# Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding

Akira Fukui et al., 2016

Presented by:

Mohammad Eshghi

January 31, 2020

# Overview

# Introduction: VQA

Visual question answering: "What is the woman feeding the giraffe?" Correct answer: "Carrot"

Credit: *medium post*

# Introduction: VG

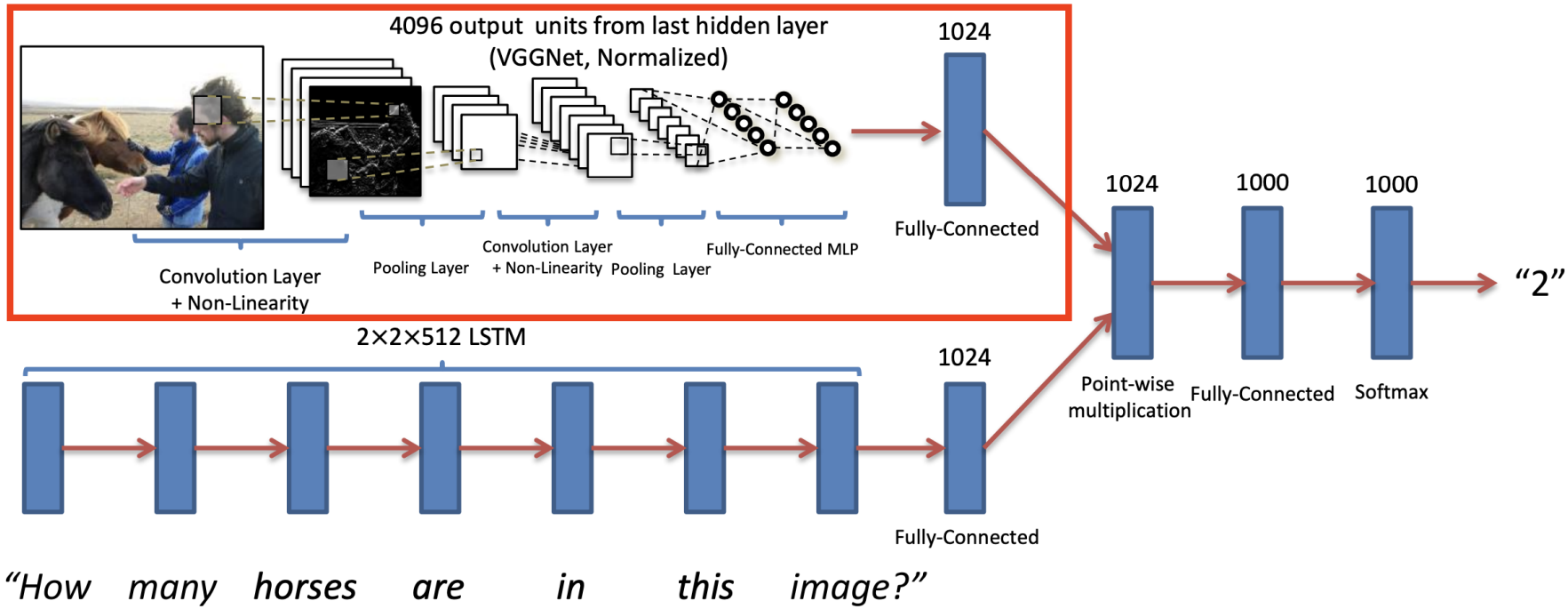
Visual grounding: "A tattooed woman (red) with a green dress (blue) and yellow backpack (green) holding a water bottle (pink) is walking across the street." The bounding boxes drawn would be excellent answers to this task.

Credit: *medium post*

# Introduction

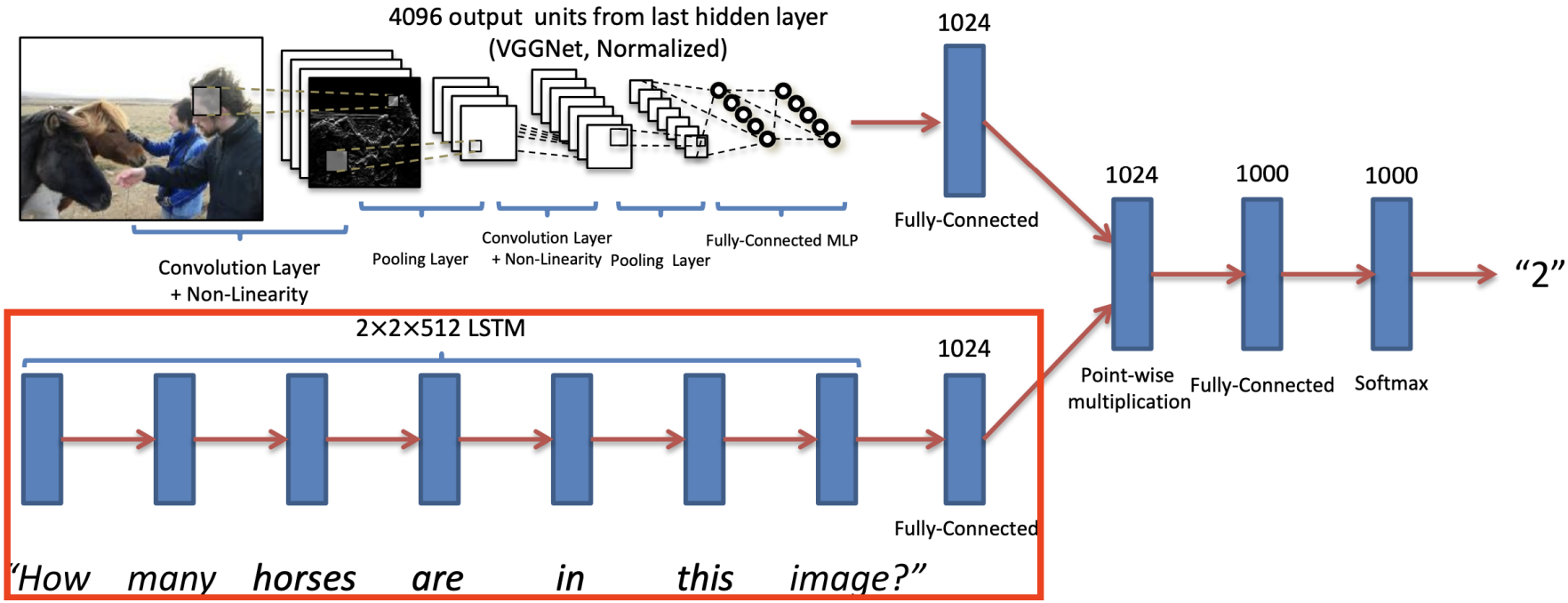
$$\hat{a} = \underset{a \in A}{\operatorname{argmax}}\, p(a|\mathbf{x}, \mathbf{q}; \theta)$$

# Image encoding



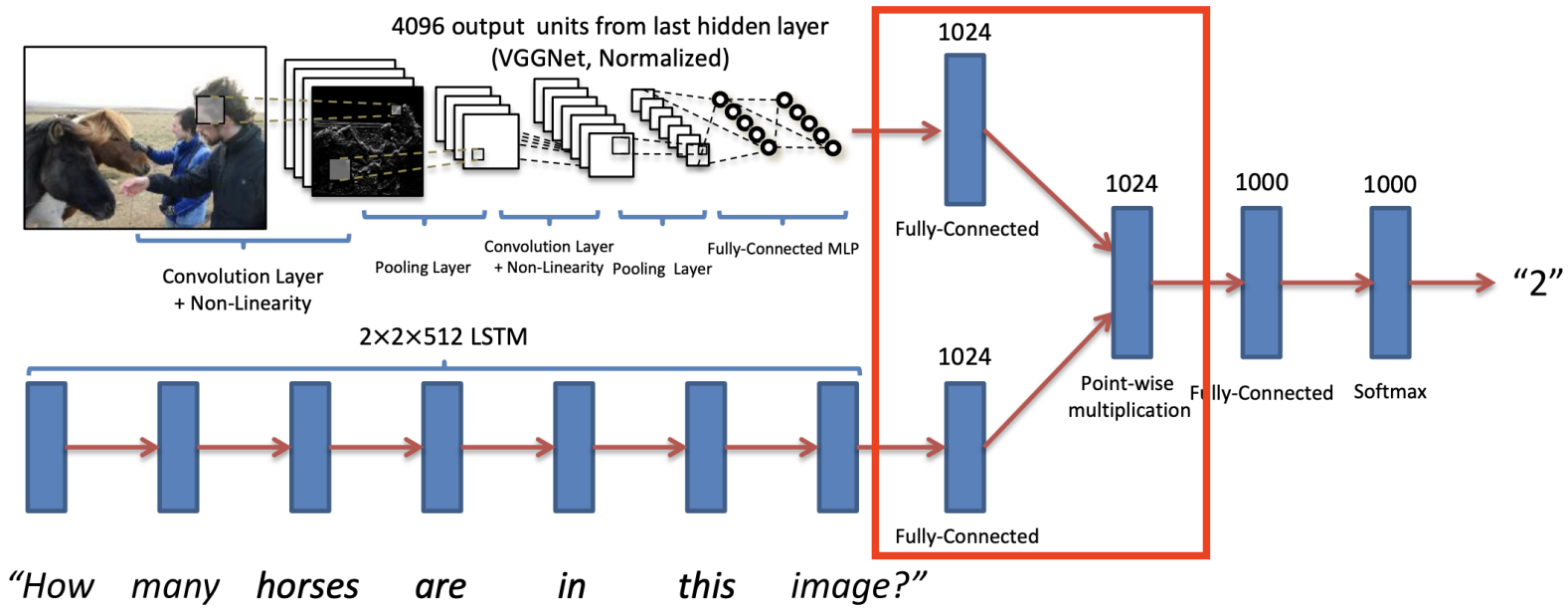

Credit: *Agrawal et al., 2019*

# Text encoding



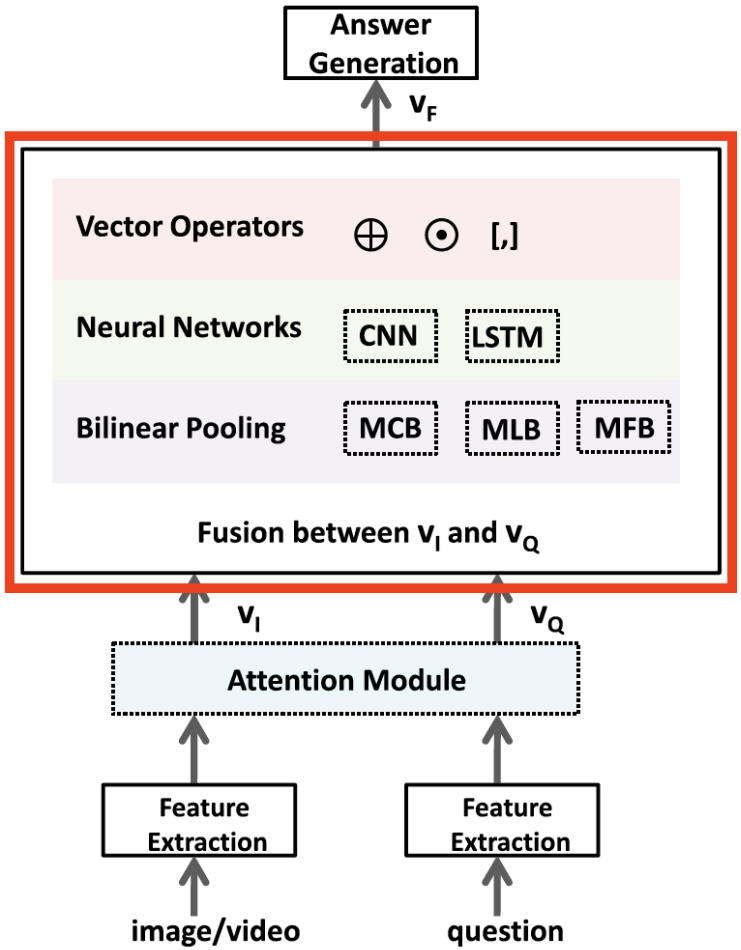

Credit: *Agrawal et al., 2019*

# Information fusion




Credit: *Agrawal et al., 2019*

# Information fusion




Credit: *Zhang et al., 2019*

# What is MCB?

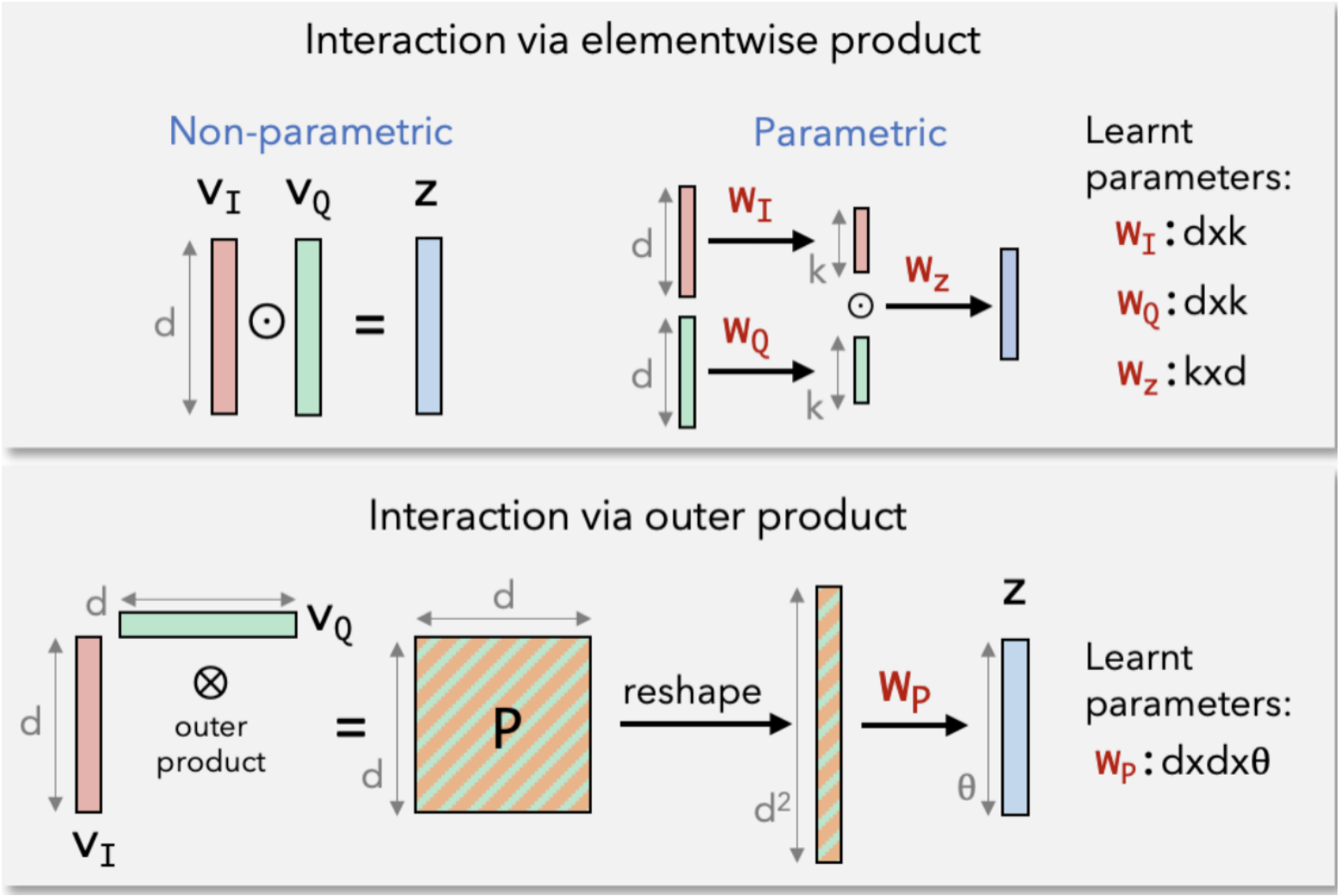
**Multimodal Pooling** is combining the two vector representations of image and question (i.e. Multimodal), respectively. This creates a joint representation of the two vectors.

**Bilinear Pooling** means the outer product of the two input vectors

- multiply every element of one vector of length N by every element of the other vector of length N, resulting in a matrix of size NxN.

**Compact** bilinear pooling means reducing the dimensionality to get almost the same level of power with way fewer parameters.

# What is MCB?

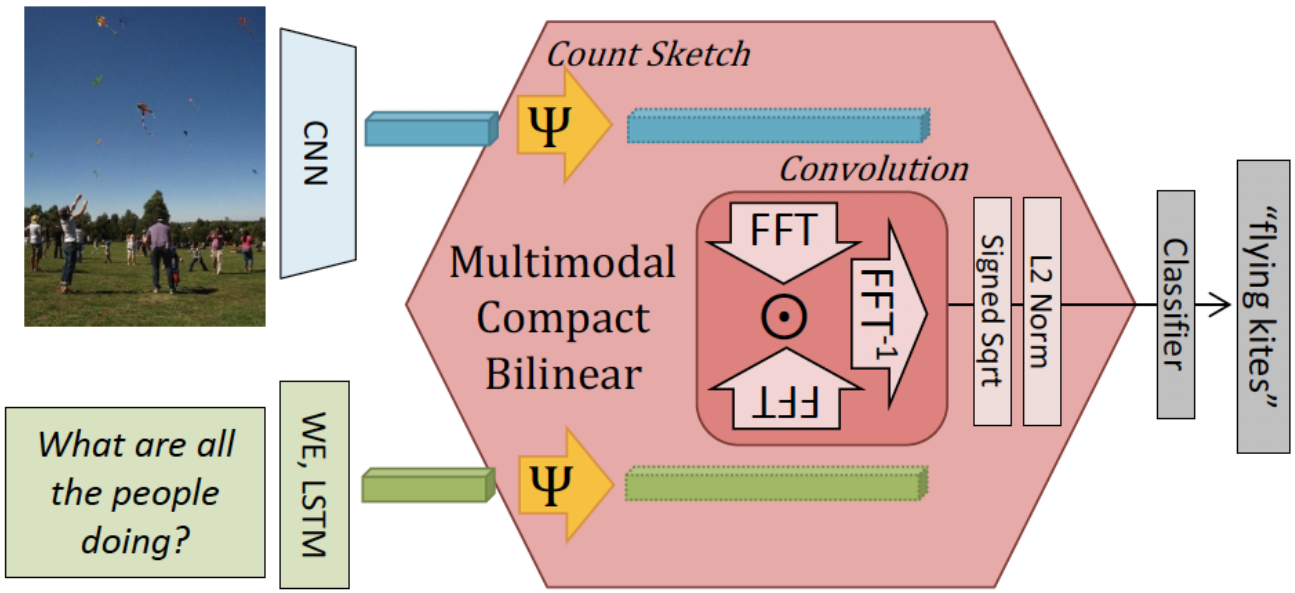



Credit: *The lure of the outer product*

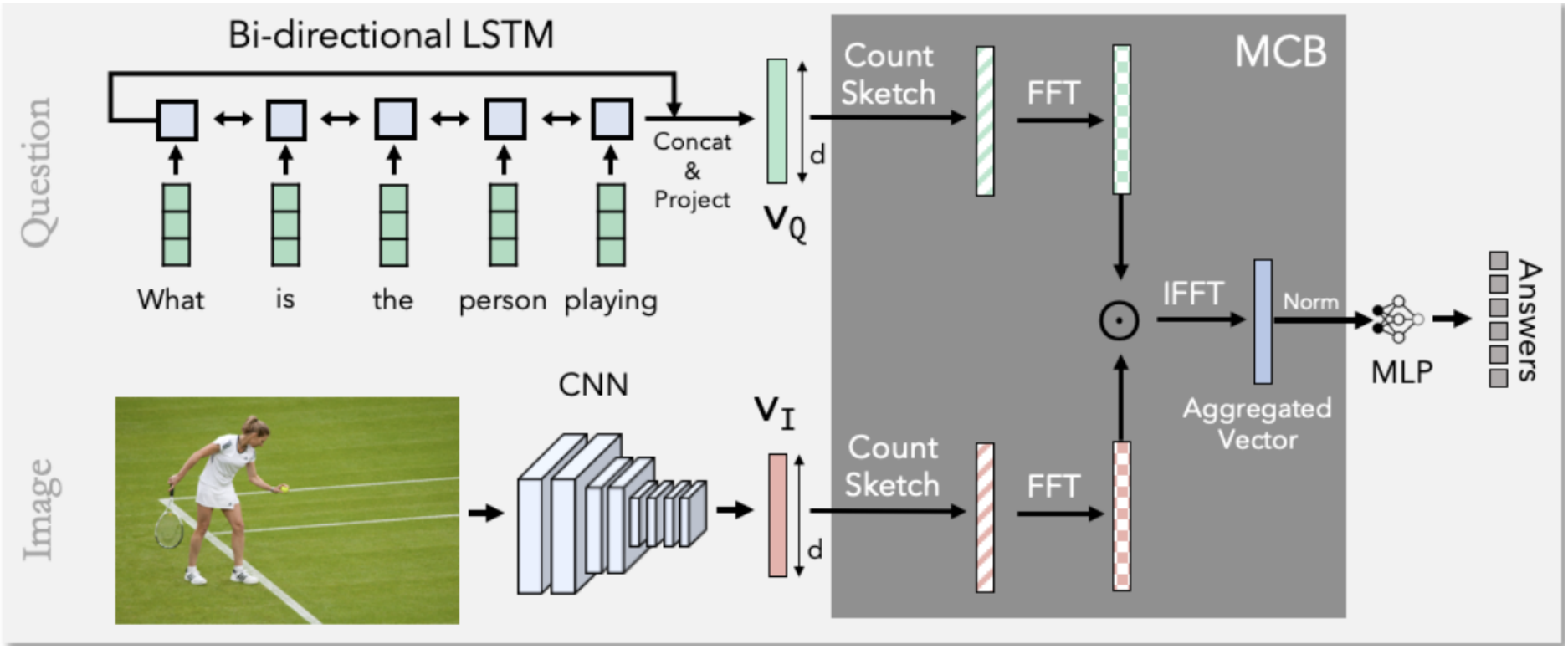
# Why MCB?

- Vector operations are too simplistic to fully capture the relationships between images and text
- So many non-linear neural networks to learn the interaction between the visual and textual features
- Research like Zhang et al., 2019 confirm that Bilinear Pooling is the promising road to take for information fusion
- Bilinear pooling is a very rich representation that allows a full multiplicative interaction between all elements of both vectors capturing any possible relationship
- However, in bilinear pooling the number of resulting parameters is too high to be practical. For example,
  - if the image and text vectors are of length 2048
  - the resulting matrix would have $2048^2$ elements
  - and we'd need to fully connect that matrix to 3000 classes
  - it results in 12.5 billion learnable parameters
- Therefore, introducing MCB to capture the discriminating abilities of bilinear pooling with only a a few thousand parameters (16k)

# How to MCB?


CNN
What are all the people doing?
WE, LSTM
Count Sketch
Ψ
Ψ
Convolution
Multimodal Compact Bilinear
FFT
FFT
FFT-1
Signed Sqrt
L2 Norm
Classifier
"flying kites"

# How to MCB?




Credit: *The lure of the outer product*

# How to MCB?

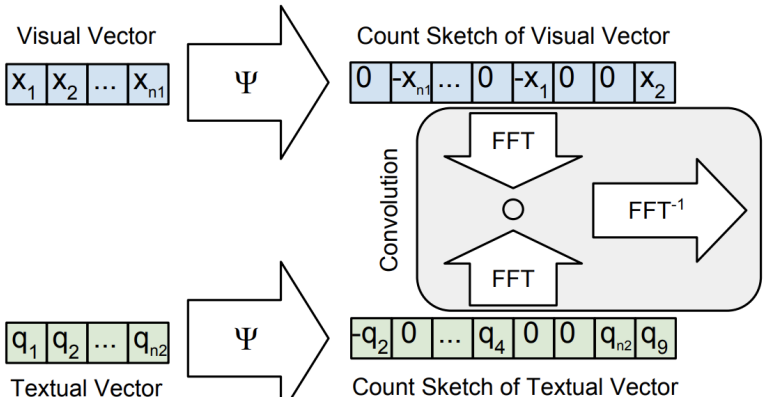
MCB is approximated by

- randomly projecting the image and text representations to a higher dimensional space using Count Sketch
- and then convolving both vectors efficiently by using element-wise product in FFT space
- $z = w[x \otimes q]$
- about 12.5 billion parameters for 2048 images and texts and 3000 classes!

MCB key feature

- using Count Sketch over outer product to use convolution, and FFT
  - projecting the outer product to a lower dimensional space
  - avoiding computing the outer product directly:
    - $\Psi(x \otimes q, h, s) = \Psi(x, h, s) * \Psi(q, h, s)$

# How to MCB?



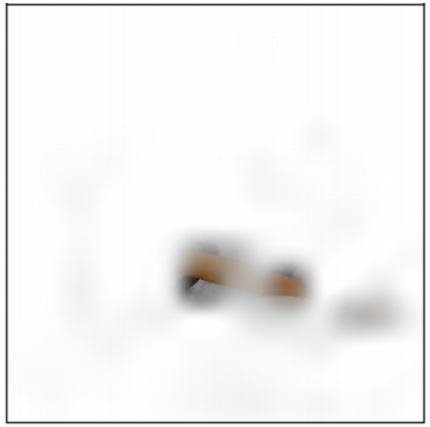

**Algorithm 1** Multimodal Compact Bilinear

input: $v_1 \in \mathbb{R}^{n_1}, v_2 \in \mathbb{R}^{n_2}$
output: $\Phi(v_1, v_2) \in \mathbb{R}^d$
**procedure** MCB$(v_1, v_2, n_1, n_2, d)$
    **for** $k \leftarrow 1 \ldots 2$ **do**
        **if** $h_k, s_k$ not initialized **then**
            **for** $i \leftarrow 1 \ldots n_k$ **do**
                sample $h_k[i]$ from $\{1, \ldots, d\}$
                sample $s_k[i]$ from $\{-1, 1\}$
        $v'_k = \Psi(v_k, h_k, s_k, n_k)$
    $\Phi = \text{FFT}^{-1}(\text{FFT}(v'_1) \odot \text{FFT}(v'_2))$
    **return** $\Phi$
**procedure** $\Psi(v, h, s, n)$
    $y = [0, \ldots, 0]$
    **for** $i \leftarrow 1 \ldots n$ **do**
        $y[h[i]] = y[h[i]] + s[i] \cdot v[i]$
    **return** $y$

# Attention models

Attention models use an attention score between the question and each region in the image. The resulting attention vector helps the model to focus on the most relevant region(s) in the image to answer the given question.

# How to MCB?

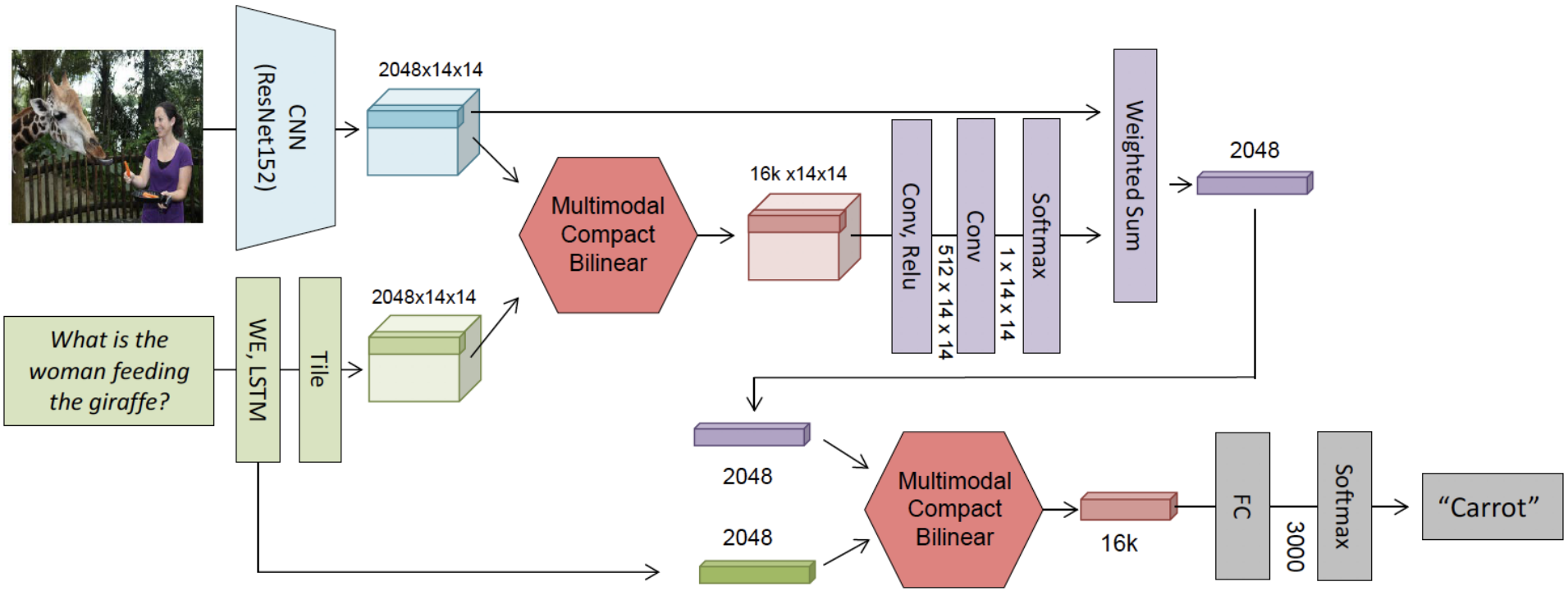


What vegetable is the dog chewing on?
MCB: carrot
GT: carrot

What kind of dog is this?
MCB: husky
GT: husky

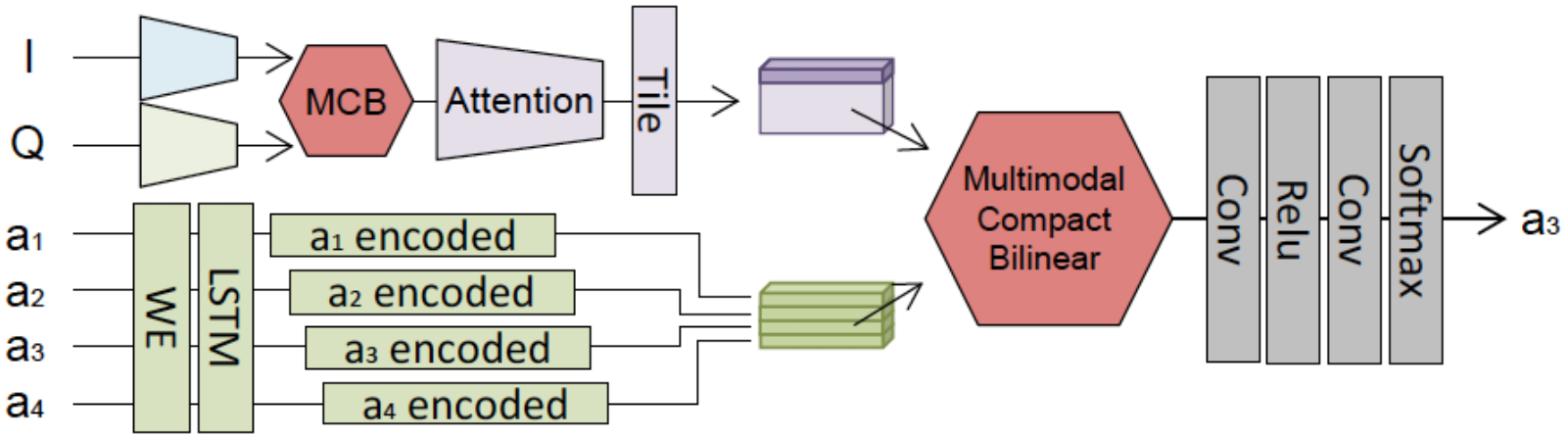
What kind of flooring does the room have?
MCB: carpet
GT: carpet

# How to MCB?


CNN (ResNet152)
2048x14x14
Multimodal Compact Bilinear
16k x14x14
Conv, Relu
512 x 14 x 14
Conv
1 x 14 x 14
Softmax
Weighted Sum
2048
What is the woman feeding the giraffe?
WE, LSTM
Tile
2048x14x14
2048
2048
Multimodal Compact Bilinear
16k
FC
3000
Softmax
"Carrot"

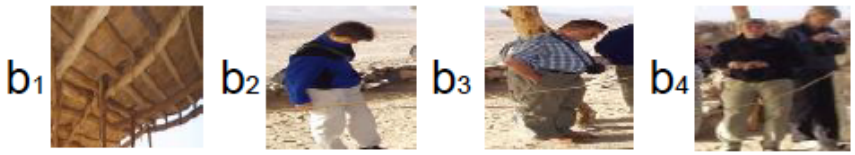
# Architecture of MCB

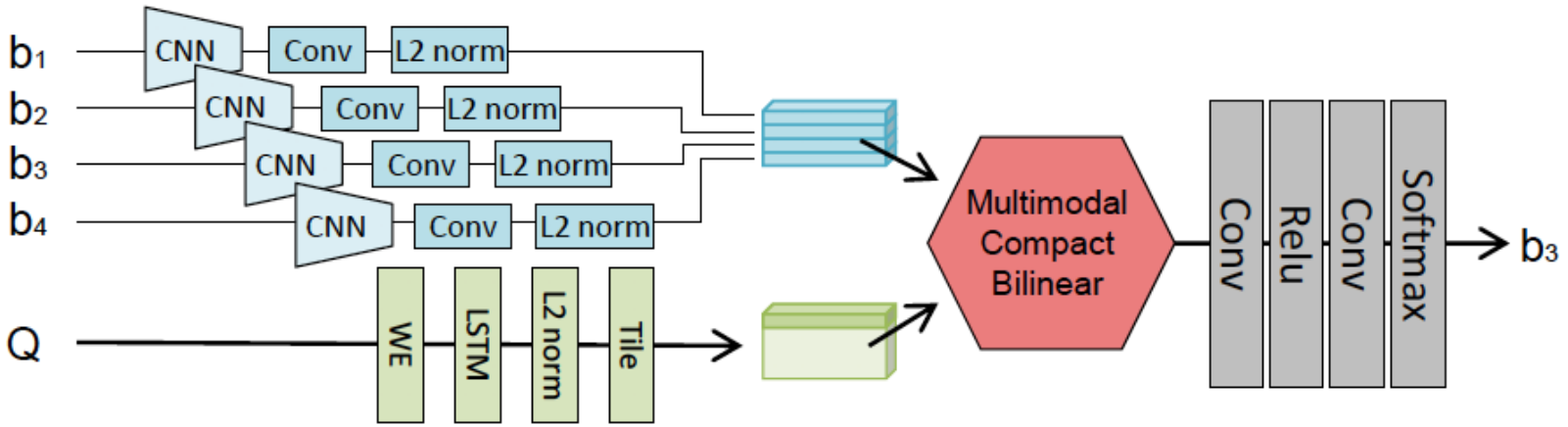
Q : “What do you see?” (Ground Truth : $a_3$)
$a_1$ : “A courtyard with flowers”
$a_2$ : “A restaurant kitchen”
$a_3$ : “A family with a stroller, tables for dining”
$a_4$ : “People waiting on a train”



# Architecture of MCB

Q: “Person in blue checkered shirt”



# Datasets

For Visual Question Answering

1. MSCOCO
2. Visual Genome
3. Visual7W

For Visual Grounding

1. Flickr30k
2. ReflectItGame

# VQA

| Method | Accuracy |
|---|---|
| Element-wise Sum | 56.50 |
| Concatenation | 57.49 |
| Concatenation + FC | 58.40 |
| Concatenation + FC + FC | 57.10 |
| Element-wise Product | 58.57 |
| Element-wise Product + FC | 56.44 |
| Element-wise Product + FC + FC | 57.88 |
| MCB ($2048 \times 2048 \rightarrow 16\text{K}$) | **59.83** |
| Full Bilinear ($128 \times 128 \rightarrow 16\text{K}$) | 58.46 |
| MCB ($128 \times 128 \rightarrow 4\text{K}$) | 58.69 |
| Element-wise Product with VGG-19 | 55.97 |
| MCB ($d = 16\text{K}$) with VGG-19 | **57.05** |
| Concatenation + FC with Attention | 58.36 |
| MCB ($d = 16\text{K}$) with Attention | **62.50** |

| **Compact Bilinear $d$** | **Accuracy** |
|---|---|
| 1024 | 58.38 |
| 2048 | 58.80 |
| 4096 | 59.42 |
| 8192 | 59.69 |
| 16000 | **59.83** |
| 32000 | 59.71 |

# VQA

| | Test-dev | | | | | Test-standard | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Open Ended | | | | MC | Open Ended | | | | MC |
| | Y/N | No. | Other | All | All | Y/N | No. | Other | All | All |
| MCB | 81.2 | 35.1 | 49.3 | 60.8 | 65.4 | - | - | - | - | - |
| MCB + Genome | 81.7 | 36.6 | 51.5 | 62.3 | 66.4 | - | - | - | - | - |
| MCB + Att. | 82.2 | 37.7 | 54.8 | 64.2 | 68.6 | - | - | - | - | - |
| MCB + Att. + GloVe | 82.5 | 37.6 | 55.6 | 64.7 | 69.1 | - | - | - | - | - |
| MCB + Att. + Genome | 81.7 | 38.2 | 57.0 | 65.1 | 69.5 | - | - | - | - | - |
| MCB + Att. + GloVe + Genome | 82.3 | 37.2 | 57.4 | 65.4 | 69.9 | - | - | - | - | - |
| Ensemble of 7 Att. models | **83.4** | **39.8** | **58.5** | **66.7** | **70.2** | **83.2** | **39.5** | **58.0** | **66.5** | **70.1** |
| Naver Labs (challenge 2nd) | 83.5 | 39.8 | 54.8 | 64.9 | 69.4 | 83.3 | 38.7 | 54.6 | 64.8 | 69.3 |
| HieCoAtt (Lu et al., 2016) | 79.7 | 38.7 | 51.7 | 61.8 | 65.8 | - | - | - | 62.1 | 66.1 |
| DMN+ (Xiong et al., 2016) | 80.5 | 36.8 | 48.3 | 60.3 | - | - | - | - | 60.4 | - |
| FDA (Ilievski et al., 2016) | 81.1 | 36.2 | 45.8 | 59.2 | - | - | - | - | 59.5 | - |
| D-NMN (Andreas et al., 2016a) | 81.1 | 38.6 | 45.5 | 59.4 | - | - | - | - | 59.4 | - |
| AMA (Wu et al., 2016) | 81.0 | 38.4 | 45.2 | 59.2 | - | 81.1 | 37.1 | 45.8 | 59.4 | - |
| SAN (Yang et al., 2015) | 79.3 | 36.6 | 46.1 | 58.7 | - | - | - | - | 58.9 | - |
| NMN (Andreas et al., 2016b) | 81.2 | 38.0 | 44.0 | 58.6 | - | 81.2 | 37.7 | 44.0 | 58.7 | - |
| AYN (Malinowski et al., 2016) | 78.4 | 36.4 | 46.3 | 58.4 | - | 78.2 | 36.3 | 46.3 | 58.4 | - |
| SMem (Xu and Saenko, 2016) | 80.9 | 37.3 | 43.1 | 58.0 | - | 80.9 | 37.5 | 43.5 | 58.2 | - |
| VQA team (Antol et al., 2015) | 80.5 | 36.8 | 43.1 | 57.8 | 62.7 | 80.6 | 36.5 | 43.7 | 58.2 | 63.1 |
| DPPnet (Noh et al., 2015) | 80.7 | 37.2 | 41.7 | 57.2 | - | 80.3 | 36.9 | 42.2 | 57.4 | - |
| iBOWIMG (Zhou et al., 2015) | 76.5 | 35.0 | 42.6 | 55.7 | - | 76.8 | 35.0 | 42.6 | 55.9 | 62.0 |

# VQA

| Method | Accuracy, % |
|---|---|
| Plummer et al. (2015) | 27.42 |
| Hu et al. (2016b) | 27.80 |
| Plummer et al. (2016)[1] | 43.84 |
| Wang et al. (2016) | 43.89 |
| Rohrbach et al. (2016) | 47.81 |
| Concatenation | 46.50 |
| Element-wise Product | 47.41 |
| Element-wise Product + Conv | 47.86 |
| MCB | **48.69** |

# VQA

| Method | Accuracy, % |
|---|---|
| Hu et al. (2016b) | 17.93 |
| Rohrbach et al. (2016) | 26.93 |
| Concatenation | 25.48 |
| Element-wise Product | 27.80 |
| Element-wise Product + Conv | 27.98 |
| MCB | **28.91** |

# Visual Grounding

| **Method** | What | Where | When | Who | Why | How | Avg |
|---|---|---|---|---|---|---|---|
| Zhu et al. | 51.5 | 57.0 | 75.0 | 59.5 | 55.5 | 49.8 | 54.3 |
| Concat+Att. | 47.8 | 56.9 | 74.1 | 62.3 | 52.7 | **51.2** | 52.8 |
| MCB+Att. | **60.3** | **70.4** | **79.5** | **69.2** | **58.2** | 51.1 | **62.2** |

# Visual Grounding

What vegetable is the dog chewing on?
MCB: carrot
GT: carrot

What kind of dog is this?
MCB: husky
GT: husky

What kind of flooring does the room have?
MCB: carpet
GT: carpet

What color is the traffic light?
MCB: green
GT: green

Is this an urban area?
MCB: yes
GT: yes

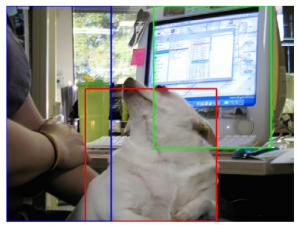
Where are the buildings?
MCB: in background
GT: on left

MCB

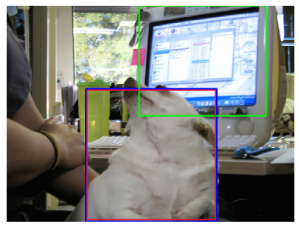
Eltwise Product + Conv

A tattooed woman with a green dress and yellow backpack holding a water bottle is walking across the street.

MCB



Eltwise Product + Conv



A dog distracts his owner from working at her computer.

# Conclusion

- At the heart of MCB is the Count Sketch function
- Count Sketch function actually limits MCB critically:
  - Hash collisions result in errors caused by the Count Sketch operation
  - To lower the probability of a collision, we should choose a larger counter array,
  - For MCB to be effective, the length of the Count Sketch vector needs to be very large ($\sim 16k$).
- The code for the paper is available in github

# Thank You