80246973038,0.0,0.0,2.72969079018,0.0,2.486587286,0.0,0.611984312534,0.0,0.0,0.0,3.29616594315,0.0,0.0,3.43808937073,0.00690513849258,0.0,0.0,0.0,3.67069149017,0.0,0.014519572258,0
.762102365494,0.0,4.05052375793,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.45769941807,0.0,0.0,0.0,0.0846917033195,0.0,0.0393972396851,0.0,0.0,3.93608927727,1.60166490078,0.0,0.0,0.0,0.0,0.83970862627,2.02167677879,0.0,0.0,1.74930214882,1.7428915500
6,0.051758825779,0.0,0.0,1.00078940392,0.0,0.0,0.0,0.0,2.87596559525,3.27827882767,0.0,1.71315157413,4.36603403091,2.70236372948,0.0,0.0,0.0,0.0,2.06962060928,3.31460881233,0.0,0.0,0.0,0.0,0.0,0.0,2.7361395359,1.51283454895,0.0,0.0,0.0,0.0,
6.34218215942,4.30632209778,0.565127670765,0.0,0.698301553726,1.22674298286,0.0,0.0,0.0,0.0,1.61911320686,1.84190011024,0.123452365398,0.0,0.0,0.151355028152,0.0,0.0,0.0,2.16302871704,0.0,0.0,0.0,0.0,2.40192508698,0.0,0.0,0.0,0.0,0.0,3.3949
6064186,0.0,2.4025850296,1.84553813934,0.0,0.0,0.0,1.35757803917,0.0,0.0,1.36634159088,1.05220234394,1.27226305008,0.0,0.736383795738,0.439304828644,0.0,0.0,2.49759888649,1.74788570404,0.831477642059,0.93570458889,0.0,0.0,0.501017808914
,4.81066179276,2.33578968048,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.73126173019,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.136575102806,0.587145805359,0.0,0.0,0.974151849747,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.88330554962,1.94996500015,0.0,0.0,0.0,0.0,0.0,3.523878
33595,0.0,0.174036294222,0.433413684368,2.29965496063,0.0,0.284784138203,0.0,0.0533930659294,1.24424386024,3.66284513474,1.49258184433,0.0,0.0,0.0,5.93552064896,0.0,0.0,0.0,0.0,2.72516584396,0.0,0.131858557463,1.46681833267,0.0,0.0,1.37
474894524,0.0,1.1940060854,2.95842838287,2.67810130119,2.10233068466,0.0,0.0,0.577270269394,2.49970245361,0.0,1.22807323933,0.0,0.0,0.173425018787,0.0,0.426530122757,0.0,0.0,0.0,1.83159899712,0.645243227482,0.0,0.0,0.0,0.972190618515,1.
54557824135,0.0,0.0,0.0,0.0,2.29482913017,0.0,0.0,0.0786389708519,0.0,0.0,0.0,0.0,1.9006125927,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.22266948223,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.02134847641,0.0,0.0,0.0,0.0,0.48422265052 8,0.0251
781344414,0.0,0.0,2.08463406563,0.358692914248,4.66984415054,0.0,1.83156585693,1.76934099197,0.0,0.0,0.0,0.0,0.0518527925014,0.0,0.0,0.0,0.0916836261749,0.0,0.620298862457,0.0,0.0,0.0,0.0,1.98567247391,0.0420876145363,0.0,0.0,0.0,1.8669
4204807,0.0,0.0,0.520200550556,1.39480829239,0.0,0.0,0.0,0.0,0.0,0.673799276352,0.0,0.0,0.0,2.66936445236,6.69432878494,1.5892418623,0.0,0.0,0.0,0.0,0.0,0.0,1.63812375069,1.30802130699,1.75629234314,0.115991652012,0.0,0.0,0.0,0.0,0.0,0.0,0.0,5.
52360439301,0.152091681957,0.0,0.877779603004,1.96704828739,0.00533550977707,2.24778842926,0.0,0.0,2.57011437416,1.97588527203,0.0,5.52754688263,0.0,0.0,1.43656945229,4.81567287445,2.69776296616,0.0,0.788179695606,0.0,1.03339791298,0.0,
```

Figure 39 Feature of An Image

For each train images, we have a vector whose shape is 4096 * 1, after we finishing extracting the image features, we can obtain a matrix whose shaper is number of images * 4096, and this large matrix will store in HDF5 binary data format file. Since HDF5 has a smaller size when we store the data.

3. Training

We build up a LSTM Model. This LSTM model contains 2 hidden layers. and each hidden layer has 512 neurons. Therefore, for each hidden layer, it will output a 512-length vector.

For Input layer, it maintains three parameters which are Wimg, Wemb, bimg. Wimg is a matrix [4096, 512] and all elements in this matrix are random values in range [-1, 1] initially. Wemb is a matrix [q_vob_len, 512] where q_vob_len is the number of words in all questions and all the elements in this matrix are 0 initially. bimg is a 512-length vector which represents the bias. The usages of three parameters will be described below. When the data batch feed in, since the dimension of image features and words is not the same, we need to regularize these dimensions. For words, we do word embedding to generate a 512-length vector to represents words using Wemb. For image, we multiply the Wimg and 4096-length image feature vector and add the bimg to generate another 512-length vector. Therefore, the dimensions of image feature vector and word vector are regularized.

4. Predicting

We first load the pre-trained QA model and VGG model from local. Then we use VGG to produce the image feature vector of input image. After that, we use pre-known knowledge to embedding the words in question and reduce the dimension of image feature vector. We concatenate these vectors and feed resultant matrix into pre-trained QA and generate the word vector representation of answer. Finally, we look up this vector in our pre-known knowledge and find out the answer.

Results and Analysis

1. Training Process

The train log can help us to improve our models, and the following figures are showing the Training Accuracy and Losses during each epoch.

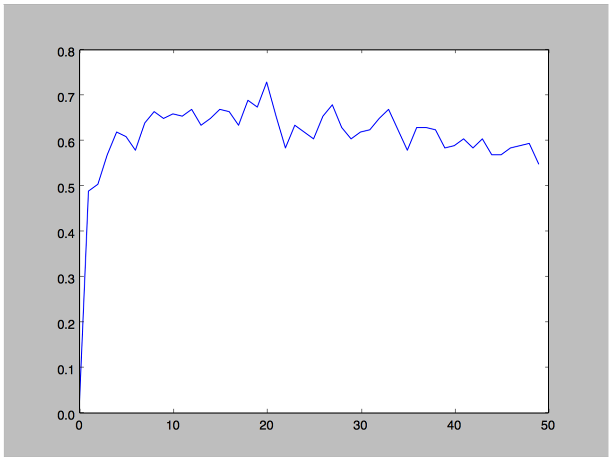


Figure 40 Training Accuracy

As we can from the above figure, the training accuracy go up quickly in the beginning of train process, then in maintain an accuracy between 50% to 60% and trend to converge in the middle of training process. And after finishing the whole training, the final accuracy is about 55%.

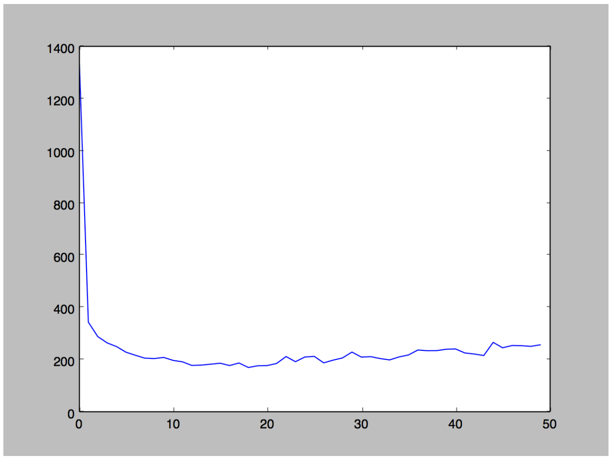


Figure 41 Training Losses

From the above figure, we can know that the train process was about to converge quickly. The training accuracy and the training losses didn't change a lot after epoch 10. And the total training loss is about 300 eventually.

2. Evaluation

2.1 Evaluation Metric

We use the evaluation code which offer by the Visual Question Answering challenge organizers. In this evaluation metric, the accuracy of machine will be averaged over all 10 choose 9 sets of human annotators to enhance the robustness of accuracy.

The accuracy formula is as follow:

$$Acc(ans) = \min\{\frac{\#\ humans\ that\ said\ ans}{3}, 1\}$$

2.2 Evaluation Dataset

The dataset we used for evaluation is mentioned in Chapter Methodology.

3. Accuracy

3.1 Overall Test Result

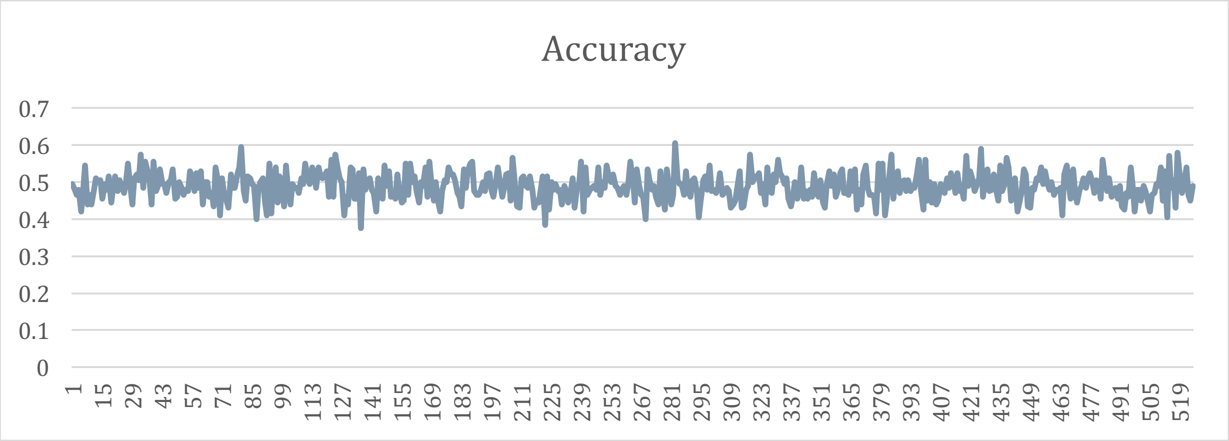


Figure 42 Overall Accuracy of Current Model

3.2 Detail Accuracy

Table 1 Accuracy of Our Model

	Test Set			
Model	Yes / No	Number	Other	Overall

Baseline All Yes	70.97	0.36	1.21	29.88
Baseline Prior Per Question Type	71.40	34.90	8.96	37.47
Baseline Nearest Neighbor	71.89	24.23	22.10	42.85
UC Berkeley & Sony	83.62	39.47	58.00	49.12
Our Model	74.62	31.76	31.32	49.12

4. Samples

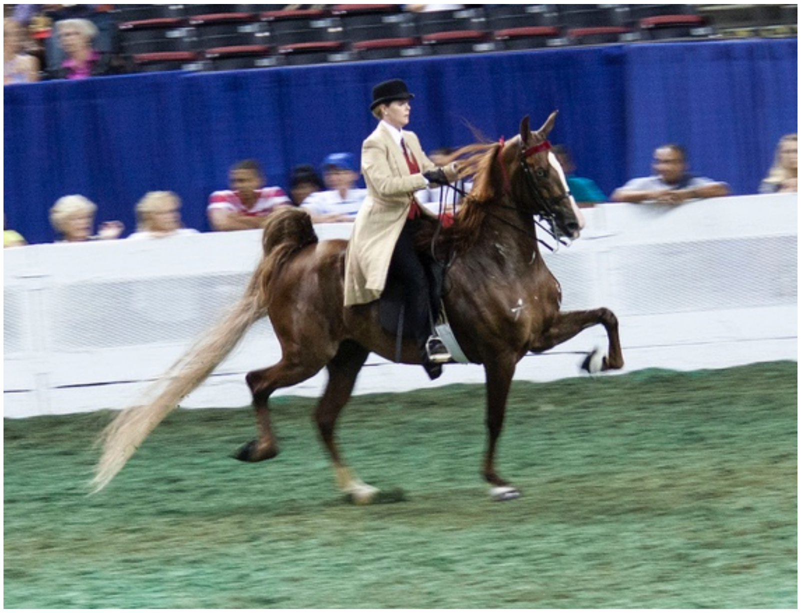


Figure 43 Sample 1

Question: What animal is this?

Top 5 Answers: horse, cow, elephant, dog, zebra

Question: What color is this animal?

Top 5 Answers: brown, white and brown, white, black and white, black

Question: What are they doing now?

Top 5 Answers: walking, talking, eating, standing, taking picture

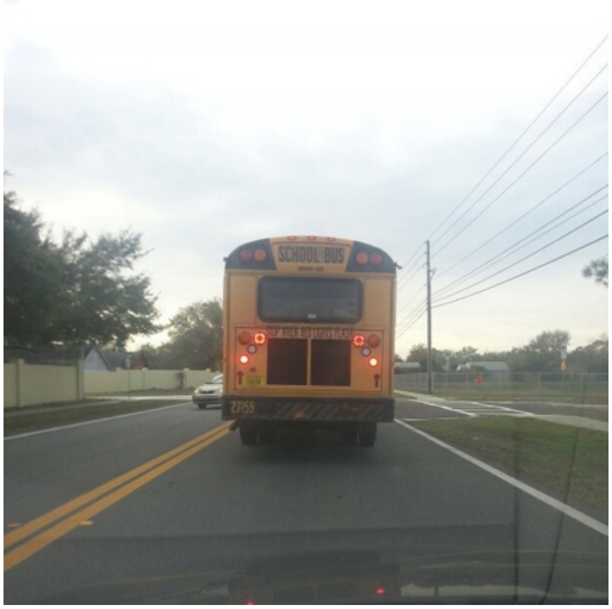


Figure 44 Sample 2

Question: What is this that in the front?

Top 5 Answers: train, building, truck, bikes, teddy bears

Question: How about the weather now?

Top 5 Answers: cloudy, cold, rainy, wet, evening

Question: How many cars are in there?

Top 5 Answers: 2, 3, 1, 5, 4

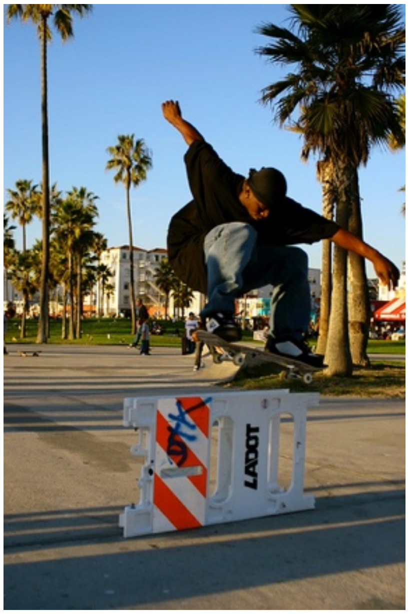


Figure 45 Sample 3

Question: What is this man doing?

Top 5 Answer: skateboarding, jumping, snowboarding, sitting, taking picture

Question: How about the weather?

Top 5 Answer: rainy, sunny, cloudy, clear, overcast

Question: What is the gender of the person?

Top 5 Answer: male, female, child, boy, women

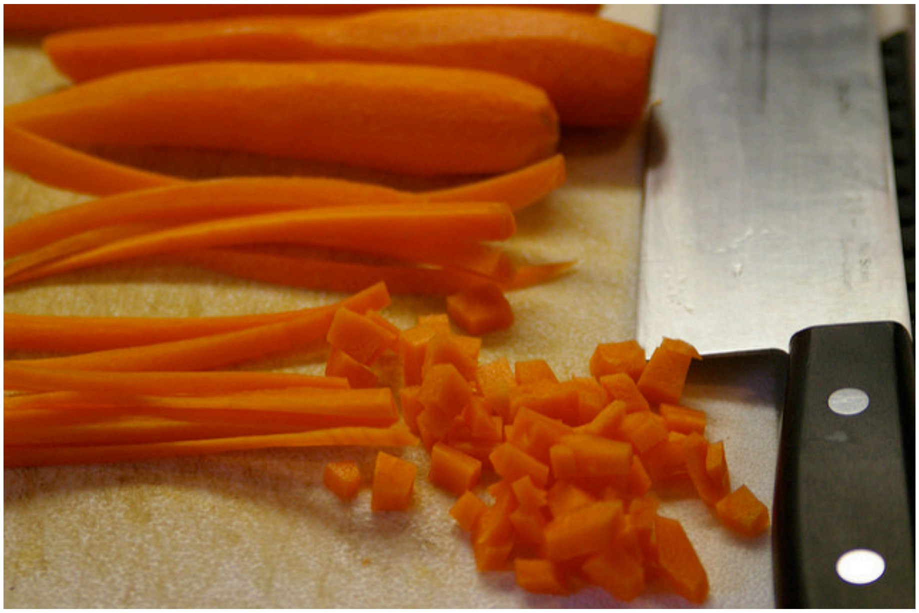


Figure 46 Sample 4

Question: what are sitting on the counter in different stages of cutting with a knife?

Top 5 Answers: napkin, chocolate, glass, bread, fries

Question: What is the color of this vegetables?

Top 5 Answer: orange, yellow, green, blue, brown

Question: What is the name of this vegetable?

Top 5 Answer: carrot, parsley, apple, bananas, carrots

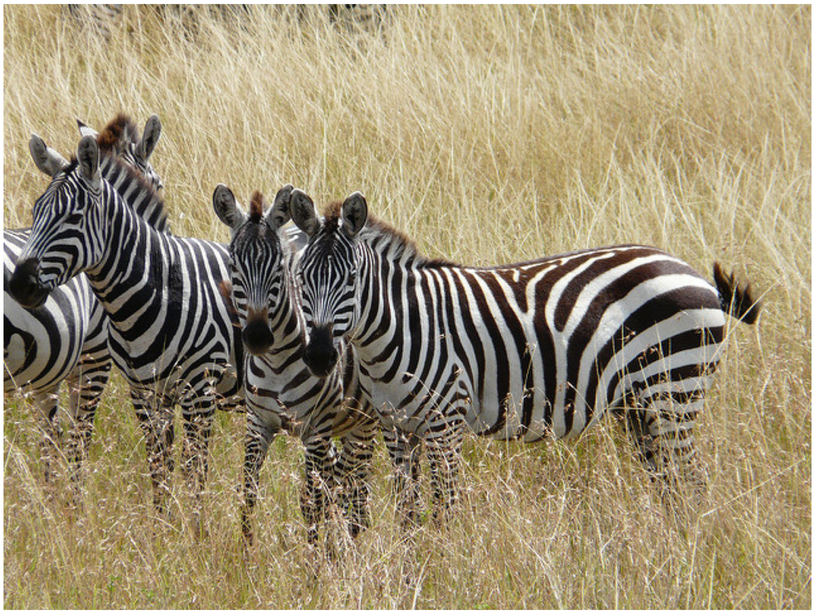


Figure 47 Sample 5

Question: What is the color of this animal?

Top 5 Answer: black and white, white, brown, black, gray

Question: What is this animal?

Top 5 Answer: zebra, giraffe, horse, cow, zebras

Question: How many animals are there?

Top 5 Answer: 2, 3, 4, 1, 5

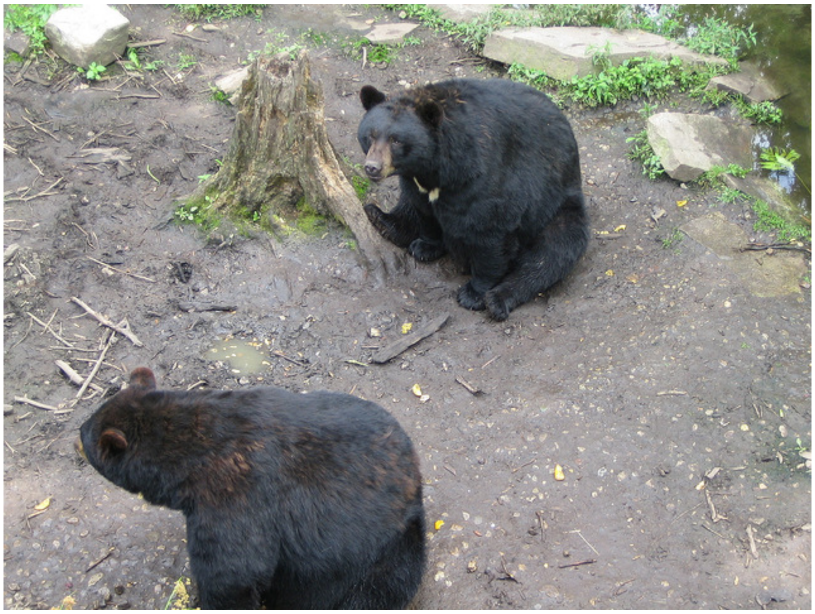


Figure 48 Sample 6

Question: How many animals are there?

Top 5 Answer: 2, 1, 3, 4, 5

Question: What is the name of this animal?

Top 5 Answer: bird, bear, elephant, cat, dog

Question: what are sitting down on the ground?

Top 5 Answer: teddy bear, bear, snow, stick, bird

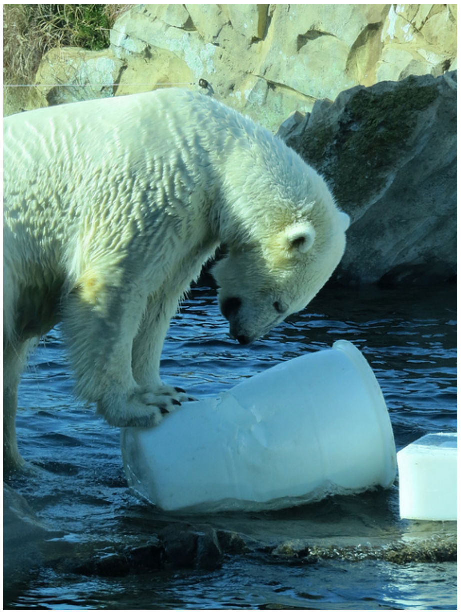


Figure 49 Sample 7

Question: what is playing with the large UNK of ice?

Top 5 Answer: surfer, surfing, surfboard, water, boat

Question: What is the color of this animal?

Top 5 Answer: white, brown, black, white and brown, gray

Question: How many animals are there in the image?

Top 5 Answer: 2, 1, 4, 3, 6

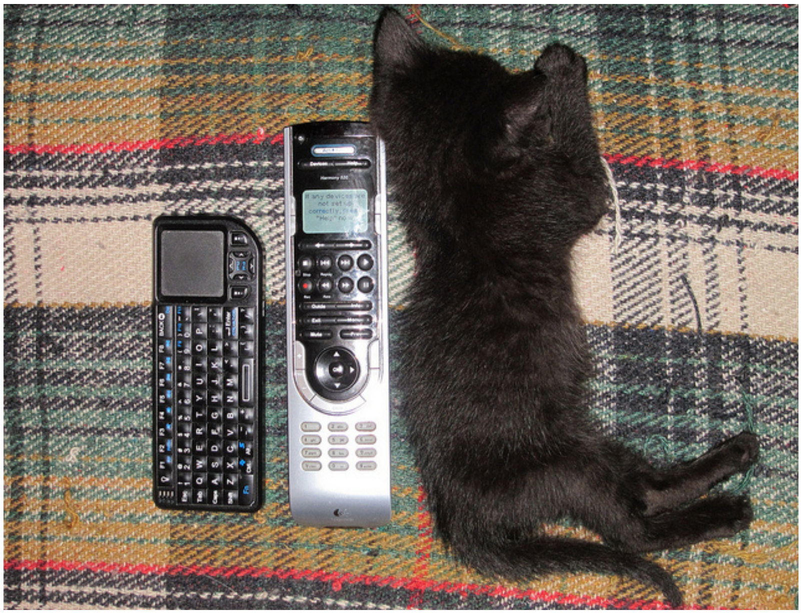


Figure 50 Sample 8

Question: How many animals are there in the image?

Top 5 Answer: 2, 3, 1, 4, 5

Question: What is the name of object that in the middle?

Top 5 Answer: remote, laptop, toy, cat, books

Question: What is the name of object that in the left?

Top 5 Answer: laptop, keyboard, remote, phone, bowl

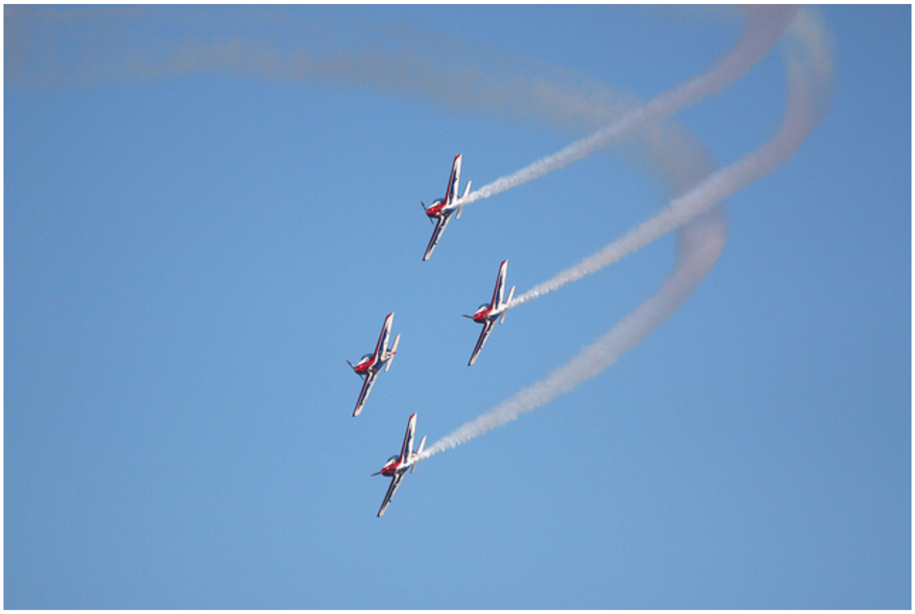


Figure 51 Sample 9

Question: what are flying through the sky?

Top 5 Answer: kites, plane, kite, clouds, airplane

Question: What is the color of background?

Top 5 Answer: blue, red, green, orange, yellow

Question: How many objects in the sky?

Top 5 Answer: 13, 10, 4, 5, 1

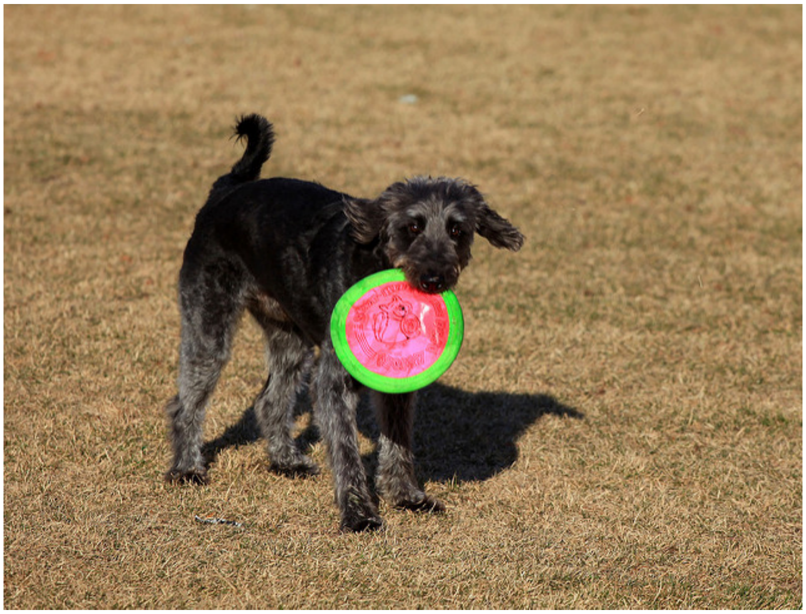


Figure 52 Sample 10

Question: what is the black dog holding?

Top 5 Answer: frisbee, kite, bat, carrot, dog

Question: What is the color of this dog?

Top 5 Answer: black, brown, black and white, white and brown, white

Question: what is the color of the object that the black dog holding?

Top 5 Answer: yellow, blue, white, pink, red

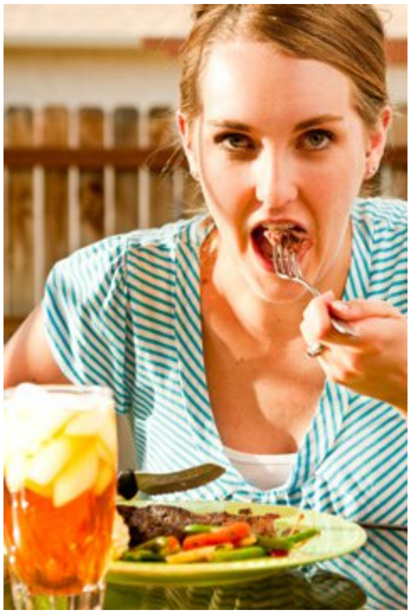


Figure 53 Sample 11

Question: What is this woman doing?

Top 5 Answer: cooking, eating, taking picture, smiling cutting

Question: What is this woman holding?

Top 5 Answer: banana, fork, plate, knife, pizza

Question: what color is this woman's eyes?

Top 5 Answer: blue, brown, black, green, red

5. Analysis of Result

The model we used now get an accuracy of 49.12% over the test set and we present the accuracy of baseline model and the model from UC Berkeley & Sony. The accuracy of model from UC Berkeley & Sony represent the highest accuracy of current Visual Question Model, we can see that the Visual Question Answering is still a hard problem over the world since the accuracy of number-relative question is much low than the accuracy of Yes / No questions.

In our opinion, there are several reasons that the accuracy is not so high. First, the Convolutional Neural Network we use is mainly used to classify the objects, which means that the image features that we extracted contained the class of each object in the image but didn't contain the count of each objects. In this case, the model is to use "common sense" to answer the number-relative questions. Thus, the model has much high accuracy on Yes / No question than the number-relative question.

In some extent, the current model has a high performance in recognizing the type of questions. If the question is asking the number of an object, the top 5 answers will be a number, and if the question is asking about the weather, the top answers will also show some kinds of weathers. In this case, we can say that our current model has high accuracy of answering the question-type but may have low accuracy in answering the question.

We will focus on all the above bottlenecks in the next term and to enhance the whole accuracy of the model.

Conclusion

1. Term Review

When we start our final year project, we know little about neural networks or even about machine learning. Therefore, we hope that we can learn something about machine learning, and then have better understanding on machine learning.

After several weeks of studying and researching, looping back to our works, we think we achieved what we want. Although we lose the direction and consider whether we needed to change the topic of the final year project, we did not give up and kept on doing research on this topic.

At the beginning, we knew nothing about Q&A model or even neural network. We set up a small millstone that we implemented a text-based Q&A first. To achieve this millstone, we read some paper about LSTM, RNN. In this step, the support from Professor Michael R. Lyu and PhDs helped us a lot.

Starting from the text-based Q&A model, we started to find a method can merge the information in image and information in text and use this information to predict the answer. Since the in the text-based Q&A model, the words are represented by vectors, it is obvious that we also can convert images into vectors. There are several ways for converting images into vectors, such as, using CNN to extract feature vectors from images, using R-CNN for find out attention and embedding the captions, which are produced by caption generator, of images to vectors. At the end, we chose the CNN to extract the feature.

Finally, based on these knowledge and technologies, we are able to implement the first version of Visual Question Answering.

2. Shortcoming and further work

2.1 The answer is not accuracy when answering number-related question.

This is the common problem for every Visual Question Answering model using CNN because it is difficult for CNN to find out the number of objects in an image. Therefore, we will deploy the R-CNN into our model to improve the accuracy of answering number-related question. The architecture of R-CNN and the principal of how R-CNN words have already been mentioned above.

2.2 The model cannot fully comprehend some questions when the questions contain logic judgement.

This is the common problem for every Visual Question Answering model because there is no question about logic judgement in training dataset and then the trained model cannot recognize this kind of problems. In term of this shortcoming, we did not find out an efficient solution and we will keep on doing research on this issue.

2.3 The current model can only support English question.

Since English is the most popular language in the world, most of the research is on English. However, as the Chinese is becoming more and more important in the world, we think it is necessary to develop a Visual Question Answering model which is consistent of Chinese. The difficulty on implement a model consistent of Chinese is that we need to embed Chinese characters to vectors and we need Dataset of Chinese Visual Question Answering. After we did some research on this topic, we found that, in term of dataset, Baidu provides a dataset for Chinese Visual Question Answering and, in term of

CONCLUSION

Chinese character embedding, Word2vec is sufficient for embedding Chinese which is the same as embedding Latin characters.

Acknowledgement

We would like to express our gratitude to our supervisor Prof. Michael Lyu for providing a lot suggestions and opinions. Secondly, we would also like to thank Mr. Edward Yau for his technical support. Last but not least, we would like to express our special thanks to Li Jian, Su Yuxin, Zeng Jichuan who gave us a lot of ideas to do this wonderful project. Without their help, we would encounter more difficulties in this project.

Reference

27 8 2015. 29 11 2016. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

n.d. 29 11 2016. <http://www.visualqa.org>.

n.d. <https://gist.github.com/baraldilorenzo/07d7802847aaad0a35d3>.

n.d. <http://tutorial.caffe.berkeleyvision.org>.

n.d. <https://upload.wikimedia.org/wikipedia/commons/thumb/4/45/Components_stress_tensor.svg/300px-Components_stress_tensor.svg.png>.

n.d. <http://image.slidesharecdn.com/april131700googleramanathan-160425195435/95/machine-intelligence-at-google-scale-tensorflow-13-638.jpg?cb=1461614089>.

n.d. <http://download.tensorflow.org/paper/whitepaper2015.pdf>.

n.d. <https://zh.wikipedia.org/wiki/图像处理>.

REFERENCE

n.d. <https://www.cs.toronto.edu/~frossard/post/vgg16/vgg16.png>.

AlphaGo. n.d. Wikipedia. 29 11 2016. <https://en.wikipedia.org/wiki/AlphaGo>.

Girshick, Ross. "Fast R-CNN." (n.d.).

GitHub. n.d. <https://github.com/paarthneekhara/neural-vqa-tensorflow>.

K. Simonyan, A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." (n.d.).

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." (n.d.).

Mikolov, Tomas, et al. "Distributed Representations of Words and Phrases and their Compositionality." (n.d.).

"Mind’s Eye: A Recurrent Visual Representation for Image Caption Generation." (n.d.).

O'Shea, Keiron Teilo and Ryan Nash. "An Introduction to Convolutional Neural Networks." (n.d.).

REFERENCE

"Show and Tell: A Neural Image Caption Generator." (n.d.).

Sukhbaatar, Sainbayar, et al. "End-To-End Memory Networks." (n.d.).

TensorFlow. n.d. 29 11 2016. <https://www.tensorflow.org>.