Grid LSTM

Kalchbrenner et al. (Google DeepMind)

Joost Bastings

Institute for Logic, Language and Computation

March 10, 2016

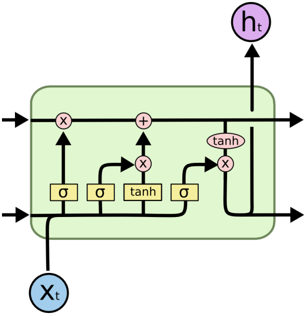
Outline

Outline

LSTM

Long Short-Term Memory (LSTM) networks have *gates* that control access to memory cells

(Hochreiter and Schmidhuber, 1997)



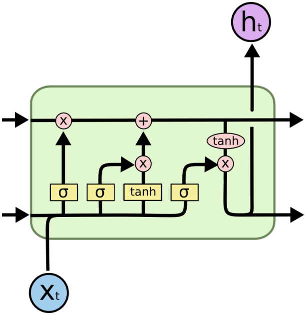
©Christopher Olah

LSTM

Long Short-Term Memory (LSTM) networks have *gates* that control access to memory cells

(Hochreiter and Schmidhuber, 1997)

- Preserve signals, propagate errors for *much longer*



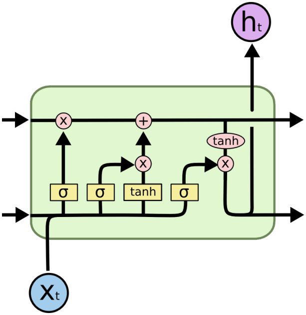
ⓒChristopher Olah

LSTM

Long Short-Term Memory (LSTM) networks have *gates* that control access to memory cells

(Hochreiter and Schmidhuber, 1997)

- Preserve signals, propagate errors for *much longer*
- Gates can learn to attend to specific parts of the input signals (and ignore others)

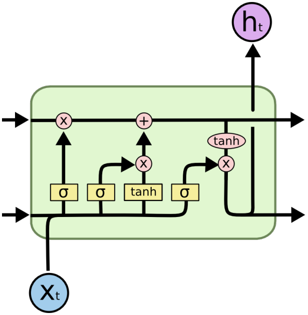


LSTM

Long Short-Term Memory (LSTM) networks have *gates* that control access to memory cells

(Hochreiter and Schmidhuber, 1997)

- Preserve signals, propagate errors for *much longer*
- Gates can learn to attend to specific parts of the input signals (and ignore others)



©Christopher Olah

These properties make LSTMs good at speech recognition, hand-writing recognition, machine translation, etc.

Motivation

- Computer vision success: **deep networks are key** to finding complex patterns

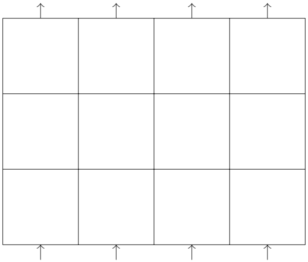
Motivation

- Computer vision success: **deep networks are key** to finding complex patterns
- However, deep networks also suffer from the **vanishing gradient** problem!

Motivation

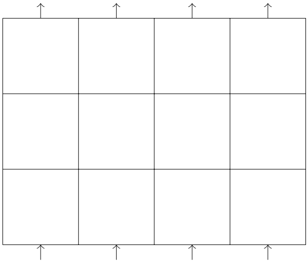
- Computer vision success: **deep networks are key** to finding complex patterns
- However, deep networks also suffer from the **vanishing gradient** problem!
- This is the motivation to generalise the advantages of LSTMs to deep computation

Idea: Grid LSTM



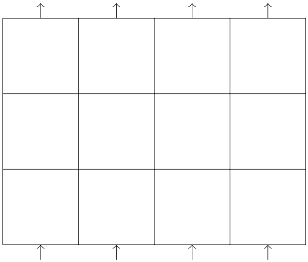
- A Grid LSTM (Kalchbrenner et al., 2015) is a network arranged in a grid of 1 or more dimensions

Idea: Grid LSTM



- A Grid LSTM (Kalchbrenner et al., 2015) is a network arranged in a grid of 1 or more dimensions
- LSTM cells in ‘any or all’ dimensions of the grid

Idea: Grid LSTM



- A Grid LSTM (Kalchbrenner et al., 2015) is a network arranged in a grid of 1 or more dimensions
- LSTM cells in 'any or all' dimensions of the grid
- Short-hand: N-dimensional Grid LSTM = N-LSTM

Outline

LSTM

- An LSTM processes input and target pairs

$$(x_1, y_1), \ldots, (x_m, y_m)$$

LSTM

- An LSTM processes input and target pairs

$$(x_1, y_1), \ldots, (x_m, y_m)$$

- Past inputs $x_1, \ldots x_{i-1}$ determine the state of the network:

$$\text{hidden} \quad \mathbf{h} \in \mathbb{R}^d$$
$$\text{memory} \, \mathbf{m} \in \mathbb{R}^d$$

LSTM

- An LSTM processes input and target pairs

$$(x_1, y_1), \ldots, (x_m, y_m)$$

- Past inputs $x_1, \ldots x_{i-1}$ determine the state of the network:

$$\text{hidden} \quad \mathbf{h} \in \mathbb{R}^d$$
$$\text{memory} \, \mathbf{m} \in \mathbb{R}^d$$

- Let $\mathbf{H} = \begin{bmatrix} I x_i \\ \mathbf{h} \end{bmatrix}$, where I is a projection matrix transforming x_i

LSTM

At each step, calculate:

1. Gates:

$$
\begin{aligned}
\mathbf{g}^u &= \sigma(\mathbf{W}^u\mathbf{H}) && \text{update} \\
\mathbf{g}^f &= \sigma(\mathbf{W}^f\mathbf{H}) && \text{forget} \\
\mathbf{g}^o &= \sigma(\mathbf{W}^o\mathbf{H}) && \text{output}
\end{aligned}
$$

LSTM

At each step, calculate:

1. Gates:

$$\begin{aligned}
\mathbf{g}^u &= \sigma(\mathbf{W}^u\mathbf{H}) && \text{update} \\
\mathbf{g}^f &= \sigma(\mathbf{W}^f\mathbf{H}) && \text{forget} \\
\mathbf{g}^o &= \sigma(\mathbf{W}^o\mathbf{H}) && \text{output} \\
\mathbf{g}^c &= \tanh(\mathbf{W}^c\mathbf{H}) && \text{content}
\end{aligned}$$

LSTM

At each step, calculate:

1. Gates:

$$
\begin{aligned}
\mathbf{g}^u &= \sigma(\mathbf{W}^u\mathbf{H}) && \text{update} \\
\mathbf{g}^f &= \sigma(\mathbf{W}^f\mathbf{H}) && \text{forget} \\
\mathbf{g}^o &= \sigma(\mathbf{W}^o\mathbf{H}) && \text{output} \\
\mathbf{g}^c &= \tanh(\mathbf{W}^c\mathbf{H}) && \text{content}
\end{aligned}
$$

2. New memory:

$$\mathbf{m}' = \mathbf{g}^f \odot \mathbf{m} + \mathbf{g}^u \odot \mathbf{g}^c$$

LSTM

At each step, calculate:

1. Gates:

$$\begin{aligned} \mathbf{g}^u &= \sigma(\mathbf{W}^u\mathbf{H}) && \text{update} \\ \mathbf{g}^f &= \sigma(\mathbf{W}^f\mathbf{H}) && \text{forget} \\ \mathbf{g}^o &= \sigma(\mathbf{W}^o\mathbf{H}) && \text{output} \\ \mathbf{g}^c &= \tanh(\mathbf{W}^c\mathbf{H}) && \text{content} \end{aligned}$$

2. New memory:

$$\mathbf{m}' = \mathbf{g}^f \odot \mathbf{m} + \mathbf{g}^u \odot \mathbf{g}^c$$

LSTM

At each step, calculate:

1. Gates:

$$\begin{aligned} \mathbf{g}^u &= \sigma(\mathbf{W}^u\mathbf{H}) && \text{update} \\ \mathbf{g}^f &= \sigma(\mathbf{W}^f\mathbf{H}) && \text{forget} \\ \mathbf{g}^o &= \sigma(\mathbf{W}^o\mathbf{H}) && \text{output} \\ \mathbf{g}^c &= \tanh(\mathbf{W}^c\mathbf{H}) && \text{content} \end{aligned}$$

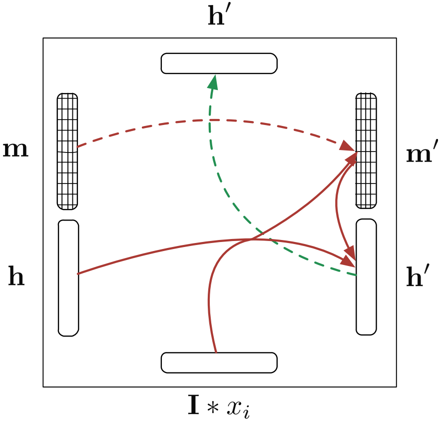
2. New memory:

$$\mathbf{m}' = \mathbf{g}^f \odot \mathbf{m} + \mathbf{g}^u \odot \mathbf{g}^c$$

3. New state:

$$\mathbf{h}' = \tanh(\mathbf{g}^o \odot \mathbf{m}')$$

LSTM



Standard LSTM block

Outline

Grid LSTM

An N-LSTM block receives as input:

- N hidden vectors $\mathbf{h}_1, \ldots, \mathbf{h}_N$
- N memory vectors $\mathbf{m}_1, \ldots, \mathbf{m}_N$

Grid LSTM

An N-LSTM block receives as input:

- N hidden vectors $\mathbf{h}_1, \ldots, \mathbf{h}_N$
- N memory vectors $\mathbf{m}_1, \ldots, \mathbf{m}_N$

We concatenate all hidden vectors into a **shared input vector**:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_N \end{bmatrix}$$

Grid LSTM

An N-LSTM block receives as input:

- N hidden vectors $\mathbf{h}_1, \ldots, \mathbf{h}_N$
- N memory vectors $\mathbf{m}_1, \ldots, \mathbf{m}_N$

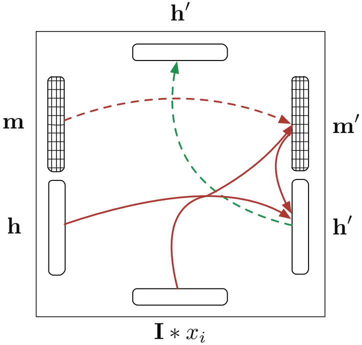
We concatenate all hidden vectors into a **shared input vector**:

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_N \end{bmatrix}$$

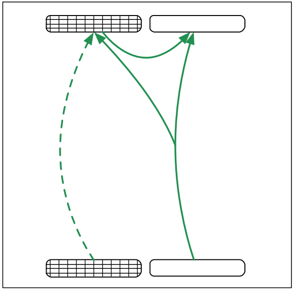
And then calculate N transforms:

$$\begin{aligned} (\mathbf{h}_1^{'}, \mathbf{m}_1^{'}) &= \mathrm{LSTM}(\mathbf{H}, \mathbf{m}_1, \mathbf{W}_1) \\ &\vdots \\ (\mathbf{h}_N^{'}, \mathbf{m}_N^{'}) &= \mathrm{LSTM}(\mathbf{H}, \mathbf{m}_N, \mathbf{W}_N) \end{aligned}$$

1D Grid LSTM

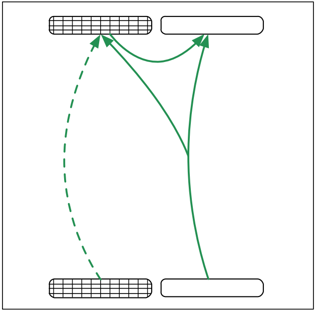


Standard LSTM block

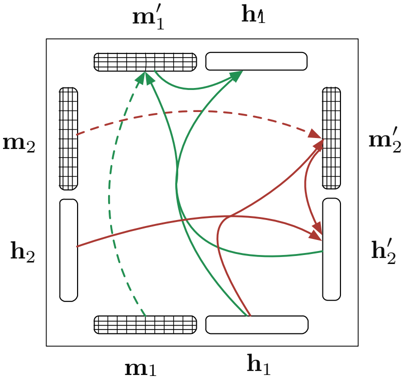


1d Grid LSTM Block

1D and 2D

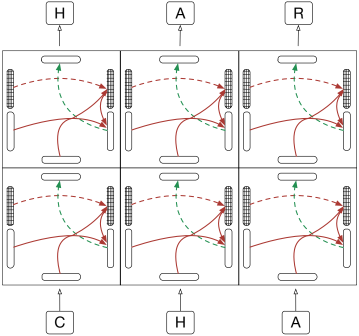


1d Grid LSTM Block

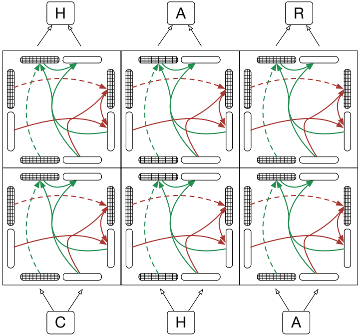


2d Grid LSTM block

Stacked LSTM vs 2-LSTM

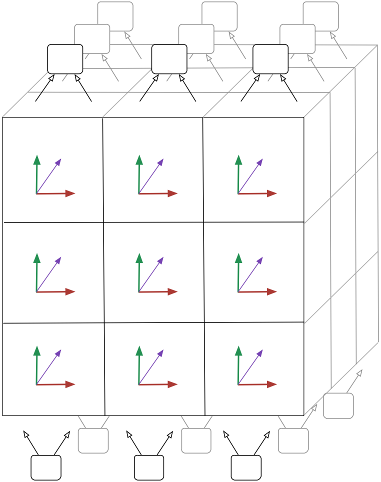


Stacked LSTM



2d Grid LSTM

3D



3d Grid LSTM

Grid LSTM – Notes

- Input is projected along the edge(s), see previous slide: character 'C' initializes $\mathbf{h}_1$ and $\mathbf{m}_1$

Grid LSTM – Notes

- Input is projected along the edge(s), see previous slide: character 'C' initializes $\mathbf{h}_1$ and $\mathbf{m}_1$
- Predictions based on both the state *and* memory of edge cells

Grid LSTM – Notes

- Input is projected along the edge(s), see previous slide: character ‘C’ initializes $\mathbf{h}_1$ and $\mathbf{m}_1$
- Predictions based on both the state *and* memory of edge cells
- It is possible to **share weights** along any dimension

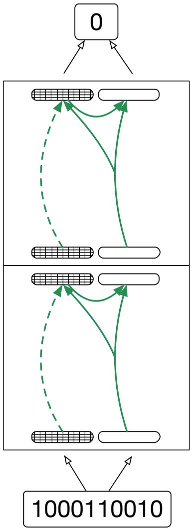
Grid LSTM – Notes

- Input is projected along the edge(s), see previous slide: character ‘C’ initializes $\mathbf{h}_1$ and $\mathbf{m}_1$
- Predictions based on both the state *and* memory of edge cells
- It is possible to **share weights** along any dimension
- If weights are shared along **all dimensions**: *Tied N-LSTM*

Outline

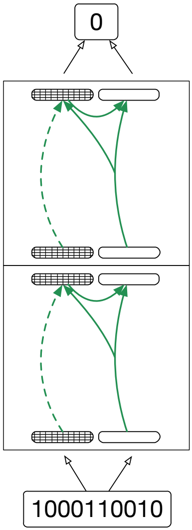
Parity

- Given k input bits, output 0 iff sum is even, else 1



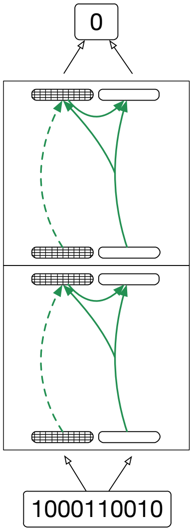
Parity

- Given k input bits, output 0 iff sum is even, else 1
- Parity is really hard, because changing one input bit changes the target

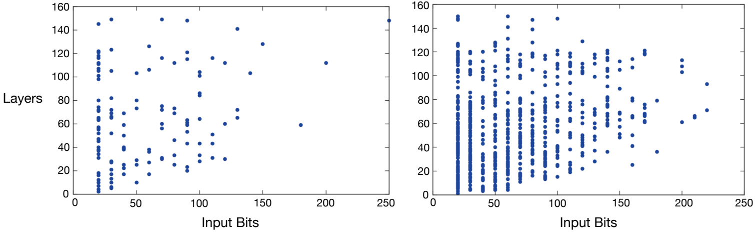


Parity

- Given k input bits, output 0 iff sum is even, else 1
- Parity is really hard, because changing one input bit changes the target
- All input at the same time (why?)



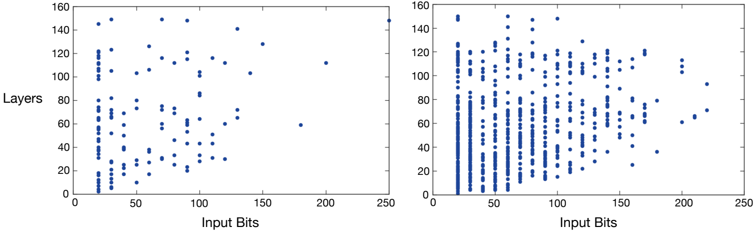
Parity



Left: #hidden = 500, Right: #hidden = 1500

Dot: 100% accuracy on k-bit input

Parity



Left: #hidden = 500, Right: #hidden = 1500
Dot: 100% accuracy on k-bit input

	Layers	Hidden	k
Tied tanh FFN	5	1500	30
Tied ReLU FFN	4	1500	30
Tied 1-LSTM	72	1500	220
Tied 1-LSTM	148	500	250

Addition

Task: sum two 15-digit integers

– 1 2 3 – 8 9 9 – – – – –

– – – – – – – – 1 0 2 2 –

Addition

Task: sum two 15-digit integers

– 1 2 3 – 8 9 9 – – – – –
– – – – – – – – 1 0 2 2 –

	Layers	Samples	Accuracy
Stacked LSTM	1	5M	51%
Untied 2-LSTM	5	5M	67%
Tied 2-LSTM	18	0.55M	>99%

Addition

Task: sum two 15-digit integers

– 1 2 3 – 8 9 9 – – – – –
– – – – – – – – 1 0 2 2 –

	Layers	Samples	Accuracy
Stacked LSTM	1	5M	51%
Untied 2-LSTM	5	5M	67%
Tied 2-LSTM	18	0.55M	>99%

- Trained up to 5M samples or until accuracy 100%
- *Tied* better because of the repetitive nature of the task
- Grid LSTM has advantage by tackling vanishing gradient

Memorization

Task: memorize random sequence of 20 symbols

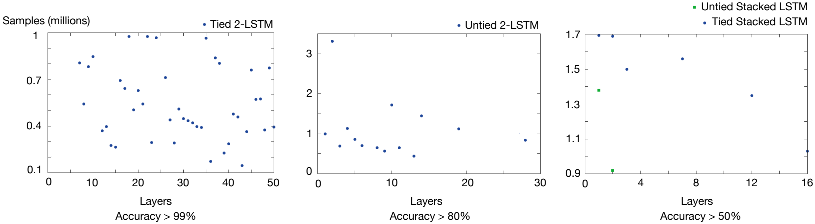
— a b c — — — —
— — — — a b c —

Memorization

Task: memorize random sequence of 20 symbols

— a b c — — — —

— — — — a b c —

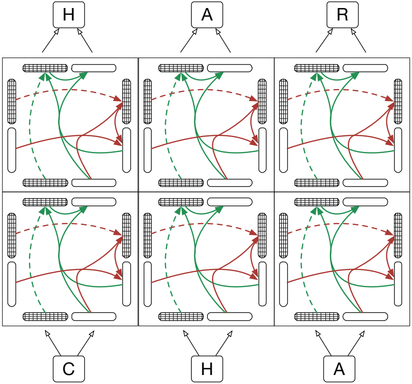


- All networks have 100 hidden units
- Vertical axis: #samples to reach threshold

Character-level LM

Task: predict next character in corpus

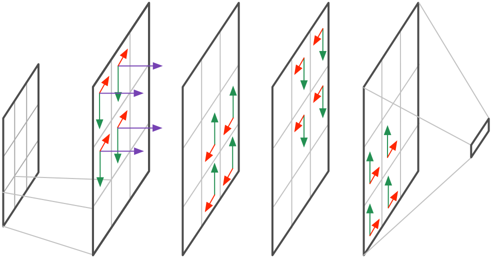
- Hutter challenge (Wikipedia data set, 100M characters)
- Sample sequences of 10000 chars, backprop every 50 chars



Character-level LM

	BPC	Params	Alphabet	Test
Stacked LSTM (Graves)	1.67	27M	205	last 4MB
MRNN (Sutskever)	1.60	4.9M	86	last 10MB
GFRNN (Chung)	1.58	20M	205	last 5MB
Tied 2-LSTM	**1.47**	16.8M	205	last 5MB

MNIST Digits

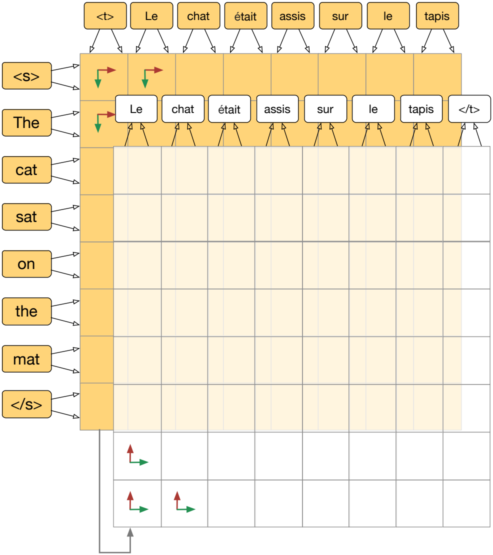


- A 3-LSTM processes non-overlapping patches of image pixels
- So, the input is a 2D grid of patches
- The 3rd dimension is the depth of the network
- Final ReLU layer + Softmax

MNIST Digits – Results

	Test Error (%)
Wan et al. (Wan et al., 2013)	0.28
Graham (Graham, 2014a)	0.31
Untied 3-LSTM	**0.32**
Ciresan et al. (Ciresan et al., 2012)	0.35
Untied 3-LSTM with ReLU	**0.36**
Mairar et al. (Mairal et al., 2014)	0.39
Lee et al. (Lee et al., 2015)	0.39
Simard et al. (Simard et al., 2003)	0.4
Graham (Graham, 2014b)	0.44
Goodfellow et al. (Goodfellow et al., 2013)	0.45
Visin et al. (Visin et al., 2015)	0.45
Lin et al. (Lin et al., 2013)	0.47

Machine Translation



Machine Translation

- We view translation as a 2-dimensional mapping

Machine Translation

- We view translation as a 2-dimensional mapping
- One dimension processes the **source** sentence, another dimension produces the **target** sentence

Machine Translation

- We view translation as a 2-dimensional mapping
- One dimension processes the **source** sentence, another dimension produces the **target** sentence
- The network **repeatedly re-encodes** the source sentence based on the part of the target sentence generated so far

Machine Translation

- We view translation as a 2-dimensional mapping
- One dimension processes the **source** sentence, another dimension produces the **target** sentence
- The network **repeatedly re-encodes** the source sentence based on the part of the target sentence generated so far
- **Weights are shared** across source and target dimensions

Machine Translation

- We view translation as a 2-dimensional mapping
- One dimension processes the **source** sentence, another dimension produces the **target** sentence
- The network **repeatedly re-encodes** the source sentence based on the part of the target sentence generated so far
- **Weights are shared** across source and target dimensions
- Regular identity connections along the 3rd dimension

Evaluation

Evaluation on IWSLT BTEC Chinese-to-English

- 44016 sentence pairs (train), 1006 (dev), 503 (test)
- Target sentences on average 12 words long
- 15 reference translations

Evaluation

Evaluation on IWSLT BTEC Chinese-to-English

- 44016 sentence pairs (train), 1006 (dev), 503 (test)
- Target sentences on average 12 words long
- 15 reference translations

	Valid-1	Test-1	Valid-15	Test-15
DGLSTM-Att.	-	34.5	-	-
CDEC	30.1	41	50.1	58.9
3-LSTM	30.3	42.4	51.8	60.2

Outline

Conclusion

- Introduction of Grid LSTM
- Cells have shown advantages is parity, addition, memorization tasks
- Applications in character prediction, MNIST, and machine translation

Bibliography

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kalchbrenner, N., Danihelka, I., and Graves, A. (2015). Grid long short-term memory. *CoRR*, abs/1507.01526.

Image Captioning & Attention

Deep Learning

Hendrik Heuer
University of Amsterdam

About me

Now

Hendrik Heuer
PhD student
QUvA Lab

prof. dr. Arnold Smeulders
dr. Christof Monz

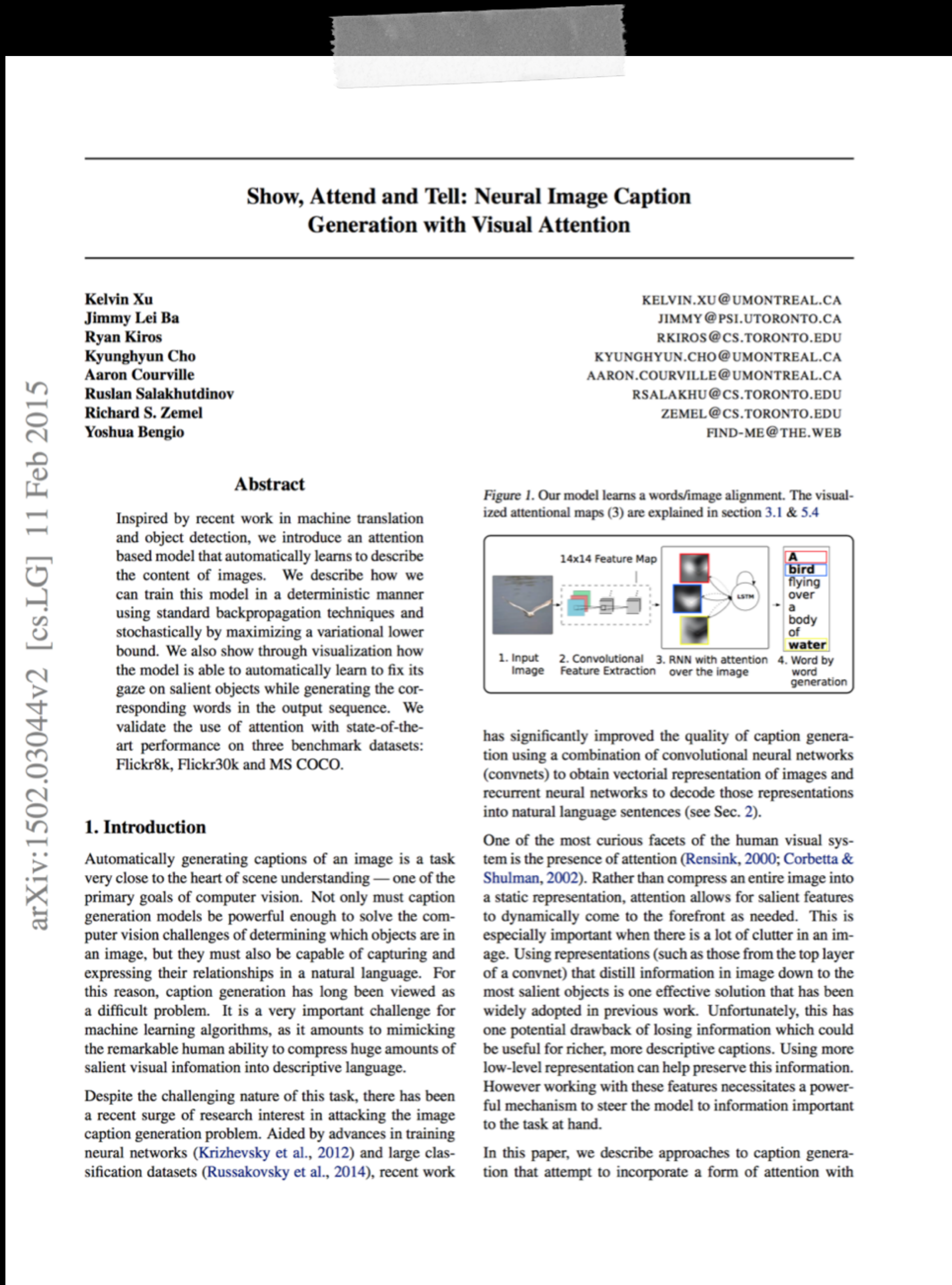
Statistical Machine Translation
& Computer Vision

M.Sc.

KTH VETENSKAP OCH KONST

Aalto University

eit | European Institute of Innovation & Technology



arXiv:1502.03044v2 [cs.LG] 11 Feb 2015

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Kelvin Xu KELVIN.XU@UMONTREAL.CA
Jimmy Lei Ba JIMMY@PSI.UTORONTO.CA
Ryan Kiros RKIROS@CS.TORONTO.EDU
Kyunghyun Cho KYUNGHYUN.CHO@UMONTREAL.CA
Aaron Courville AARON.COURVILLE@UMONTREAL.CA
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Richard S. Zemel ZEMEL@CS.TORONTO.EDU
Yoshua Bengio FIND-ME@THE.WEB

Abstract

Inspired by recent work in machine translation and object detection, we introduce an attention based model that automatically learns to describe the content of images. We describe how we can train this model in a deterministic manner using standard backpropagation techniques and stochastically by maximizing a variational lower bound. We also show through visualization how the model is able to automatically learn to fix its gaze on salient objects while generating the corresponding words in the output sequence. We validate the use of attention with state-of-the-art performance on three benchmark datasets: Flickr8k, Flickr30k and MS COCO.

Figure 1. Our model learns a words/image alignment. The visualized attentional maps (3) are explained in section 3.1 & 5.4

1. Introduction

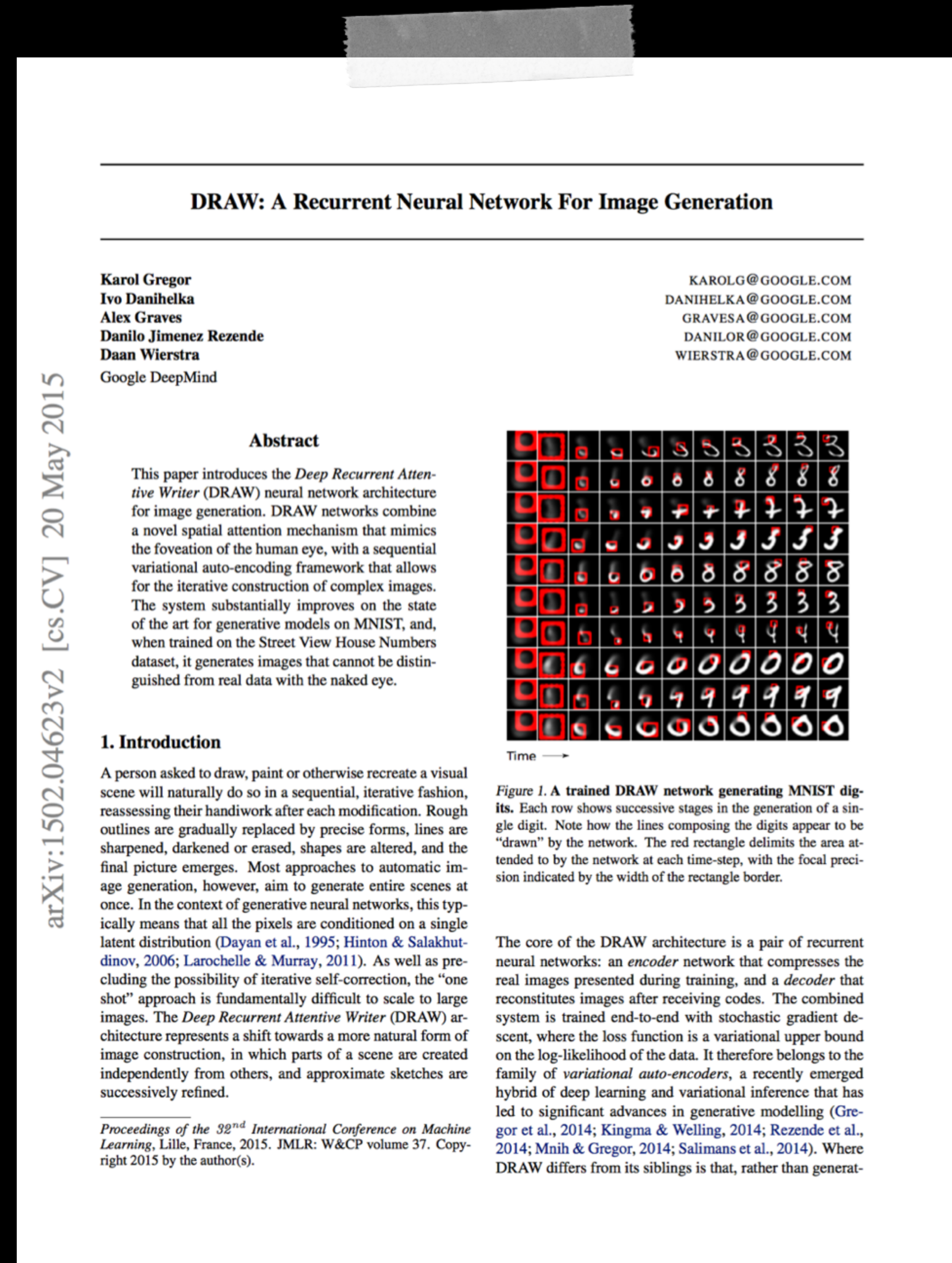
Automatically generating captions of an image is a task very close to the heart of scene understanding — one of the primary goals of computer vision. Not only must caption generation models be powerful enough to solve the computer vision challenges of determining which objects are in an image, but they must also be capable of capturing and expressing their relationships in a natural language. For this reason, caption generation has long been viewed as a difficult problem. It is a very important challenge for machine learning algorithms, as it amounts to mimicking the remarkable human ability to compress huge amounts of salient visual infomation into descriptive language.

Despite the challenging nature of this task, there has been a recent surge of research interest in attacking the image caption generation problem. Aided by advances in training neural networks (Krizhevsky et al., 2012) and large classification datasets (Russakovsky et al., 2014), recent work has significantly improved the quality of caption generation using a combination of convolutional neural networks (convnets) to obtain vectorial representation of images and recurrent neural networks to decode those representations into natural language sentences (see Sec. 2).

One of the most curious facets of the human visual system is the presence of attention (Rensink, 2000; Corbetta & Shulman, 2002). Rather than compress an entire image into a static representation, attention allows for salient features to dynamically come to the forefront as needed. This is especially important when there is a lot of clutter in an image. Using representations (such as those from the top layer of a convnet) that distill information in image down to the most salient objects is one effective solution that has been widely adopted in previous work. Unfortunately, this has one potential drawback of losing information which could be useful for richer, more descriptive captions. Using more low-level representation can help preserve this information. However working with these features necessitates a powerful mechanism to steer the model to information important to the task at hand.

In this paper, we describe approaches to caption generation that attempt to incorporate a form of attention with

Describe images



arXiv:1502.04623v2 [cs.CV] 20 May 2015

DRAW: A Recurrent Neural Network For Image Generation

Karol Gregor KAROLG@GOOGLE.COM
Ivo Danihelka DANIHELKA@GOOGLE.COM
Alex Graves GRAVESA@GOOGLE.COM
Danilo Jimenez Rezende DANILOR@GOOGLE.COM
Daan Wierstra WIERSTRA@GOOGLE.COM
Google DeepMind

Abstract

This paper introduces the *Deep Recurrent Attentive Writer* (DRAW) neural network architecture for image generation. DRAW networks combine a novel spatial attention mechanism that mimics the foveation of the human eye, with a sequential variational auto-encoding framework that allows for the iterative construction of complex images. The system substantially improves on the state of the art for generative models on MNIST, and, when trained on the Street View House Numbers dataset, it generates images that cannot be distinguished from real data with the naked eye.

Figure 1. **A trained DRAW network generating MNIST digits.** Each row shows successive stages in the generation of a single digit. Note how the lines composing the digits appear to be "drawn" by the network. The red rectangle delimits the area attended to by the network at each time-step, with the focal precision indicated by the width of the rectangle border.

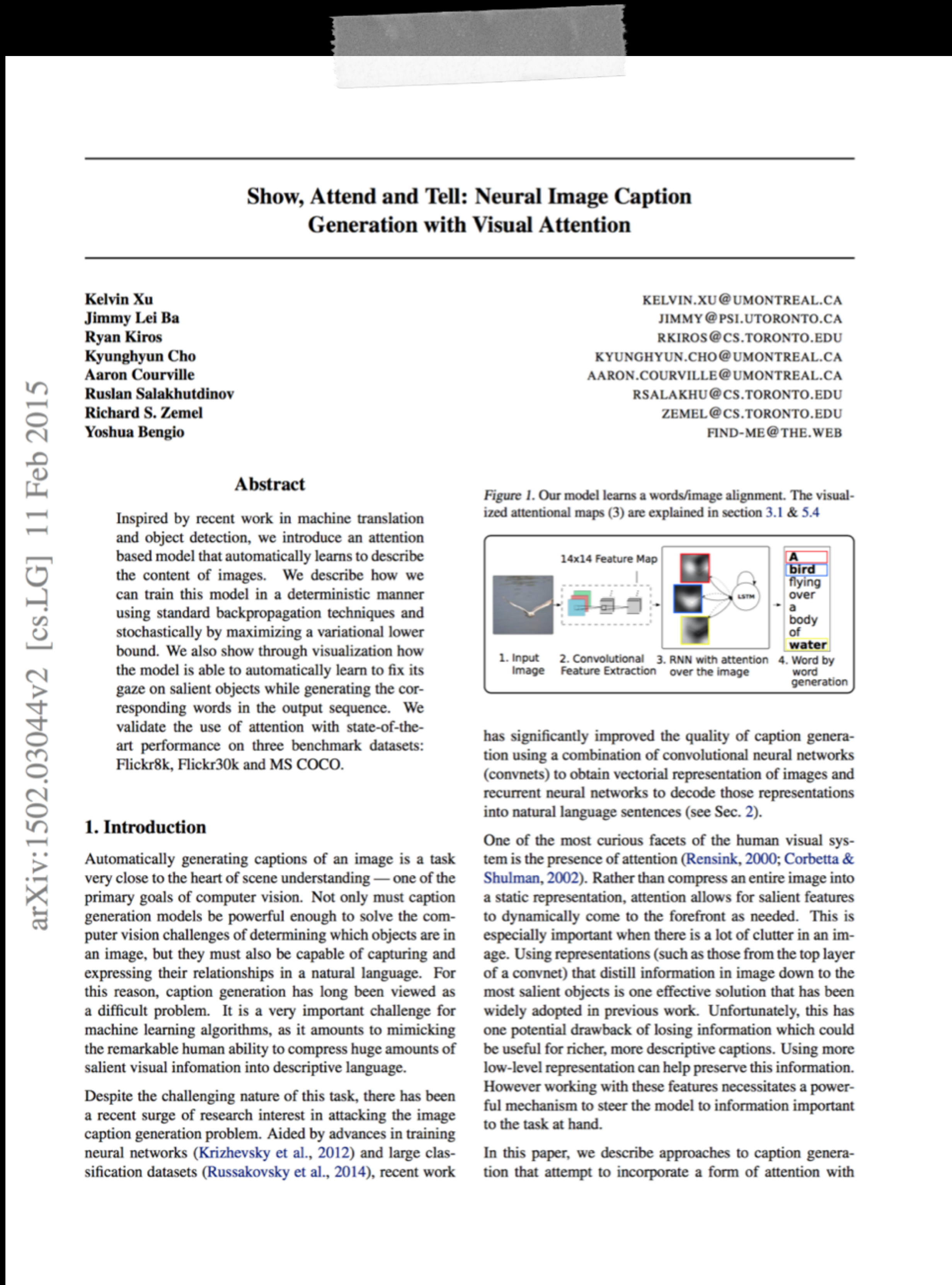
1. Introduction

A person asked to draw, paint or otherwise recreate a visual scene will naturally do so in a sequential, iterative fashion, reassessing their handiwork after each modification. Rough outlines are gradually replaced by precise forms, lines are sharpened, darkened or erased, shapes are altered, and the final picture emerges. Most approaches to automatic image generation, however, aim to generate entire scenes at once. In the context of generative neural networks, this typically means that all the pixels are conditioned on a single latent distribution (Dayan et al., 1995; Hinton & Salakhutdinov, 2006; Larochelle & Murray, 2011). As well as precluding the possibility of iterative self-correction, the "one shot" approach is fundamentally difficult to scale to large images. The *Deep Recurrent Attentive Writer* (DRAW) architecture represents a shift towards a more natural form of image construction, in which parts of a scene are created independently from others, and approximate sketches are successively refined.

The core of the DRAW architecture is a pair of recurrent neural networks: an *encoder* network that compresses the real images presented during training, and a *decoder* that reconstitutes images after receiving codes. The combined system is trained end-to-end with stochastic gradient descent, where the loss function is a variational upper bound on the log-likelihood of the data. It therefore belongs to the family of *variational auto-encoders*, a recently emerged hybrid of deep learning and variational inference that has led to significant advances in generative modelling (Gregor et al., 2014; Kingma & Welling, 2014; Rezende et al., 2014; Mnih & Gregor, 2014; Salimans et al., 2014). Where DRAW differs from its siblings is that, rather than generat-

Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

Generate images



Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Kelvin Xu KELVIN.XU@UMONTREAL.CA
Jimmy Lei Ba JIMMY@PSI.UTORONTO.CA
Ryan Kiros RKIROS@CS.TORONTO.EDU
Kyunghyun Cho KYUNGHYUN.CHO@UMONTREAL.CA
Aaron Courville AARON.COURVILLE@UMONTREAL.CA
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Richard S. Zemel ZEMEL@CS.TORONTO.EDU
Yoshua Bengio FIND-ME@THE.WEB

arXiv:1502.03044v2 [cs.LG] 11 Feb 2015

Abstract

Inspired by recent work in machine translation and object detection, we introduce an attention based model that automatically learns to describe the content of images. We describe how we can train this model in a deterministic manner using standard backpropagation techniques and stochastically by maximizing a variational lower bound. We also show through visualization how the model is able to automatically learn to fix its gaze on salient objects while generating the corresponding words in the output sequence. We validate the use of attention with state-of-the-art performance on three benchmark datasets: Flickr8k, Flickr30k and MS COCO.

Figure 1. Our model learns a words/image alignment. The visualized attentional maps (3) are explained in section 3.1 & 5.4

14x14 Feature Map
LSTM
A bird flying over a body of water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

1. Introduction

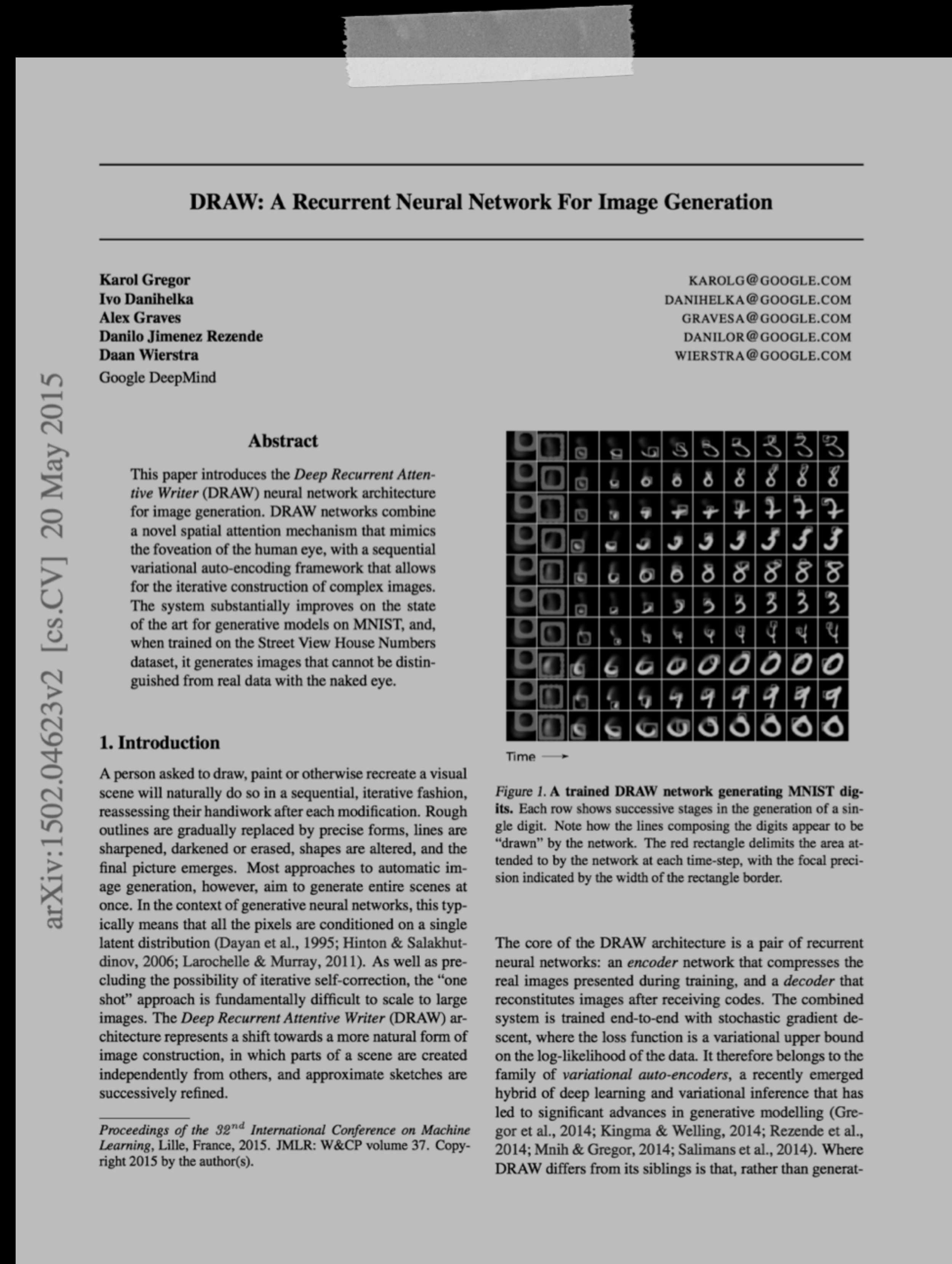
Automatically generating captions of an image is a task very close to the heart of scene understanding — one of the primary goals of computer vision. Not only must caption generation models be powerful enough to solve the computer vision challenges of determining which objects are in an image, but they must also be capable of capturing and expressing their relationships in a natural language. For this reason, caption generation has long been viewed as a difficult problem. It is a very important challenge for machine learning algorithms, as it amounts to mimicking the remarkable human ability to compress huge amounts of salient visual infomation into descriptive language.

Despite the challenging nature of this task, there has been a recent surge of research interest in attacking the image caption generation problem. Aided by advances in training neural networks (Krizhevsky et al., 2012) and large classification datasets (Russakovsky et al., 2014), recent work has significantly improved the quality of caption generation using a combination of convolutional neural networks (convnets) to obtain vectorial representation of images and recurrent neural networks to decode those representations into natural language sentences (see Sec. 2).

One of the most curious facets of the human visual system is the presence of attention (Rensink, 2000; Corbetta & Shulman, 2002). Rather than compress an entire image into a static representation, attention allows for salient features to dynamically come to the forefront as needed. This is especially important when there is a lot of clutter in an image. Using representations (such as those from the top layer of a convnet) that distill information in image down to the most salient objects is one effective solution that has been widely adopted in previous work. Unfortunately, this has one potential drawback of losing information which could be useful for richer, more descriptive captions. Using more low-level representation can help preserve this information. However working with these features necessitates a powerful mechanism to steer the model to information important to the task at hand.

In this paper, we describe approaches to caption generation that attempt to incorporate a form of attention with

Describe images



DRAW: A Recurrent Neural Network For Image Generation

Karol Gregor KAROLG@GOOGLE.COM
Ivo Danihelka DANIHELKA@GOOGLE.COM
Alex Graves GRAVESA@GOOGLE.COM
Danilo Jimenez Rezende DANILOR@GOOGLE.COM
Daan Wierstra WIERSTRA@GOOGLE.COM
Google DeepMind

arXiv:1502.04623v2 [cs.CV] 20 May 2015

Abstract

This paper introduces the *Deep Recurrent Attentive Writer* (DRAW) neural network architecture for image generation. DRAW networks combine a novel spatial attention mechanism that mimics the foveation of the human eye, with a sequential variational auto-encoding framework that allows for the iterative construction of complex images. The system substantially improves on the state of the art for generative models on MNIST, and, when trained on the Street View House Numbers dataset, it generates images that cannot be distinguished from real data with the naked eye.

Time →

Figure 1. **A trained DRAW network generating MNIST digits.** Each row shows successive stages in the generation of a single digit. Note how the lines composing the digits appear to be "drawn" by the network. The red rectangle delimits the area attended to by the network at each time-step, with the focal precision indicated by the width of the rectangle border.

1. Introduction

A person asked to draw, paint or otherwise recreate a visual scene will naturally do so in a sequential, iterative fashion, reassessing their handiwork after each modification. Rough outlines are gradually replaced by precise forms, lines are sharpened, darkened or erased, shapes are altered, and the final picture emerges. Most approaches to automatic image generation, however, aim to generate entire scenes at once. In the context of generative neural networks, this typically means that all the pixels are conditioned on a single latent distribution (Dayan et al., 1995; Hinton & Salakhutdinov, 2006; Larochelle & Murray, 2011). As well as precluding the possibility of iterative self-correction, the "one shot" approach is fundamentally difficult to scale to large images. The *Deep Recurrent Attentive Writer* (DRAW) architecture represents a shift towards a more natural form of image construction, in which parts of a scene are created independently from others, and approximate sketches are successively refined.

The core of the DRAW architecture is a pair of recurrent neural networks: an *encoder* network that compresses the real images presented during training, and a *decoder* that reconstitutes images after receiving codes. The combined system is trained end-to-end with stochastic gradient descent, where the loss function is a variational upper bound on the log-likelihood of the data. It therefore belongs to the family of *variational auto-encoders*, a recently emerged hybrid of deep learning and variational inference that has led to significant advances in generative modelling (Gregor et al., 2014; Kingma & Welling, 2014; Rezende et al., 2014; Mnih & Gregor, 2014; Salimans et al., 2014). Where DRAW differs from its siblings is that, rather than generat-

Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

Generate images

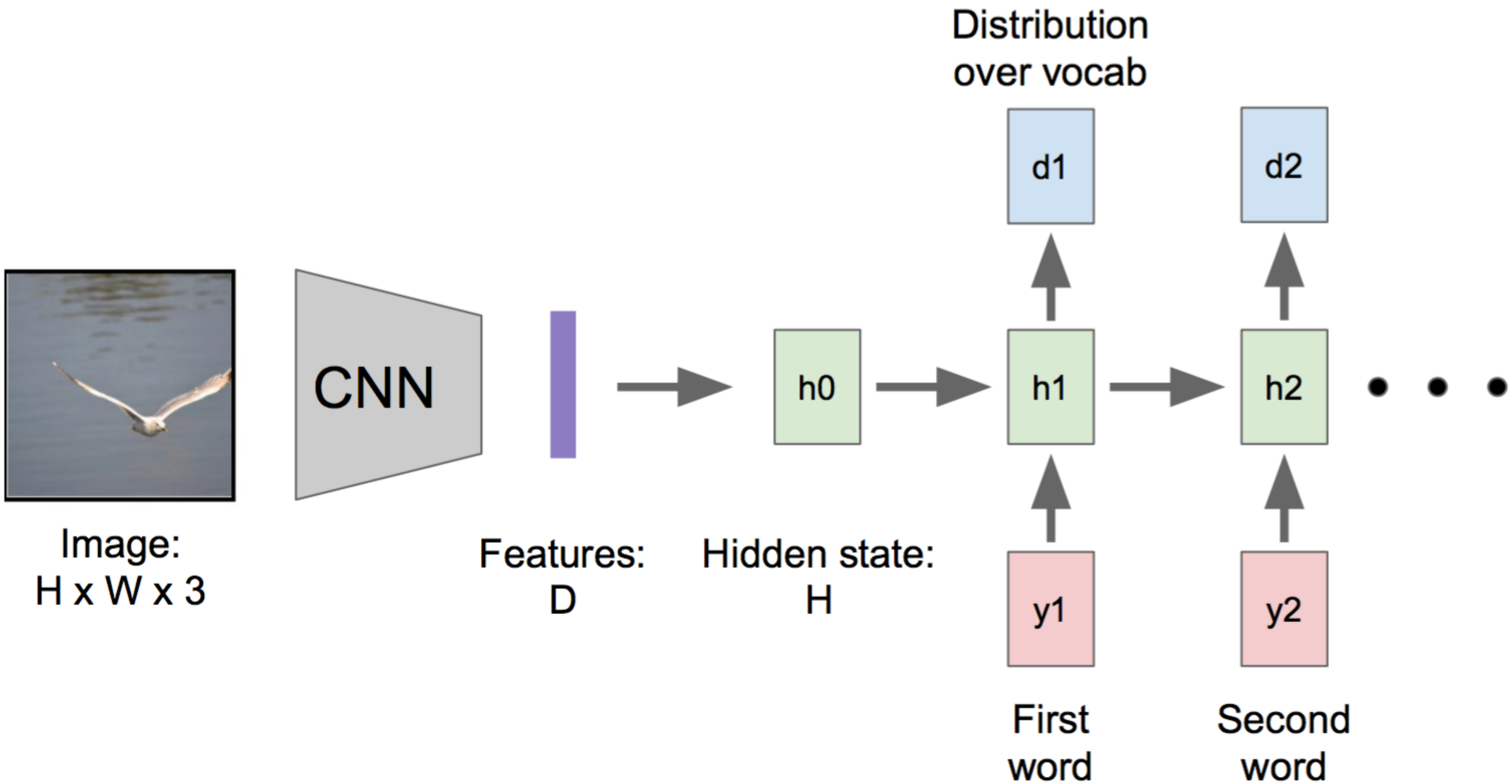
Grammatical description of images

https://vimeo.com/146492001

Grammatical description of images

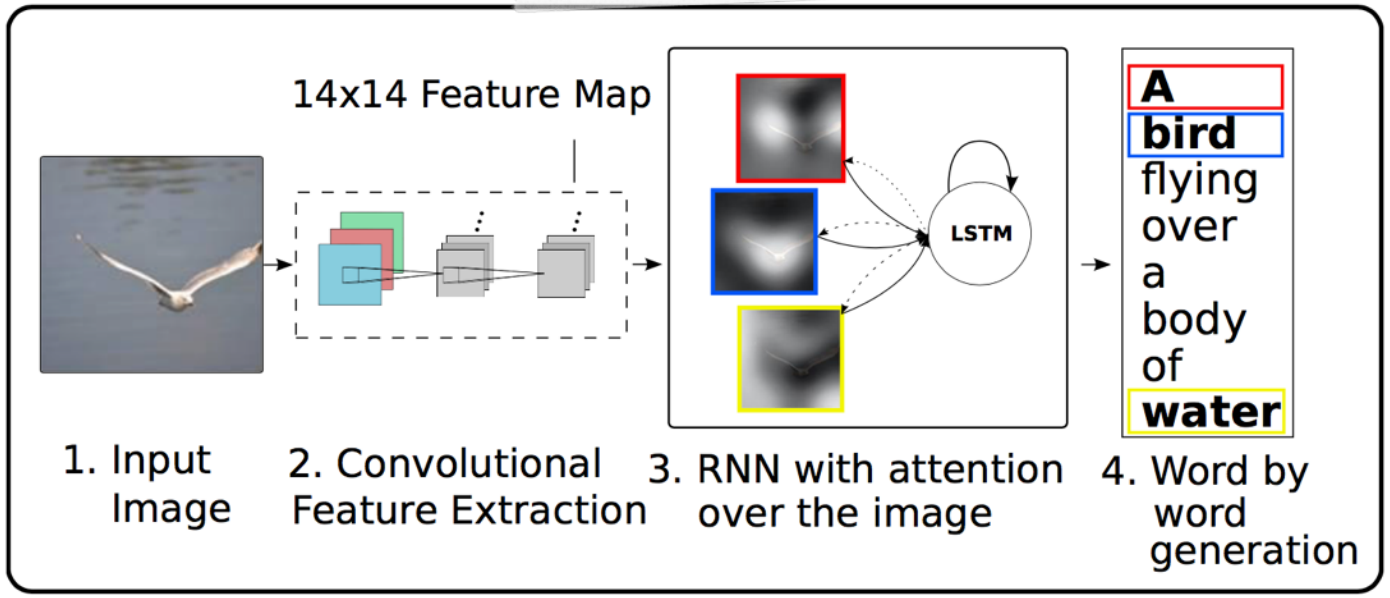
- a smiling old lady holds a pizza on a plate.
- a woman holding a plate with a pizza on it
- a woman carrying homemade pizza to the table.
- a woman holding a pizza on a red plate.
- a woman walking with a pan in her hands with a whole pizza on it.

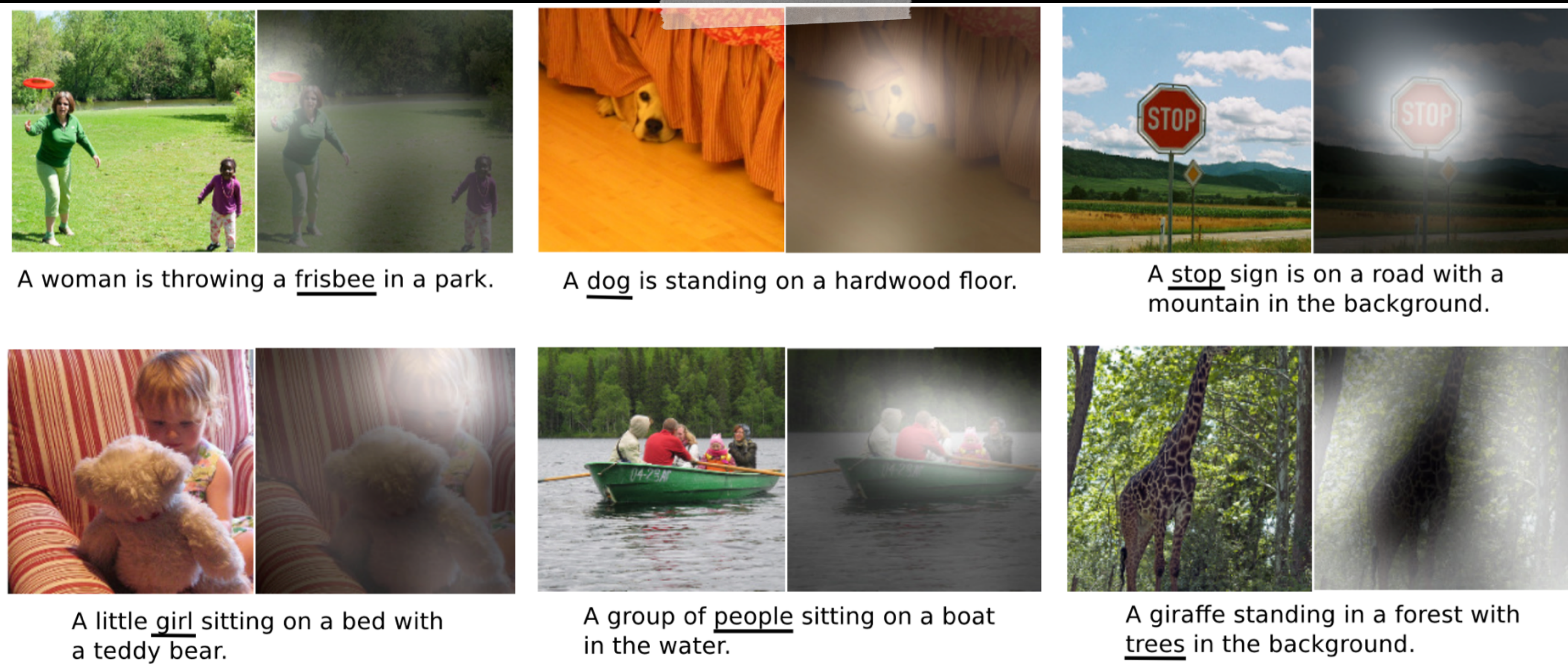
RNN for Captioning

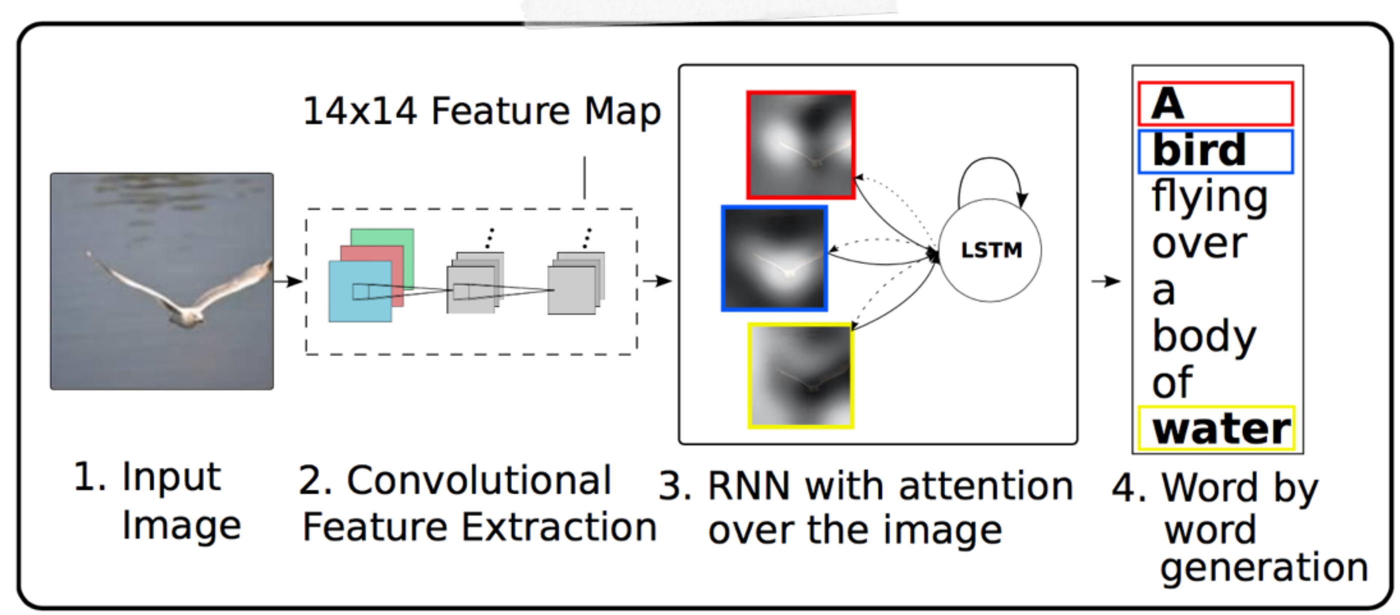


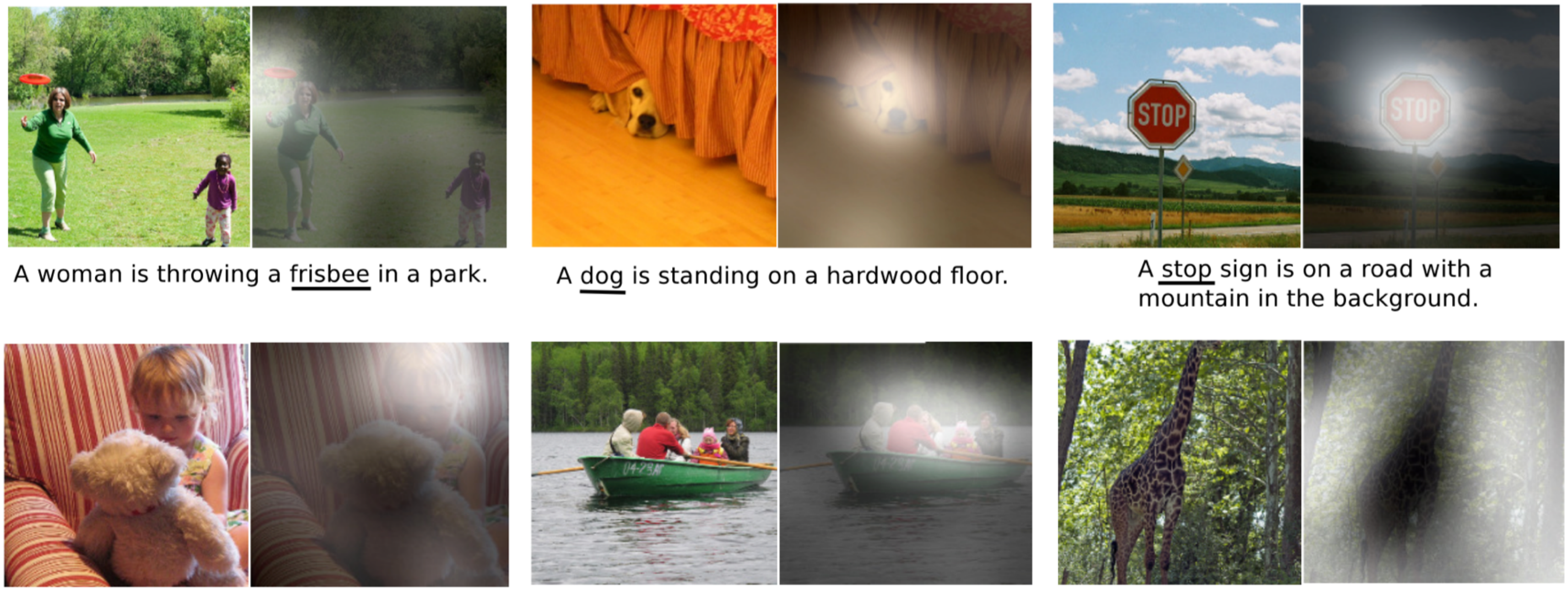
Source: http://cs231n.stanford.edu/slides/winter1516_lecture13.pdf

(b) A woman is throwing a frisbee in a park.




STOP
A woman is throwing a frisbee in a park.
A dog is standing on a hardwood floor.
A stop sign is on a road with a mountain in the background.
A little girl sitting on a bed with a teddy bear.
A group of people sitting on a boat in the water.
A giraffe standing in a forest with trees in the background.


14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation


A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

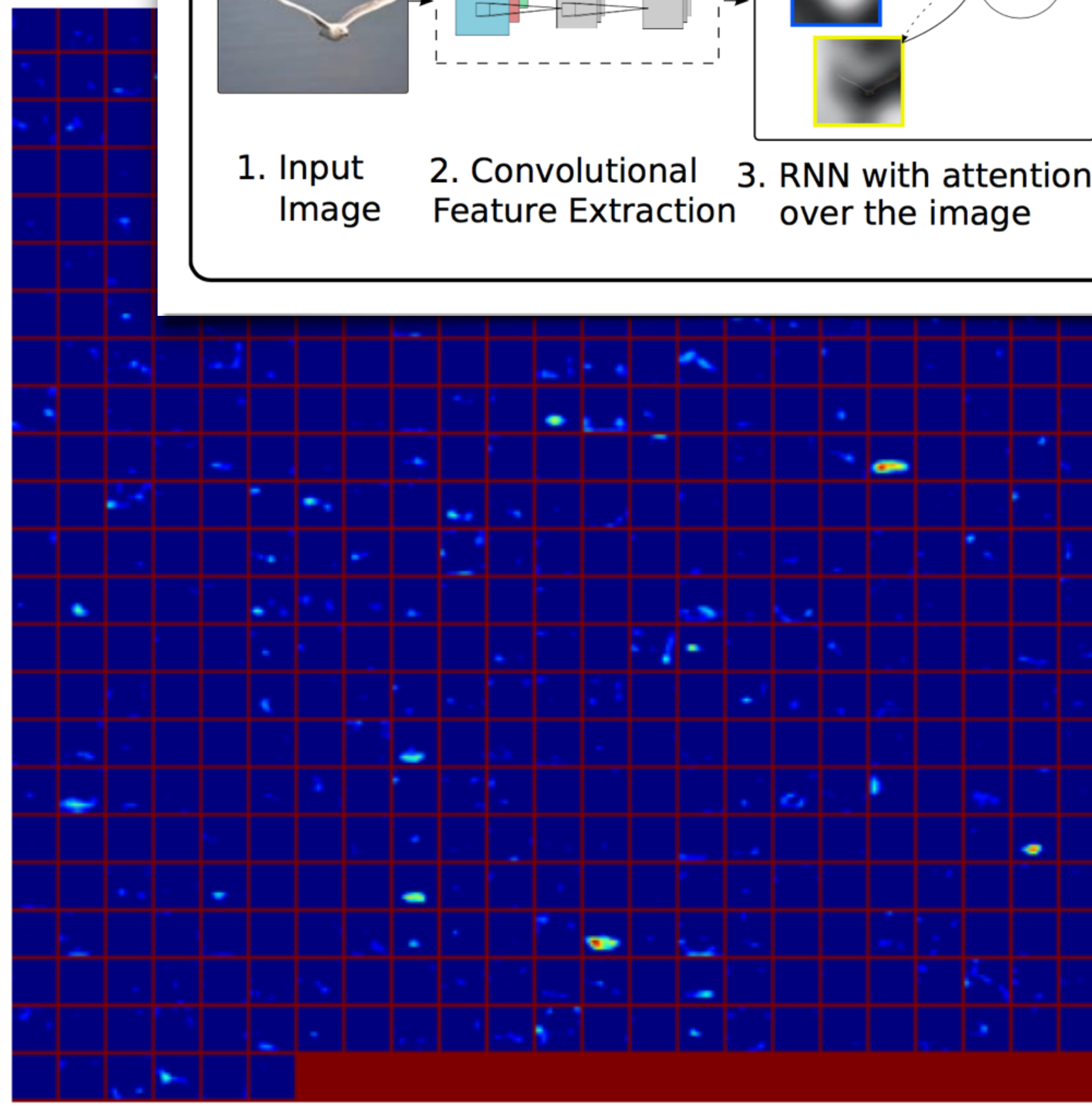
A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

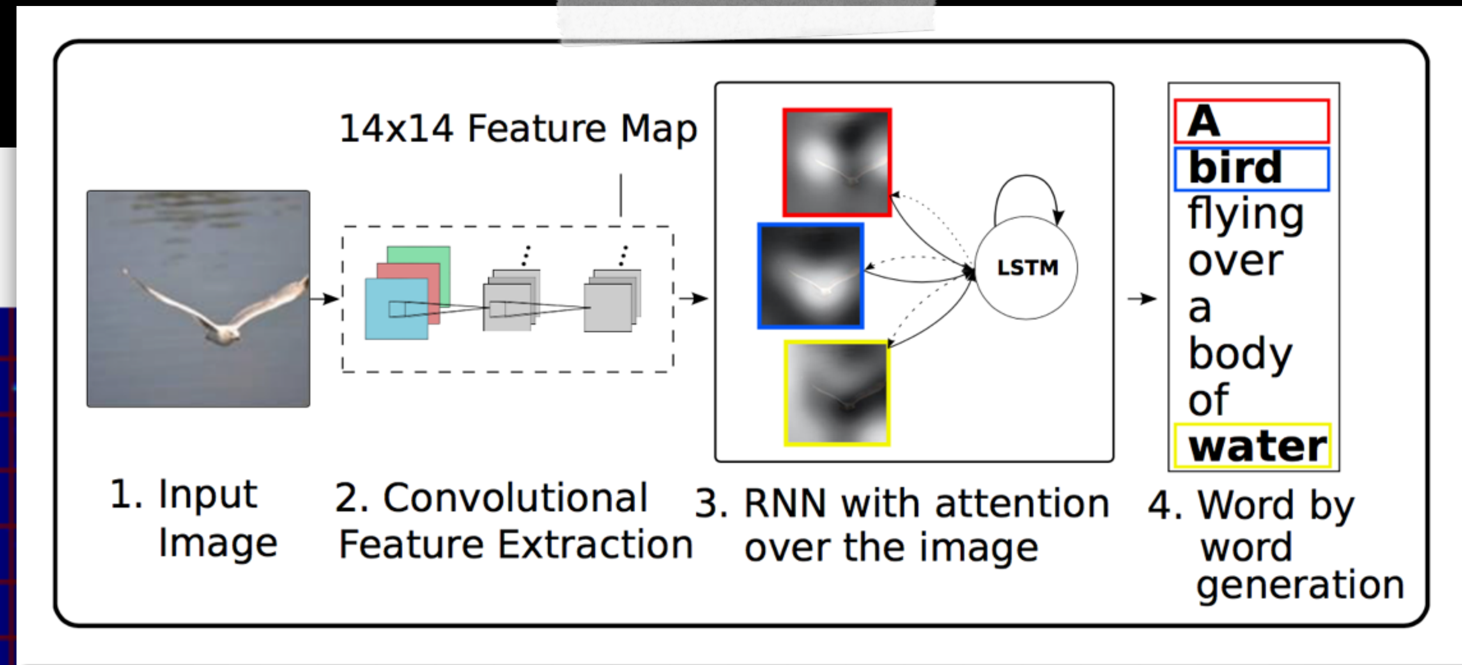
A giraffe standing in a forest with trees in the background.

Intuition: Since we usually see dogs at a certain position, we expect dogs at certain positions.

The model learns correlation structures in the input and starts putting attention weight where dogs can be expected (and actually exist in the training data).



512 filter, each 14x14 pixel



"soft" deterministic attention

summarize all locations, so that context vector z is

$$\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t] = \sum_{i=1}^{L} \alpha_{t,i}\mathbf{a}_i$$

you can take the derivative dz/dp

trainable by back-propagation

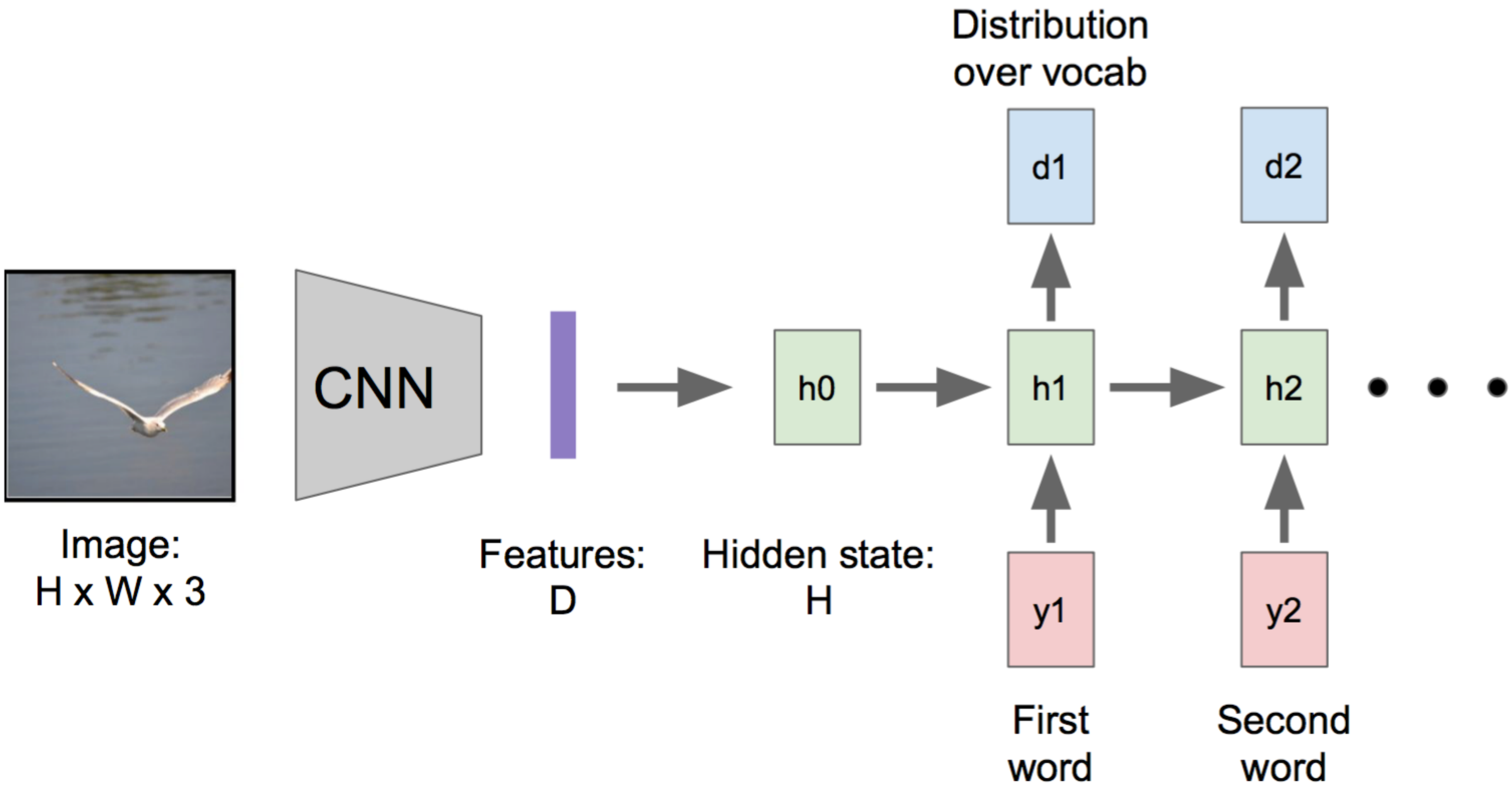
"hard" stochastic attention

sample one locatio

since you do argmax, gradient is zero almost everywhere, so you can't use gradient descent

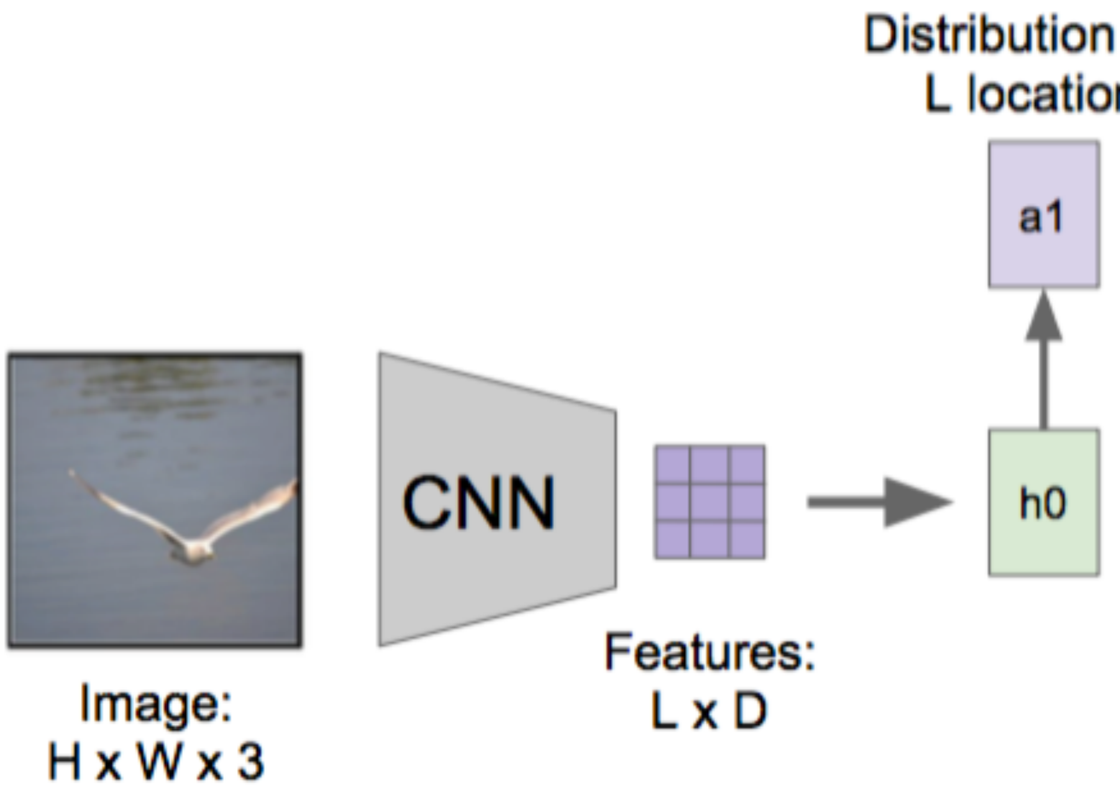
reinforcement learning:
REINFORCE (Williams, 1992)

RNN for Captioning



Source: http://cs231n.stanford.edu/slides/winter1516_lecture13.pdf

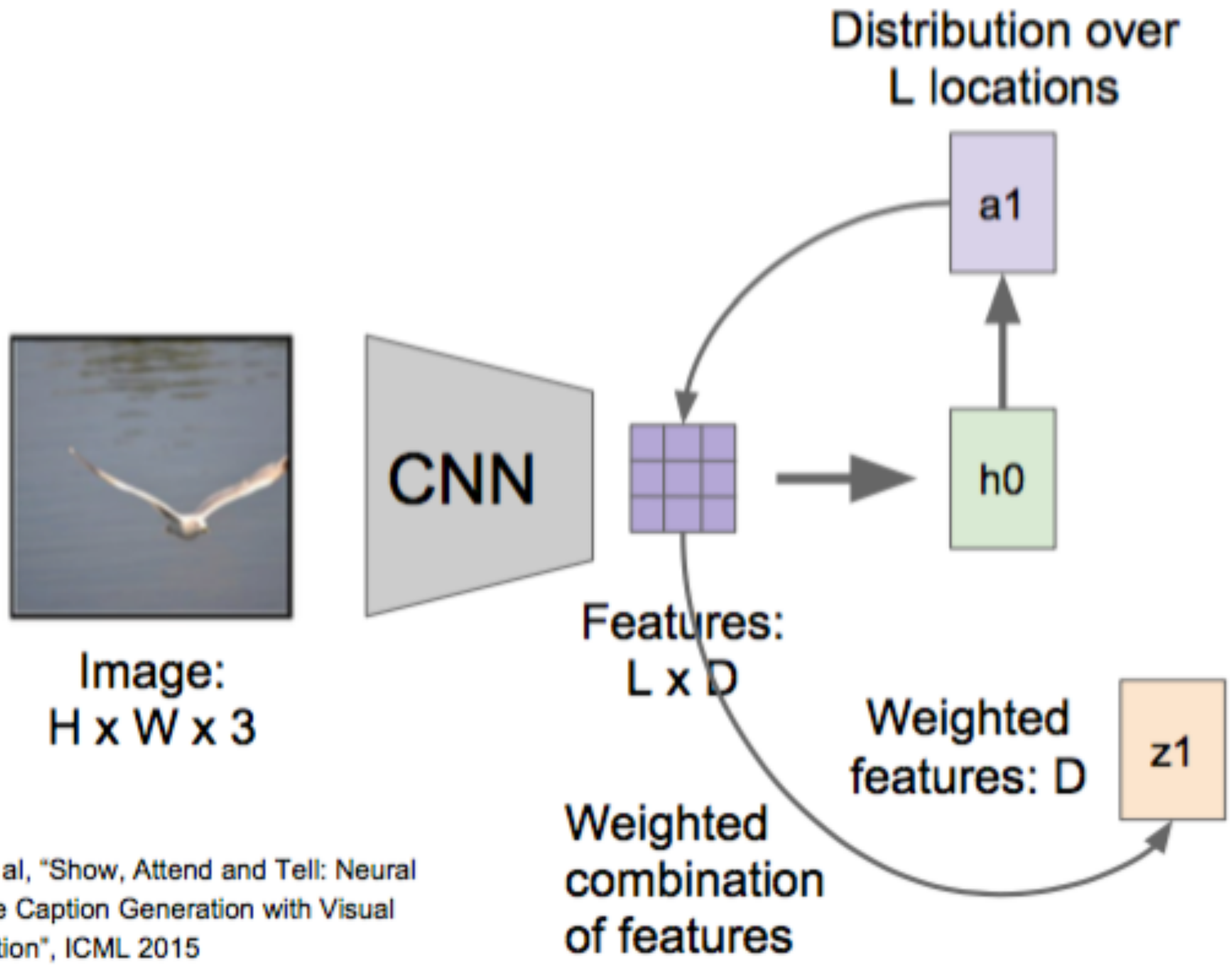
Soft Attention for Captioning



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Source: http://cs231n.stanford.edu/slides/winter1516_lecture13.pdf

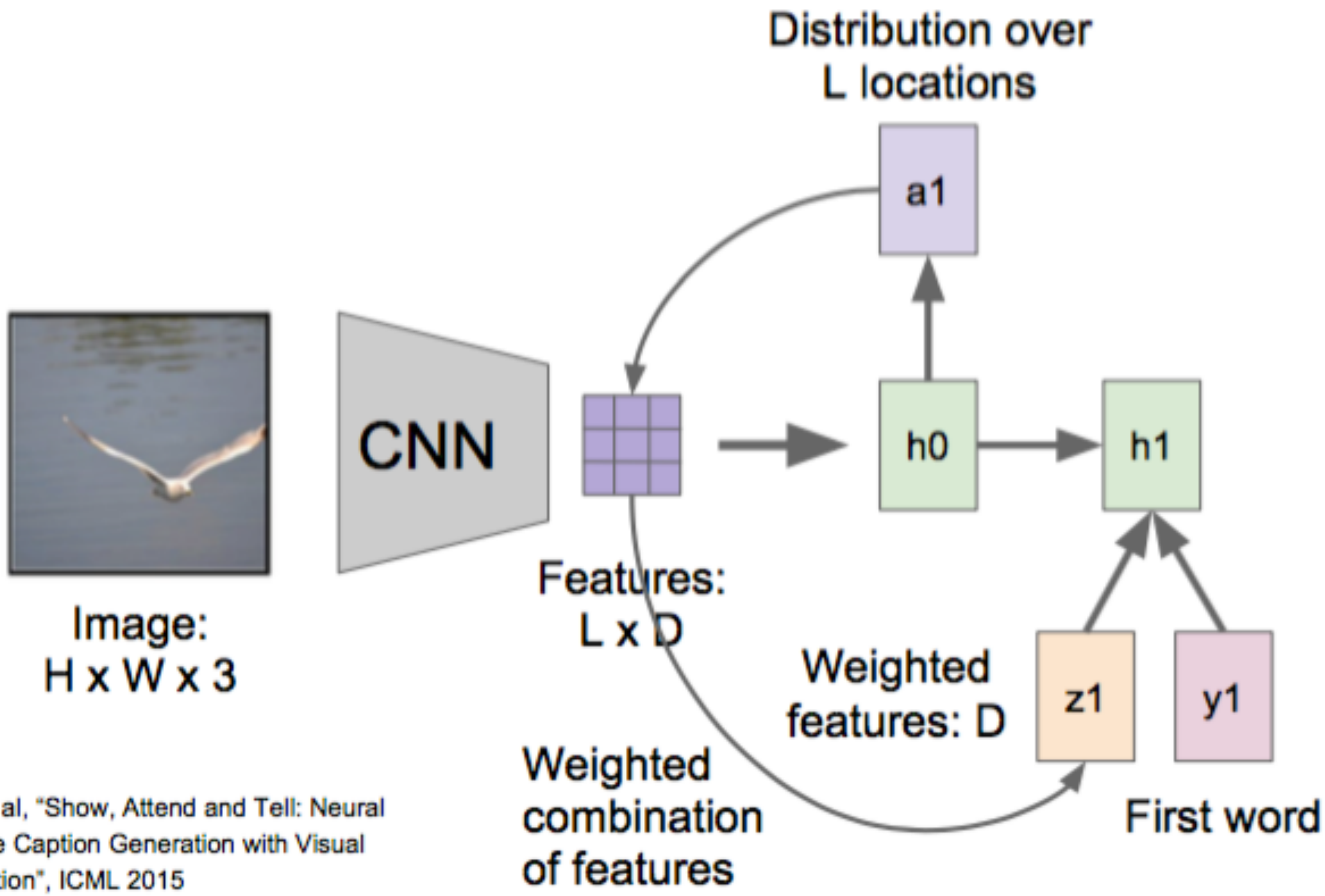
Soft Attention for Captioning



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Source: http://cs231n.stanford.edu/slides/winter1516_lecture13.pdf

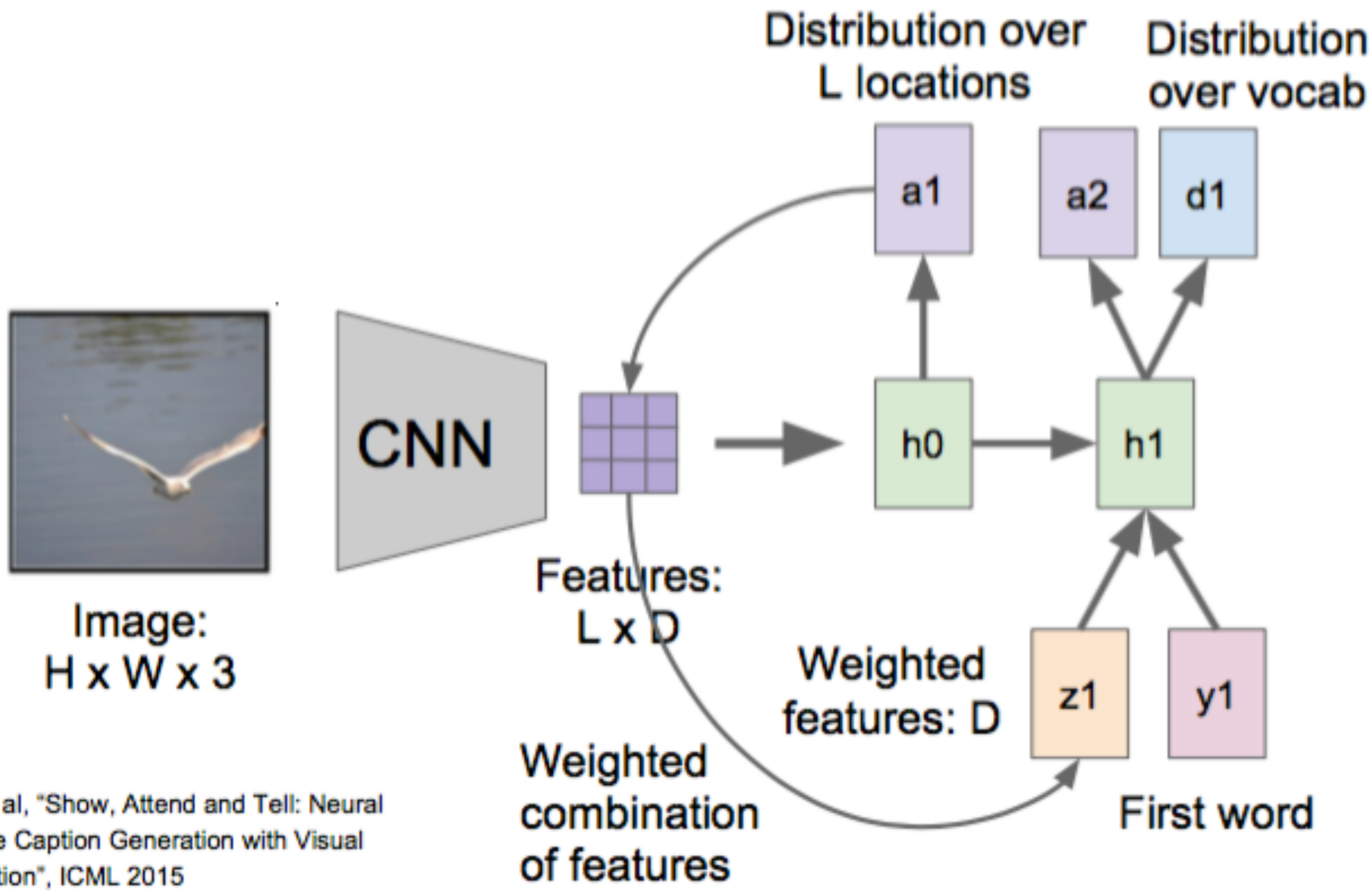
Soft Attention for Captioning



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Source: http://cs231n.stanford.edu/slides/winter1516_lecture13.pdf

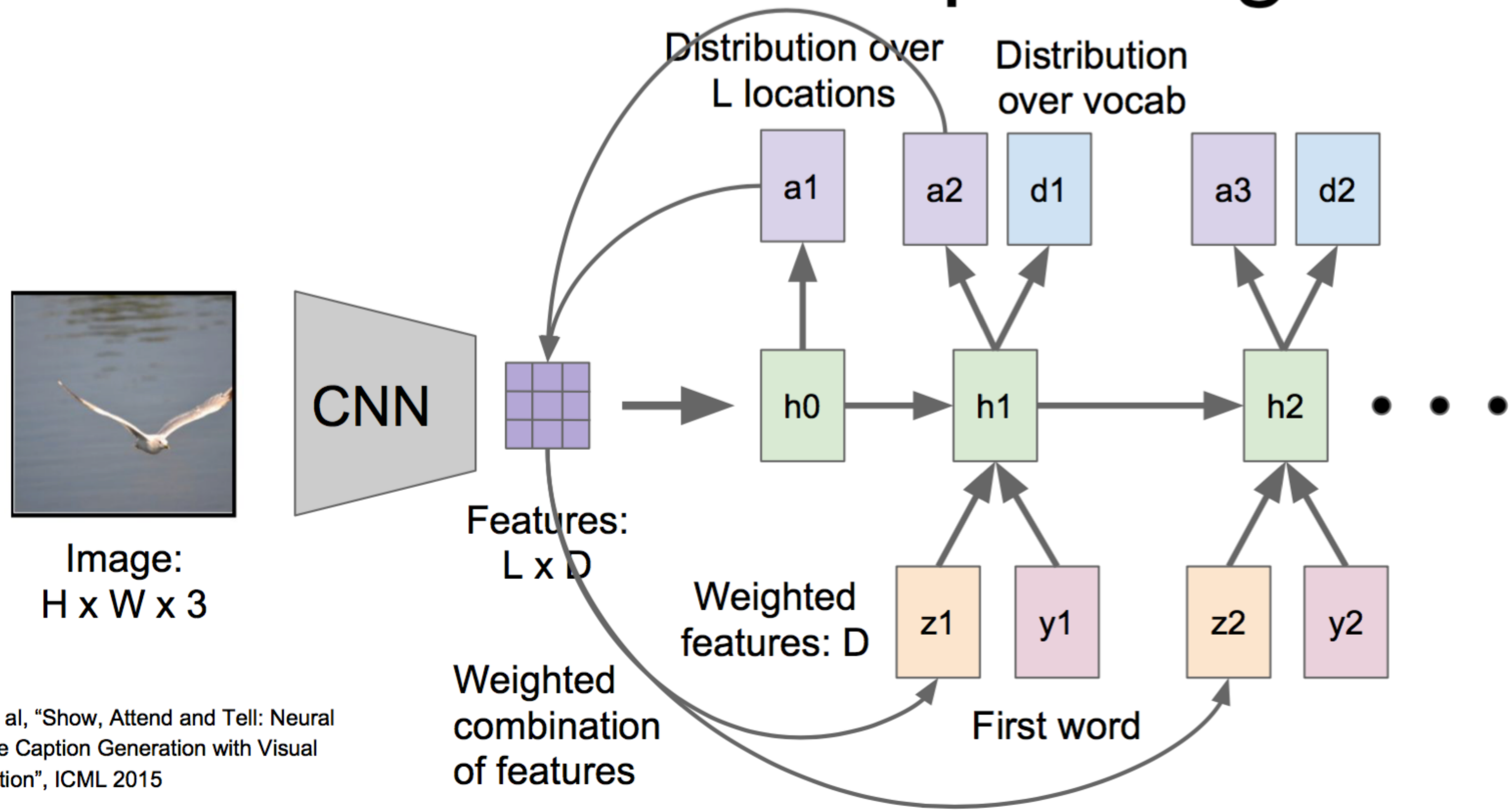
Soft Attention for Captioning



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Source: http://cs231n.stanford.edu/slides/winter1516_lecture13.pdf

Soft Attention for Captioning



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

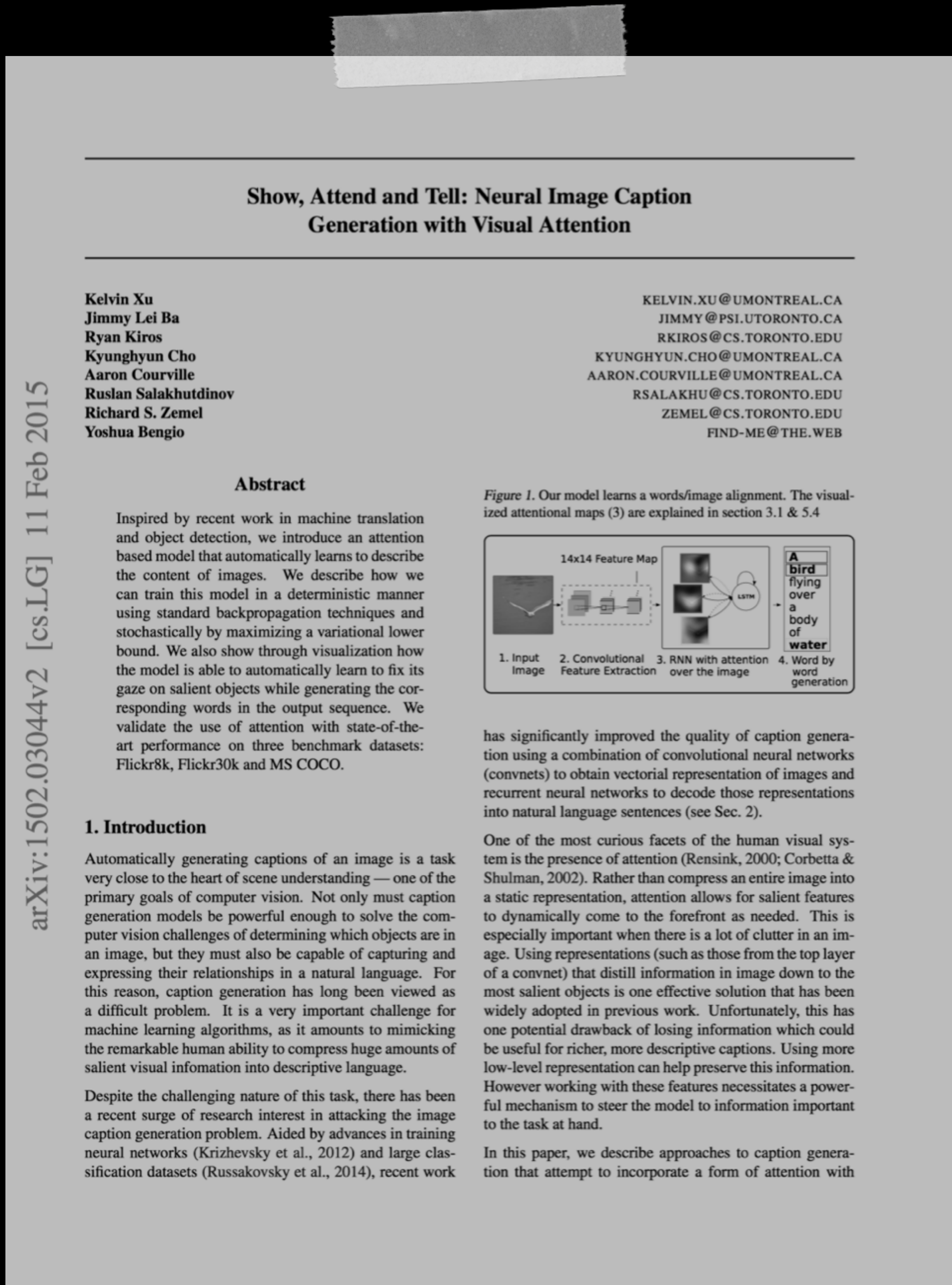
Source: http://cs231n.stanford.edu/slides/winter1516_lecture13.pdf

Evaluation

Dataset	Model	BLEU B-1	BLEU B-2	BLEU B-3	BLEU B-4	METEOR
Flickr8k	Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)$^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC$^{\dagger\circ\Sigma}$	66.3	42.3	27.7	18.3	—
	Log Bilinear	60.0	38	25.4	17.1	16.88
	Soft-Attention	66.7	43.4	28.8	19.1	**18.49**
	Hard-Attention	**66.9**	**43.9**	**29.6**	**19.9**	18.46
COCO	CMU/MS Research (Chen & Zitnick, 2014)^{a}	—	—	—	—	20.41
	MS Research (Fang et al., 2014)$^{\dagger a}$	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)$^{\circ}$	64.2	45.1	30.4	20.3	—
	Google NIC$^{\dagger\circ\Sigma}$	66.6	46.1	32.9	24.6	—
	Log Bilinear$^{\circ}$	70.8	48.9	34.4	24.3	20.03
	Soft-Attention	70.7	49.2	34.4	24.3	**23.90**
	Hard-Attention	**71.8**	**50.4**	**35.7**	**25.0**	23.04

BLEU is n-gram precision

METEOR is a combination of unigram-precision, unigram-recall, and a measure of fragmentation (how well-ordered matched words are)



arXiv:1502.03044v2 [cs.LG] 11 Feb 2015

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Kelvin Xu KELVIN.XU@UMONTREAL.CA
Jimmy Lei Ba JIMMY@PSI.UTORONTO.CA
Ryan Kiros RKIROS@CS.TORONTO.EDU
Kyunghyun Cho KYUNGHYUN.CHO@UMONTREAL.CA
Aaron Courville AARON.COURVILLE@UMONTREAL.CA
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Richard S. Zemel ZEMEL@CS.TORONTO.EDU
Yoshua Bengio FIND-ME@THE.WEB

Abstract

Inspired by recent work in machine translation and object detection, we introduce an attention based model that automatically learns to describe the content of images. We describe how we can train this model in a deterministic manner using standard backpropagation techniques and stochastically by maximizing a variational lower bound. We also show through visualization how the model is able to automatically learn to fix its gaze on salient objects while generating the corresponding words in the output sequence. We validate the use of attention with state-of-the-art performance on three benchmark datasets: Flickr8k, Flickr30k and MS COCO.

Figure 1. Our model learns a words/image alignment. The visualized attentional maps (3) are explained in section 3.1 & 5.4

14x14 Feature Map
A bird flying over a body of water
1. Input Image 2. Convolutional Feature Extraction 3. RNN with attention over the image 4. Word by word generation

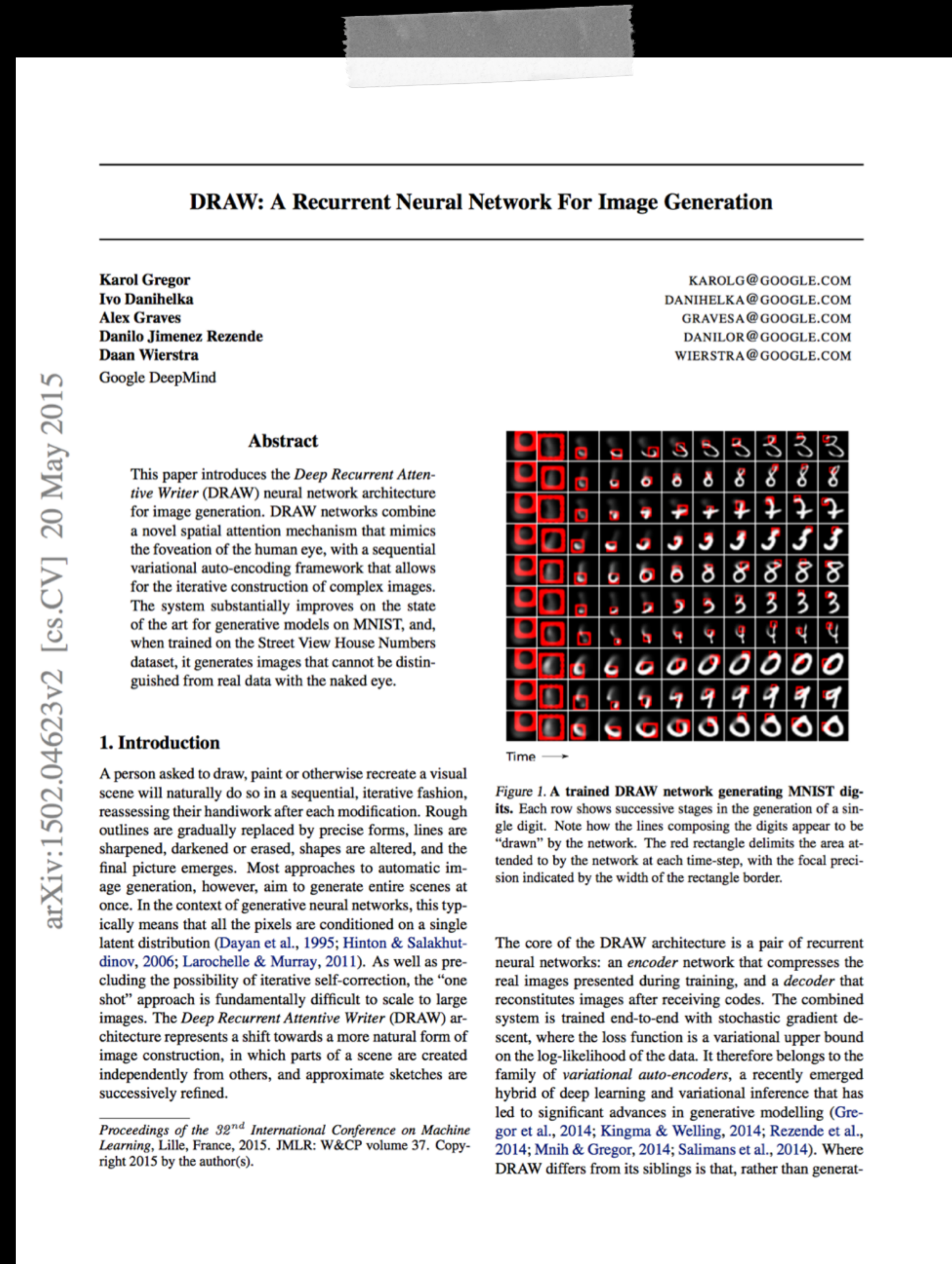
1. Introduction

Automatically generating captions of an image is a task very close to the heart of scene understanding — one of the primary goals of computer vision. Not only must caption generation models be powerful enough to solve the computer vision challenges of determining which objects are in an image, but they must also be capable of capturing and expressing their relationships in a natural language. For this reason, caption generation has long been viewed as a difficult problem. It is a very important challenge for machine learning algorithms, as it amounts to mimicking the remarkable human ability to compress huge amounts of salient visual infomation into descriptive language.

Despite the challenging nature of this task, there has been a recent surge of research interest in attacking the image caption generation problem. Aided by advances in training neural networks (Krizhevsky et al., 2012) and large classification datasets (Russakovsky et al., 2014), recent work has significantly improved the quality of caption generation using a combination of convolutional neural networks (convnets) to obtain vectorial representation of images and recurrent neural networks to decode those representations into natural language sentences (see Sec. 2).

One of the most curious facets of the human visual system is the presence of attention (Rensink, 2000; Corbetta & Shulman, 2002). Rather than compress an entire image into a static representation, attention allows for salient features to dynamically come to the forefront as needed. This is especially important when there is a lot of clutter in an image. Using representations (such as those from the top layer of a convnet) that distill information in image down to the most salient objects is one effective solution that has been widely adopted in previous work. Unfortunately, this has one potential drawback of losing information which could be useful for richer, more descriptive captions. Using more low-level representation can help preserve this information. However working with these features necessitates a powerful mechanism to steer the model to information important to the task at hand.

In this paper, we describe approaches to caption generation that attempt to incorporate a form of attention with



arXiv:1502.04623v2 [cs.CV] 20 May 2015

DRAW: A Recurrent Neural Network For Image Generation

Karol Gregor KAROLG@GOOGLE.COM
Ivo Danihelka DANIHELKA@GOOGLE.COM
Alex Graves GRAVESA@GOOGLE.COM
Danilo Jimenez Rezende DANILOR@GOOGLE.COM
Daan Wierstra WIERSTRA@GOOGLE.COM
Google DeepMind

Abstract

This paper introduces the *Deep Recurrent Attentive Writer* (DRAW) neural network architecture for image generation. DRAW networks combine a novel spatial attention mechanism that mimics the foveation of the human eye, with a sequential variational auto-encoding framework that allows for the iterative construction of complex images. The system substantially improves on the state of the art for generative models on MNIST, and, when trained on the Street View House Numbers dataset, it generates images that cannot be distinguished from real data with the naked eye.

Time →

Figure 1. **A trained DRAW network generating MNIST digits.** Each row shows successive stages in the generation of a single digit. Note how the lines composing the digits appear to be "drawn" by the network. The red rectangle delimits the area attended to by the network at each time-step, with the focal precision indicated by the width of the rectangle border.

1. Introduction

A person asked to draw, paint or otherwise recreate a visual scene will naturally do so in a sequential, iterative fashion, reassessing their handiwork after each modification. Rough outlines are gradually replaced by precise forms, lines are sharpened, darkened or erased, shapes are altered, and the final picture emerges. Most approaches to automatic image generation, however, aim to generate entire scenes at once. In the context of generative neural networks, this typically means that all the pixels are conditioned on a single latent distribution (Dayan et al., 1995; Hinton & Salakhutdinov, 2006; Larochelle & Murray, 2011). As well as precluding the possibility of iterative self-correction, the "one shot" approach is fundamentally difficult to scale to large images. The *Deep Recurrent Attentive Writer* (DRAW) architecture represents a shift towards a more natural form of image construction, in which parts of a scene are created independently from others, and approximate sketches are successively refined.

Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

The core of the DRAW architecture is a pair of recurrent neural networks: an *encoder* network that compresses the real images presented during training, and a *decoder* that reconstitutes images after receiving codes. The combined system is trained end-to-end with stochastic gradient descent, where the loss function is a variational upper bound on the log-likelihood of the data. It therefore belongs to the family of *variational auto-encoders*, a recently emerged hybrid of deep learning and variational inference that has led to significant advances in generative modelling (Gregor et al., 2014; Kingma & Welling, 2014; Rezende et al., 2014; Mnih & Gregor, 2014; Salimans et al., 2014). Where DRAW differs from its siblings is that, rather than generat-

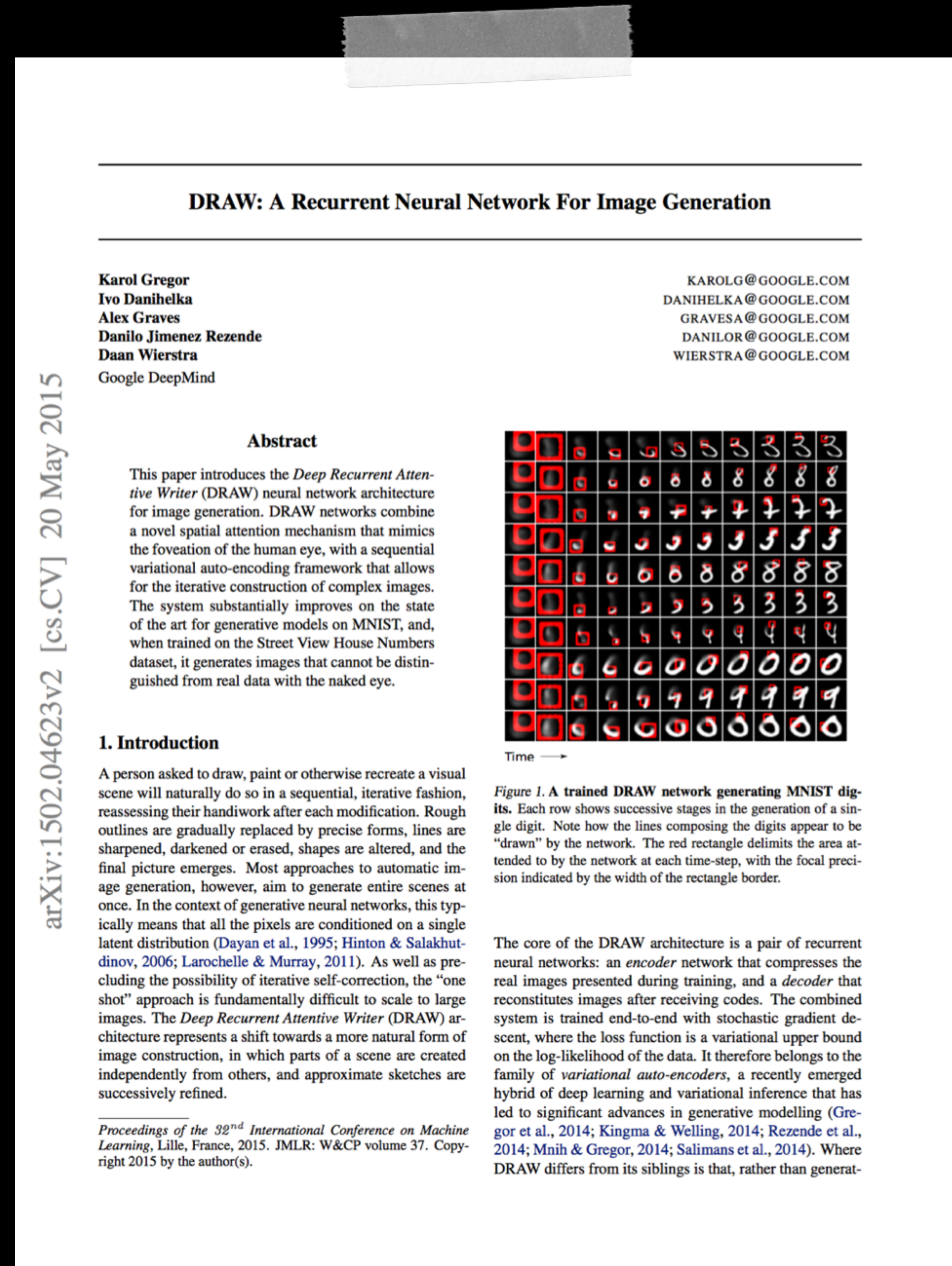
Describe images

Generate images

Goal: Generate images

"dreaming up" images - transforming random noise into an endless stream of images that the model has never even seen before

Works for MNIST and Street View House Numbers



DRAW: A Recurrent Neural Network For Image Generation

Karol Gregor KAROLG@GOOGLE.COM
Ivo Danihelka DANIHELKA@GOOGLE.COM
Alex Graves GRAVESA@GOOGLE.COM
Danilo Jimenez Rezende DANILOR@GOOGLE.COM
Daan Wierstra WIERSTRA@GOOGLE.COM
Google DeepMind

Abstract

This paper introduces the *Deep Recurrent Attentive Writer* (DRAW) neural network architecture for image generation. DRAW networks combine a novel spatial attention mechanism that mimics the foveation of the human eye, with a sequential variational auto-encoding framework that allows for the iterative construction of complex images. The system substantially improves on the state of the art for generative models on MNIST, and, when trained on the Street View House Numbers dataset, it generates images that cannot be distinguished from real data with the naked eye.

1. Introduction

A person asked to draw, paint or otherwise recreate a visual scene will naturally do so in a sequential, iterative fashion, reassessing their handiwork after each modification. Rough outlines are gradually replaced by precise forms, lines are sharpened, darkened or erased, shapes are altered, and the final picture emerges. Most approaches to automatic image generation, however, aim to generate entire scenes at once. In the context of generative neural networks, this typically means that all the pixels are conditioned on a single latent distribution (Dayan et al., 1995; Hinton & Salakhutdinov, 2006; Larochelle & Murray, 2011). As well as precluding the possibility of iterative self-correction, the "one shot" approach is fundamentally difficult to scale to large images. The *Deep Recurrent Attentive Writer* (DRAW) architecture represents a shift towards a more natural form of image construction, in which parts of a scene are created independently from others, and approximate sketches are successively refined.

Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

Time →

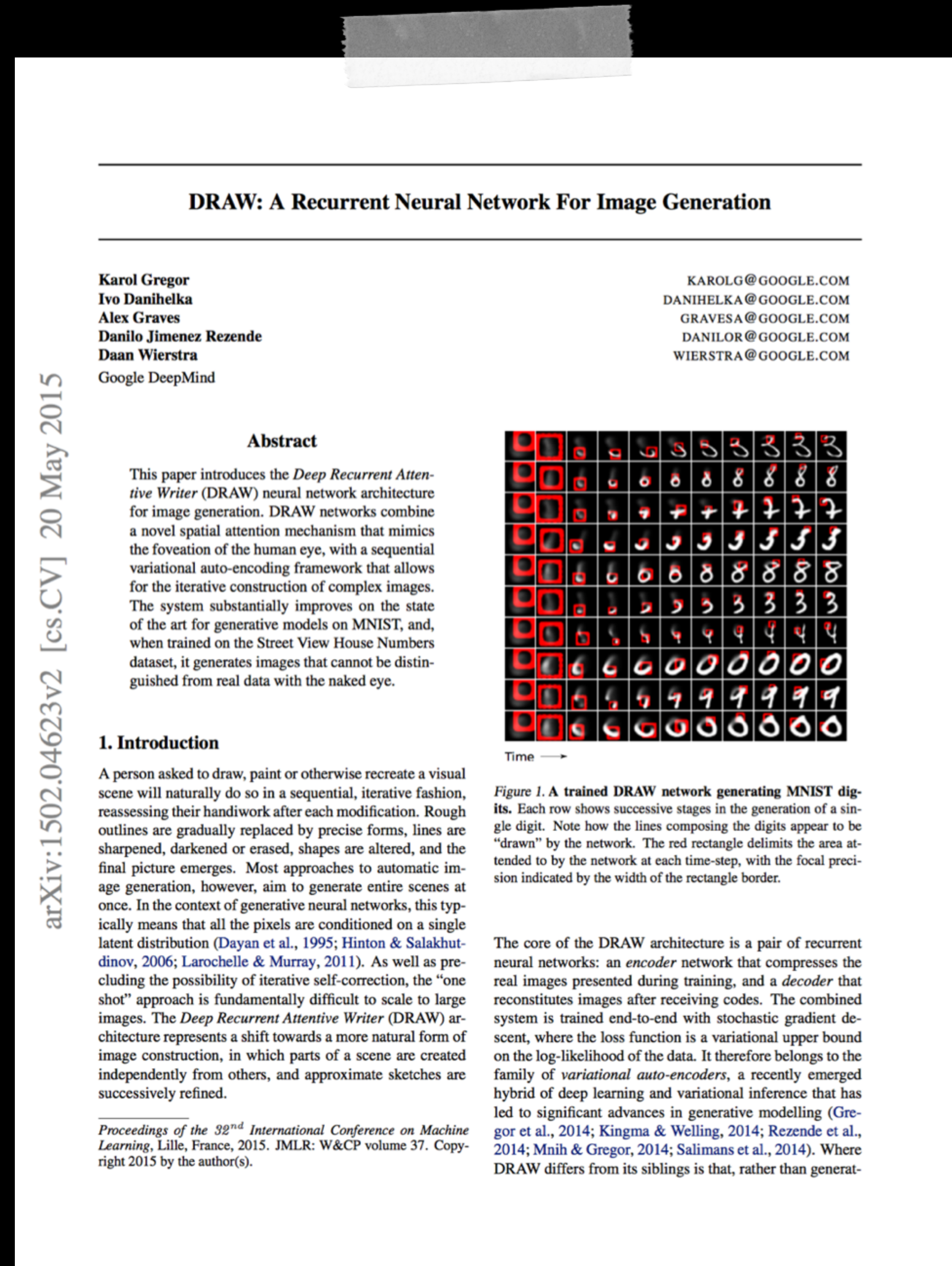
Figure 1. **A trained DRAW network generating MNIST digits.** Each row shows successive stages in the generation of a single digit. Note how the lines composing the digits appear to be "drawn" by the network. The red rectangle delimits the area attended to by the network at each time-step, with the focal precision indicated by the width of the rectangle border.

The core of the DRAW architecture is a pair of recurrent neural networks: an *encoder* network that compresses the real images presented during training, and a *decoder* that reconstitutes images after receiving codes. The combined system is trained end-to-end with stochastic gradient descent, where the loss function is a variational upper bound on the log-likelihood of the data. It therefore belongs to the family of *variational auto-encoders*, a recently emerged hybrid of deep learning and variational inference that has led to significant advances in generative modelling (Gregor et al., 2014; Kingma & Welling, 2014; Rezende et al., 2014; Mnih & Gregor, 2014; Salimans et al., 2014). Where DRAW differs from its siblings is that, rather than generat-

Variational Autoencoder

Encoder: determines a distribution that captures salient information about the input data
Decoder: samples from the distribution

Spatial selective attention mechanism that mimics the foveation of the human eye, with a sequential variational auto-encoding framework that allows for the iterative construction of complex images



DRAW: A Recurrent Neural Network For Image Generation

Karol Gregor KAROLG@GOOGLE.COM
Ivo Danihelka DANIHELKA@GOOGLE.COM
Alex Graves GRAVESA@GOOGLE.COM
Danilo Jimenez Rezende DANILOR@GOOGLE.COM
Daan Wierstra WIERSTRA@GOOGLE.COM
Google DeepMind

Abstract

This paper introduces the *Deep Recurrent Attentive Writer* (DRAW) neural network architecture for image generation. DRAW networks combine a novel spatial attention mechanism that mimics the foveation of the human eye, with a sequential variational auto-encoding framework that allows for the iterative construction of complex images. The system substantially improves on the state of the art for generative models on MNIST, and, when trained on the Street View House Numbers dataset, it generates images that cannot be distinguished from real data with the naked eye.

Time →

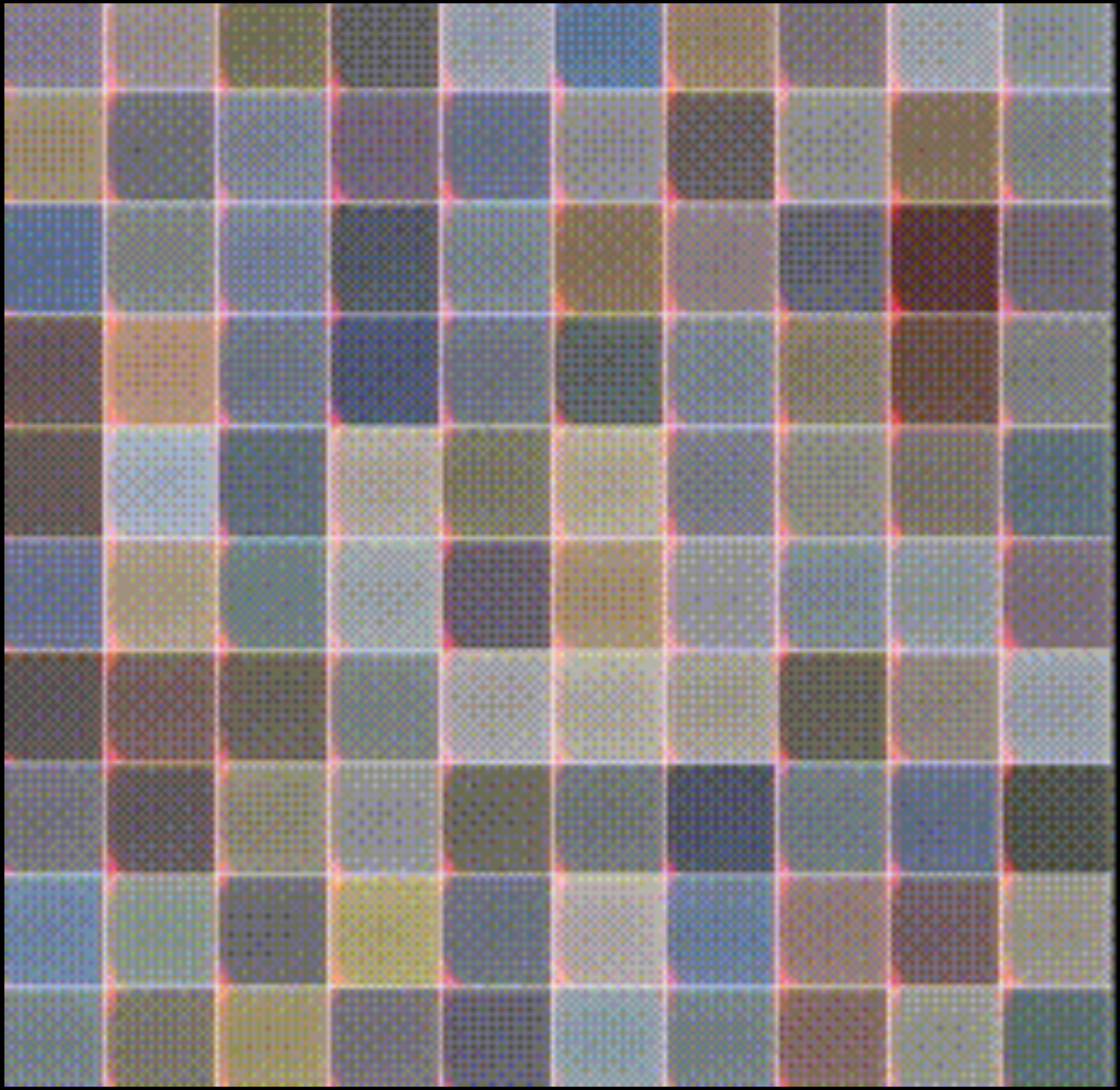
Figure 1. **A trained DRAW network generating MNIST digits.** Each row shows successive stages in the generation of a single digit. Note how the lines composing the digits appear to be "drawn" by the network. The red rectangle delimits the area attended to by the network at each time-step, with the focal precision indicated by the width of the rectangle border.

1. Introduction

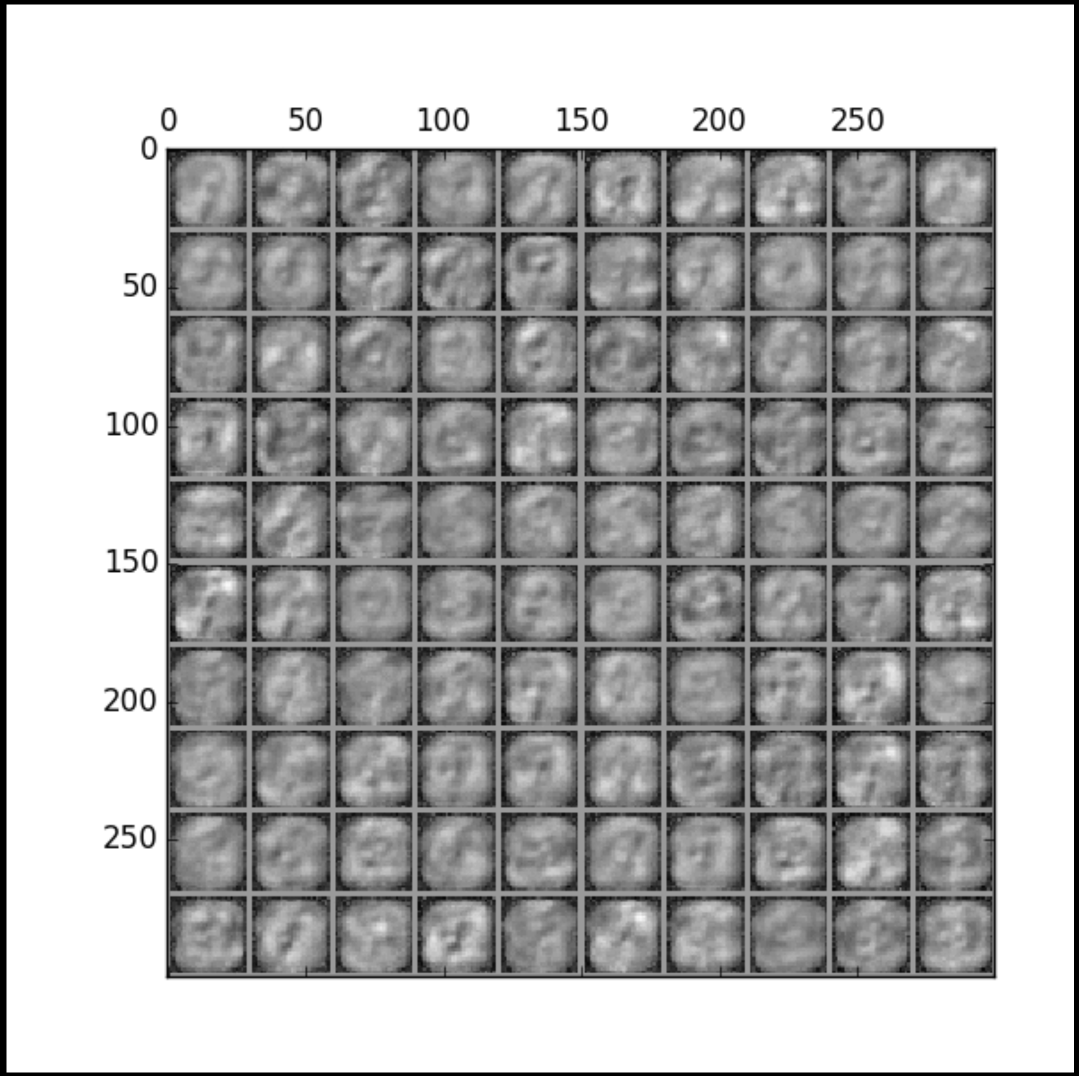
A person asked to draw, paint or otherwise recreate a visual scene will naturally do so in a sequential, iterative fashion, reassessing their handiwork after each modification. Rough outlines are gradually replaced by precise forms, lines are sharpened, darkened or erased, shapes are altered, and the final picture emerges. Most approaches to automatic image generation, however, aim to generate entire scenes at once. In the context of generative neural networks, this typically means that all the pixels are conditioned on a single latent distribution (Dayan et al., 1995; Hinton & Salakhutdinov, 2006; Larochelle & Murray, 2011). As well as precluding the possibility of iterative self-correction, the "one shot" approach is fundamentally difficult to scale to large images. The *Deep Recurrent Attentive Writer* (DRAW) architecture represents a shift towards a more natural form of image construction, in which parts of a scene are created independently from others, and approximate sketches are successively refined.

The core of the DRAW architecture is a pair of recurrent neural networks: an *encoder* network that compresses the real images presented during training, and a *decoder* that reconstitutes images after receiving codes. The combined system is trained end-to-end with stochastic gradient descent, where the loss function is a variational upper bound on the log-likelihood of the data. It therefore belongs to the family of *variational auto-encoders*, a recently emerged hybrid of deep learning and variational inference that has led to significant advances in generative modelling (Gregor et al., 2014; Kingma & Welling, 2014; Rezende et al., 2014; Mnih & Gregor, 2014; Salimans et al., 2014). Where DRAW differs from its siblings is that, rather than generat-

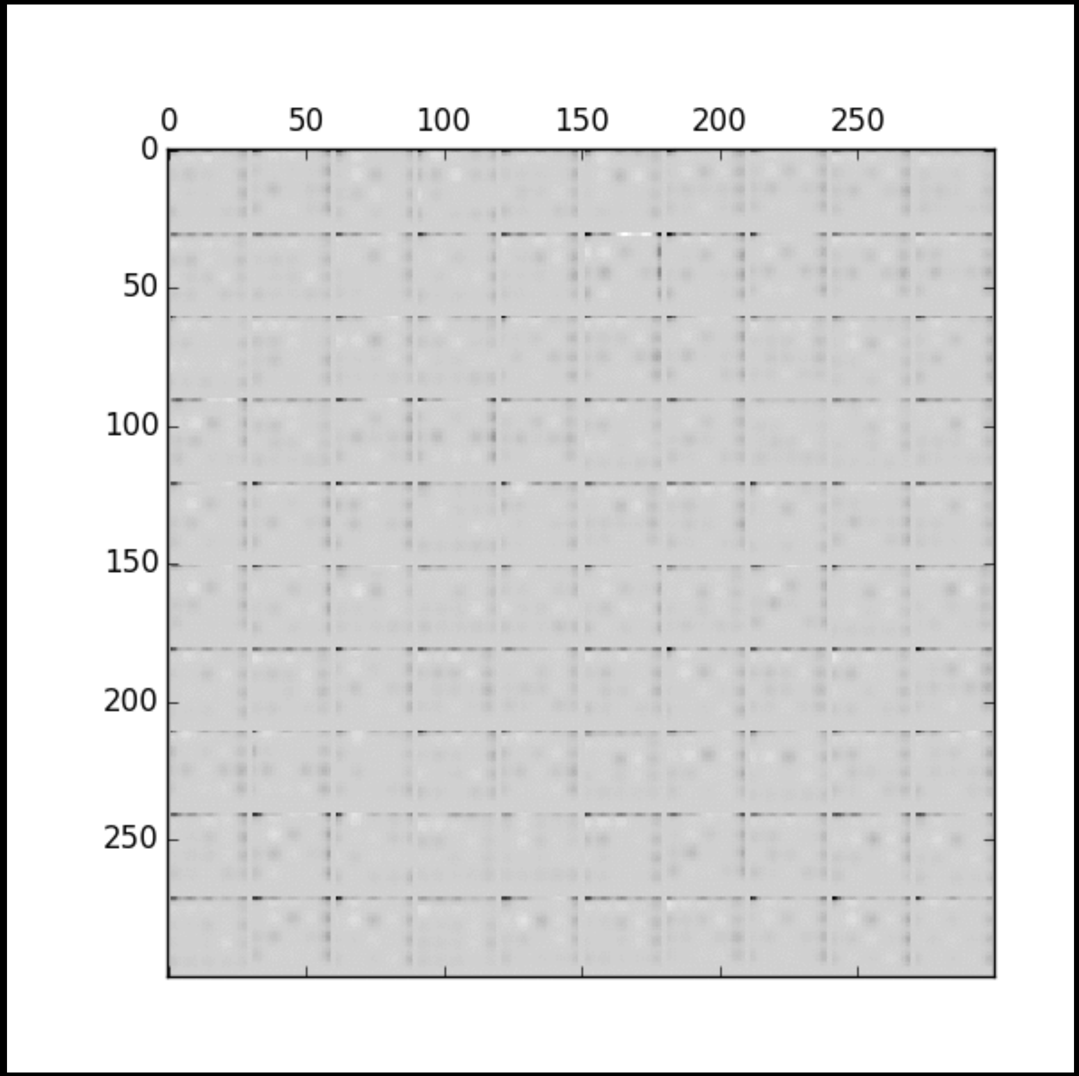
Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).



without attention



with attention



network decides at every step "where to read", "where to write", and "what to write"

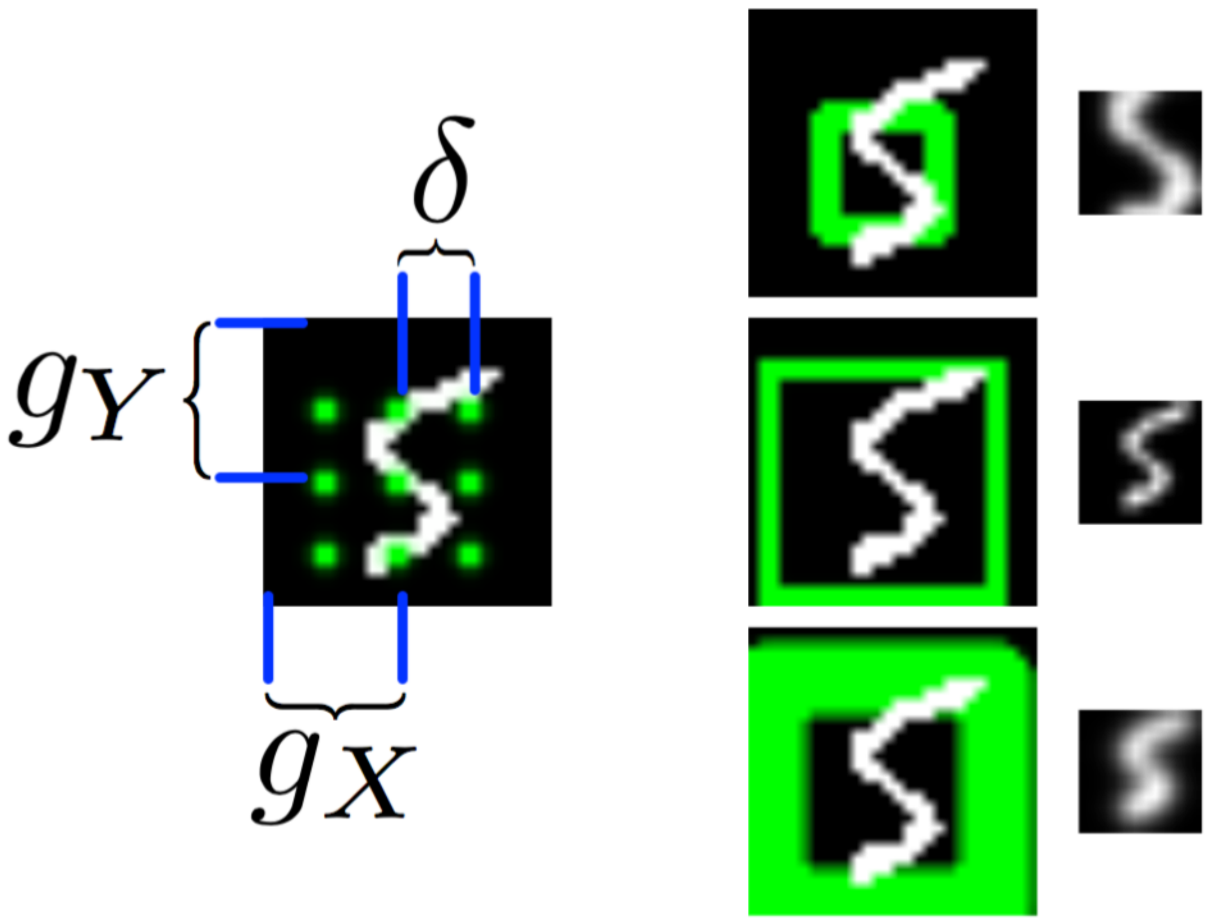


Figure 3. **Left:** A 3×3 grid of filters superimposed on an image. The stride (δ) and centre location (g_X, g_Y) are indicated. **Right:** Three $N \times N$ patches extracted from the image ($N = 12$). The green rectangles on the left indicate the boundary and precision (σ) of the patches, while the patches themselves are shown to the right. The top patch has a small δ and high σ, giving a zoomed-in but blurry view of the centre of the digit; the middle patch has large δ and low σ, effectively downsampling the whole image; and the bottom patch has high δ and σ.

Selective Attention Model

An N×N grid of Gaussian filters is positioned on the image by specifying the co-ordinates of the grid centre and the stride distance between adjacent filters.

stride / delta = zoom

it starts covering the entire image and then zooms in

Implementations

Show, Attend, and Tell

https://github.com/jazzsaxmafia/show_attend_and_tell.tensorflow/

DRAW

https://github.com/ikostrikov/TensorFlow-VAE-GAN-DRAW

https://github.com/ericjang/draw

Great blog post about DRAW

http://evjang.com/articles/draw

Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering

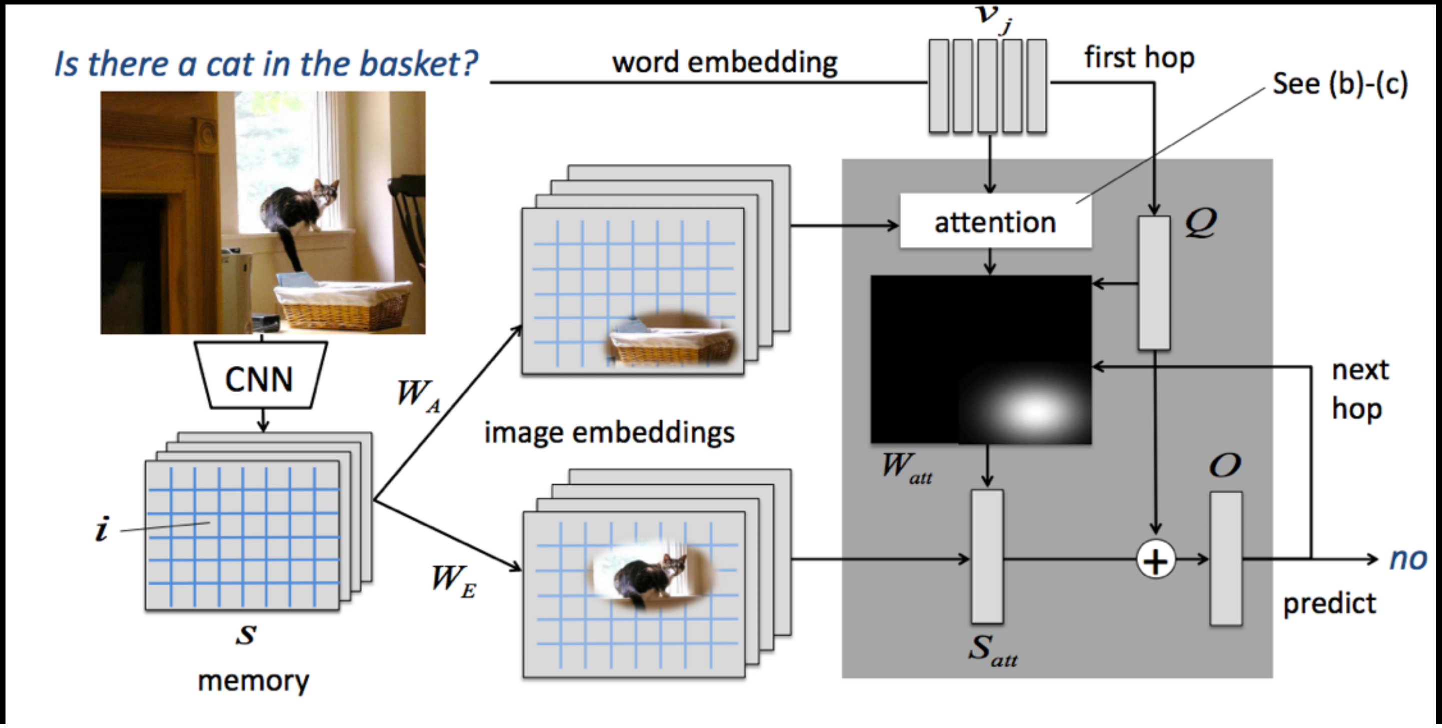


Figure 6. Visualization of the spatial attention weights in the one-hop and two-hop model on VQA (top two rows) and DAQUAR (bottom row) datasets. For each image and question pair, we show the original image, the attention weights W_{att} of the one-hop model, and the two attention weights W_{att} and W_{att2} of the two-hop model in order.

Image Captioning & Attention

Deep Learning

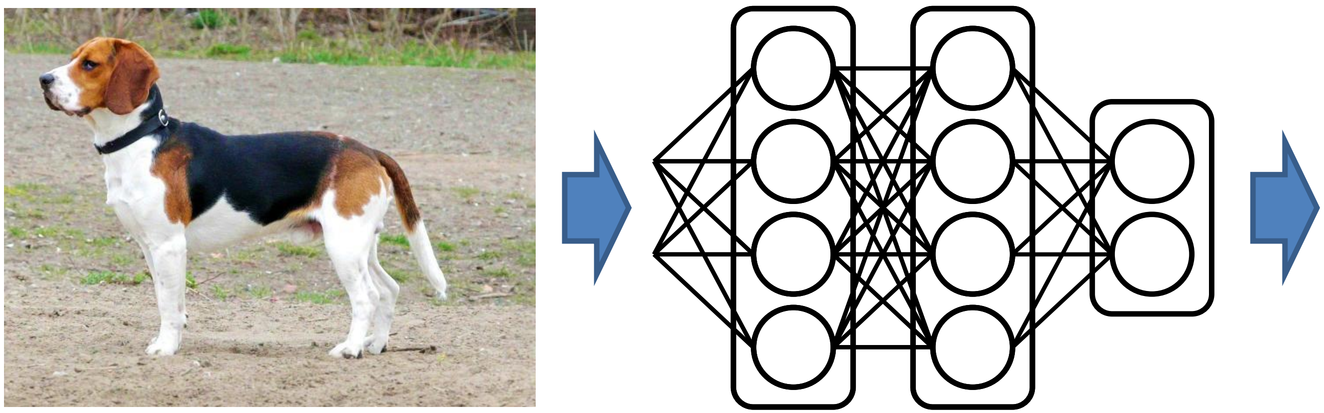
Hendrik Heuer
University of Amsterdam

Unsupervised Visual Representation Learning by Context Prediction

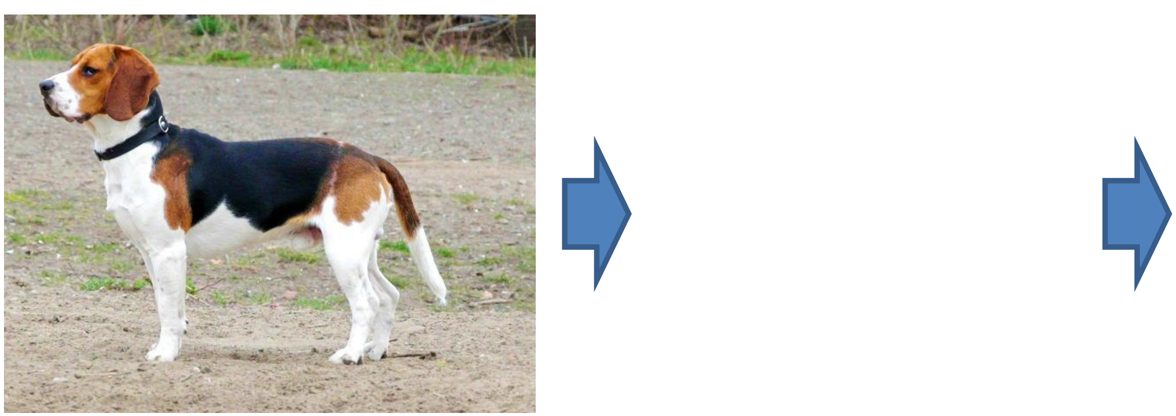
Carl Doersch

Joint work with Alexei A. Efros & Abhinav Gupta

ImageNet + Deep Learning

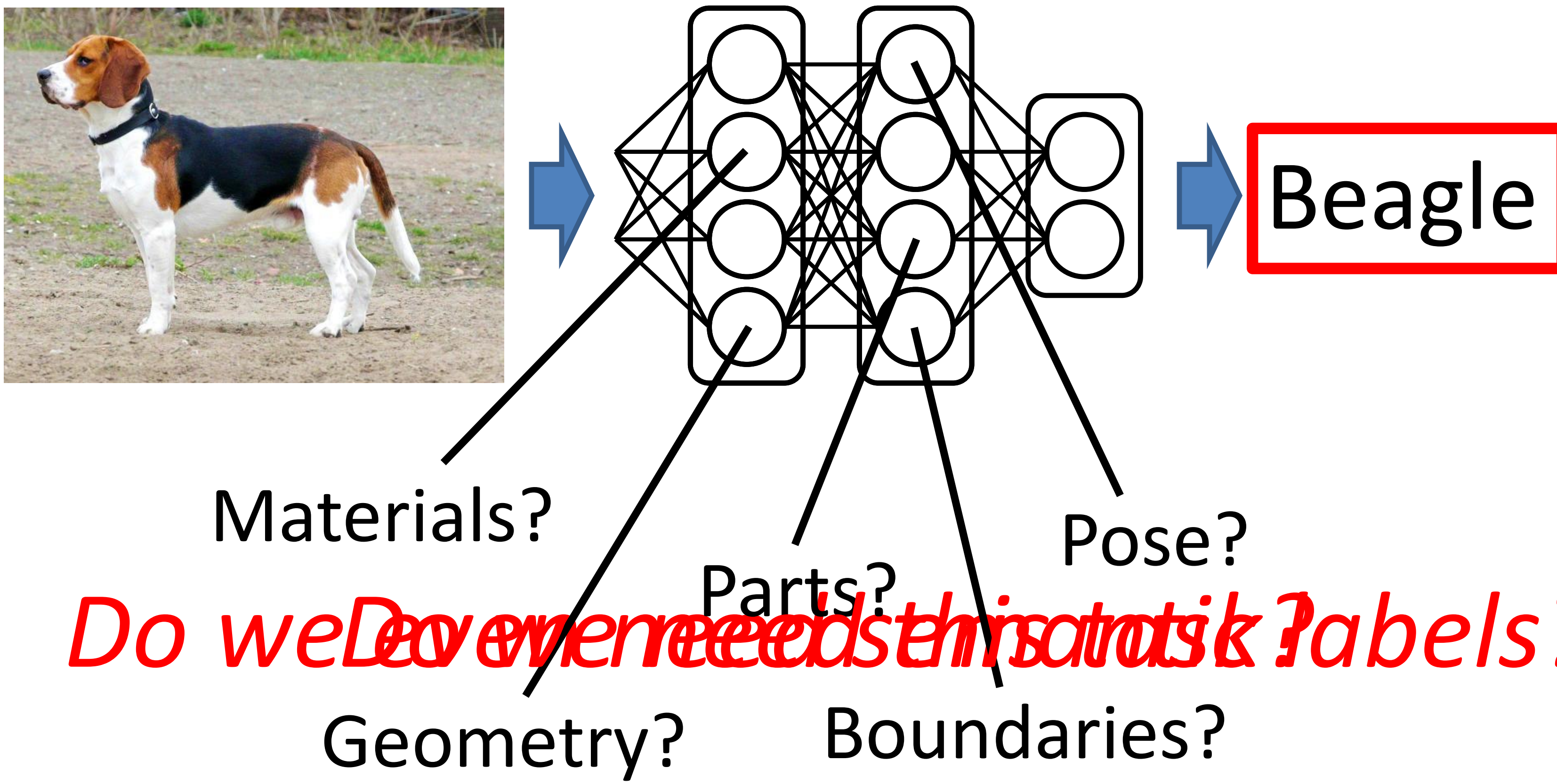


Beagle



- Image Retrieval
- Detection (RCNN)
- Segmentation (FCN)
- Depth Estimation
- ...

ImageNet + Deep Learning

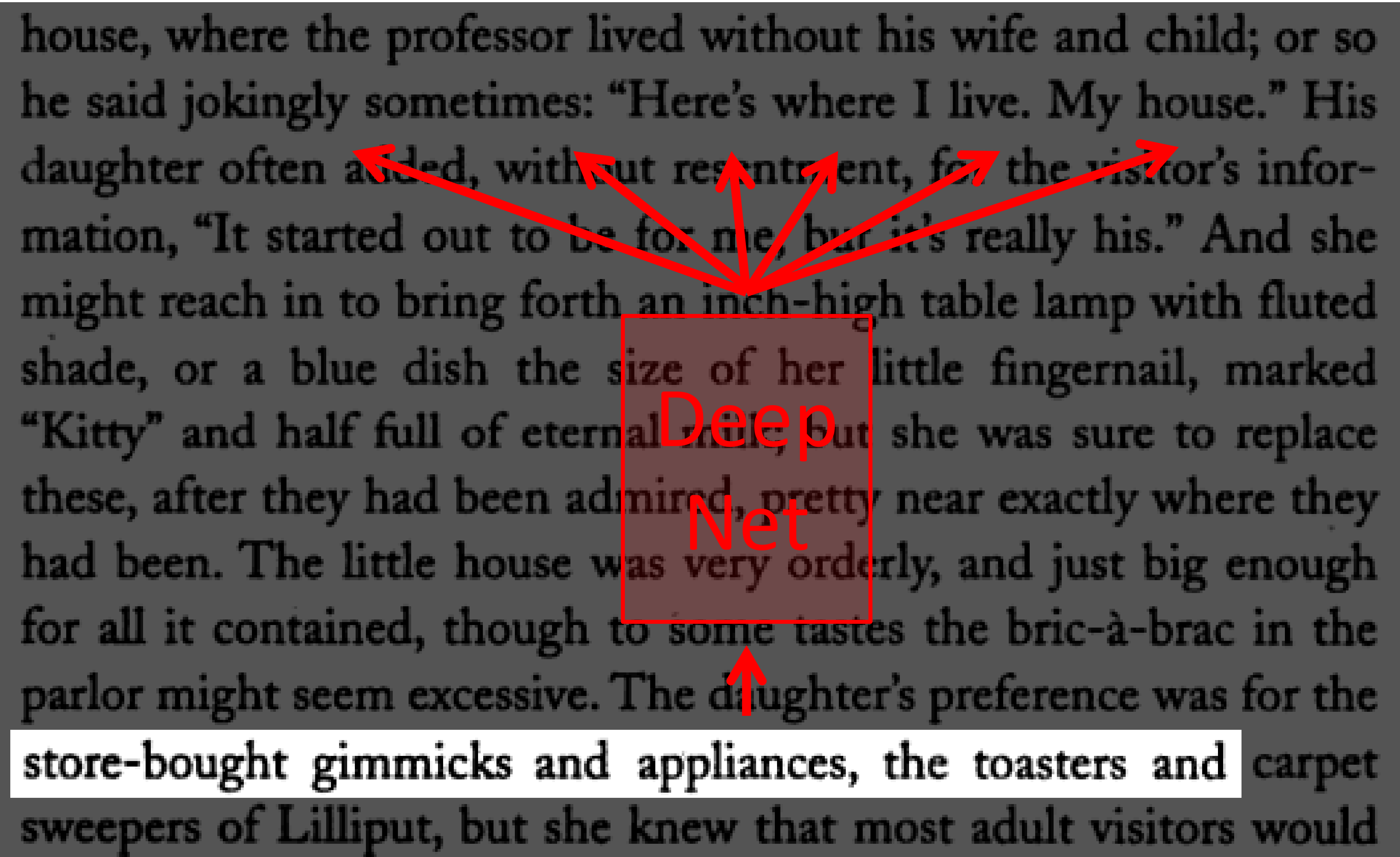


Do we even need semantic labels?

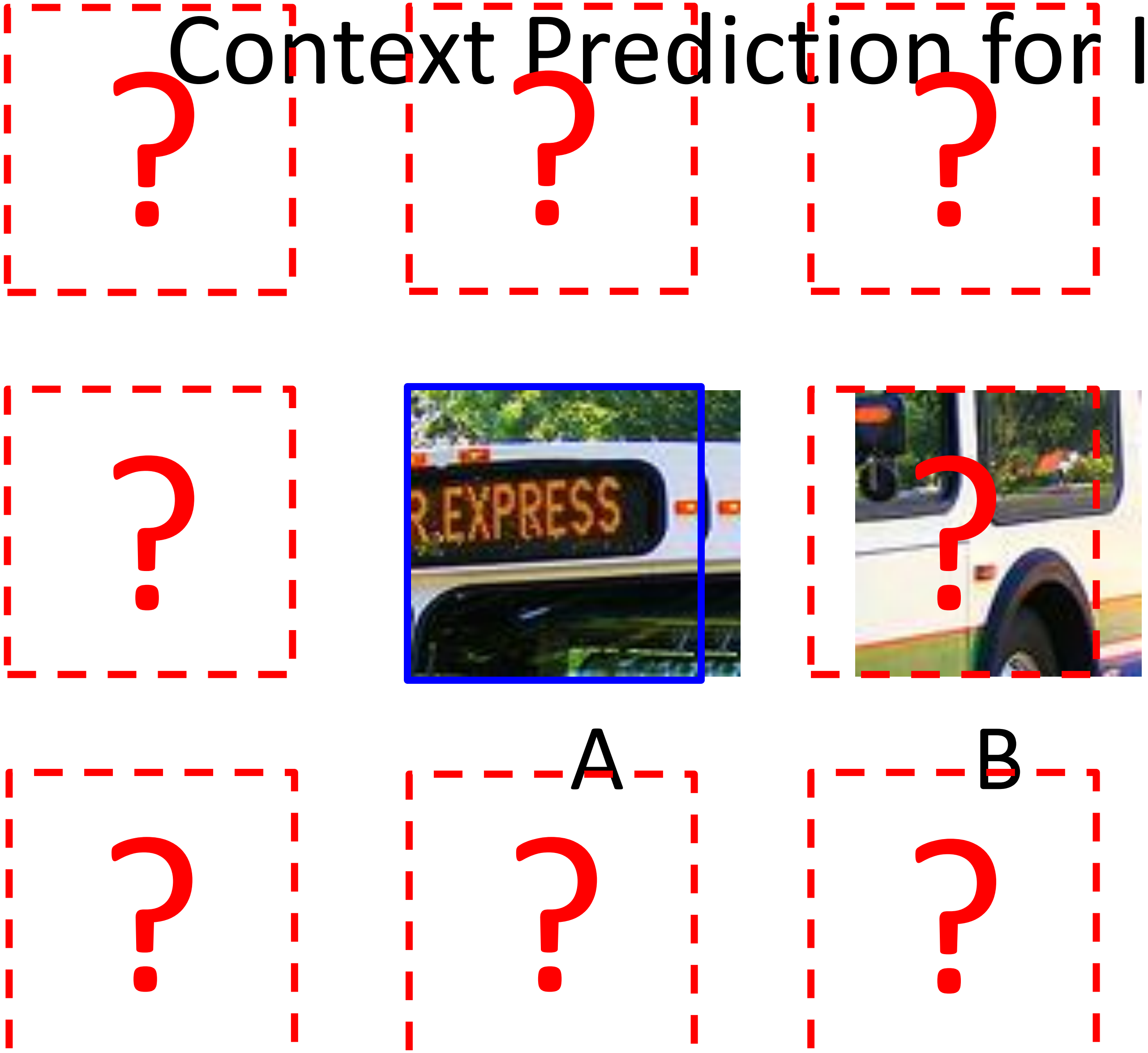
Do we need this task?

Context as Supervision

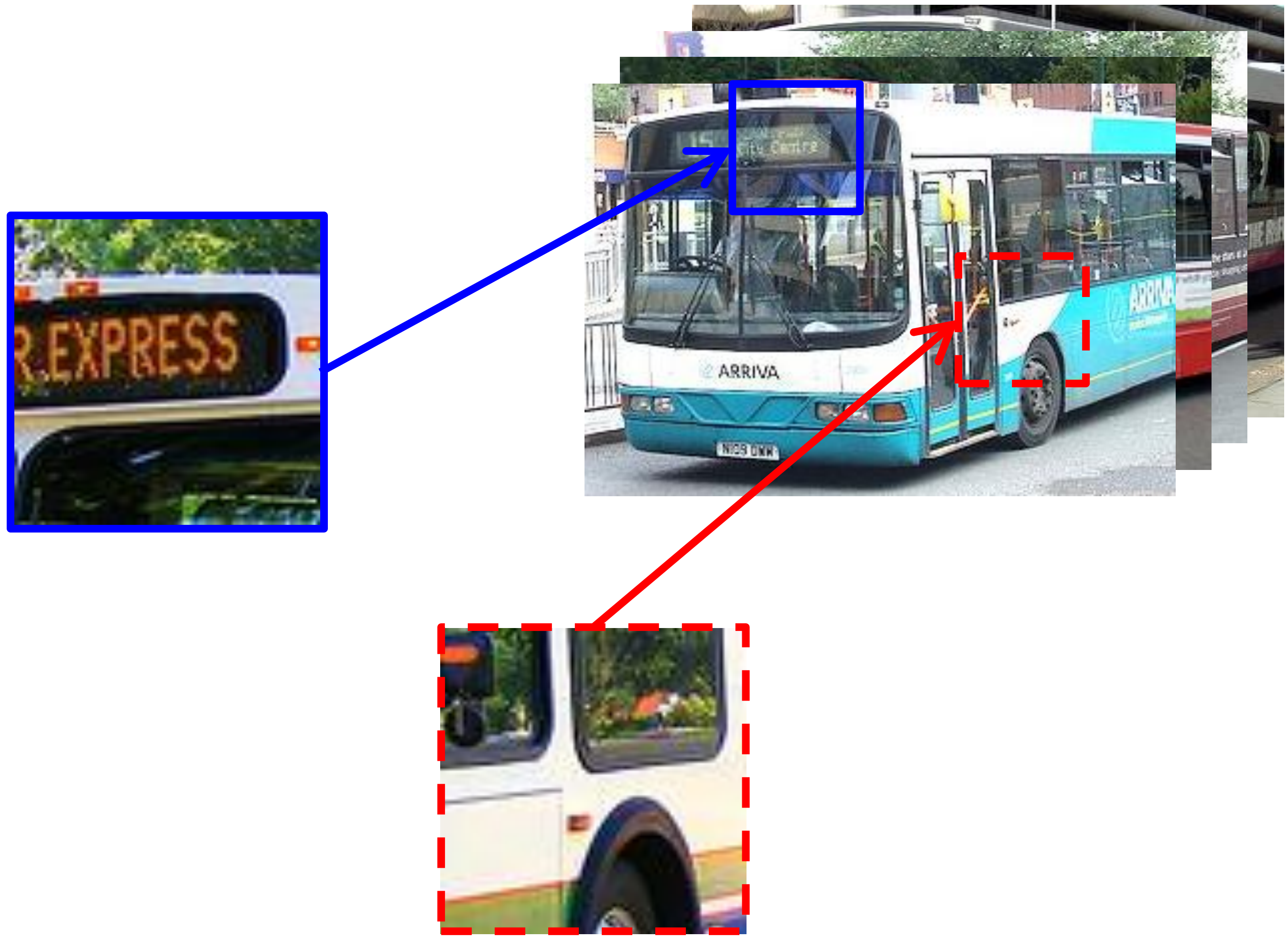
[Collobert & Weston 2008; Mikolov et al. 2013]



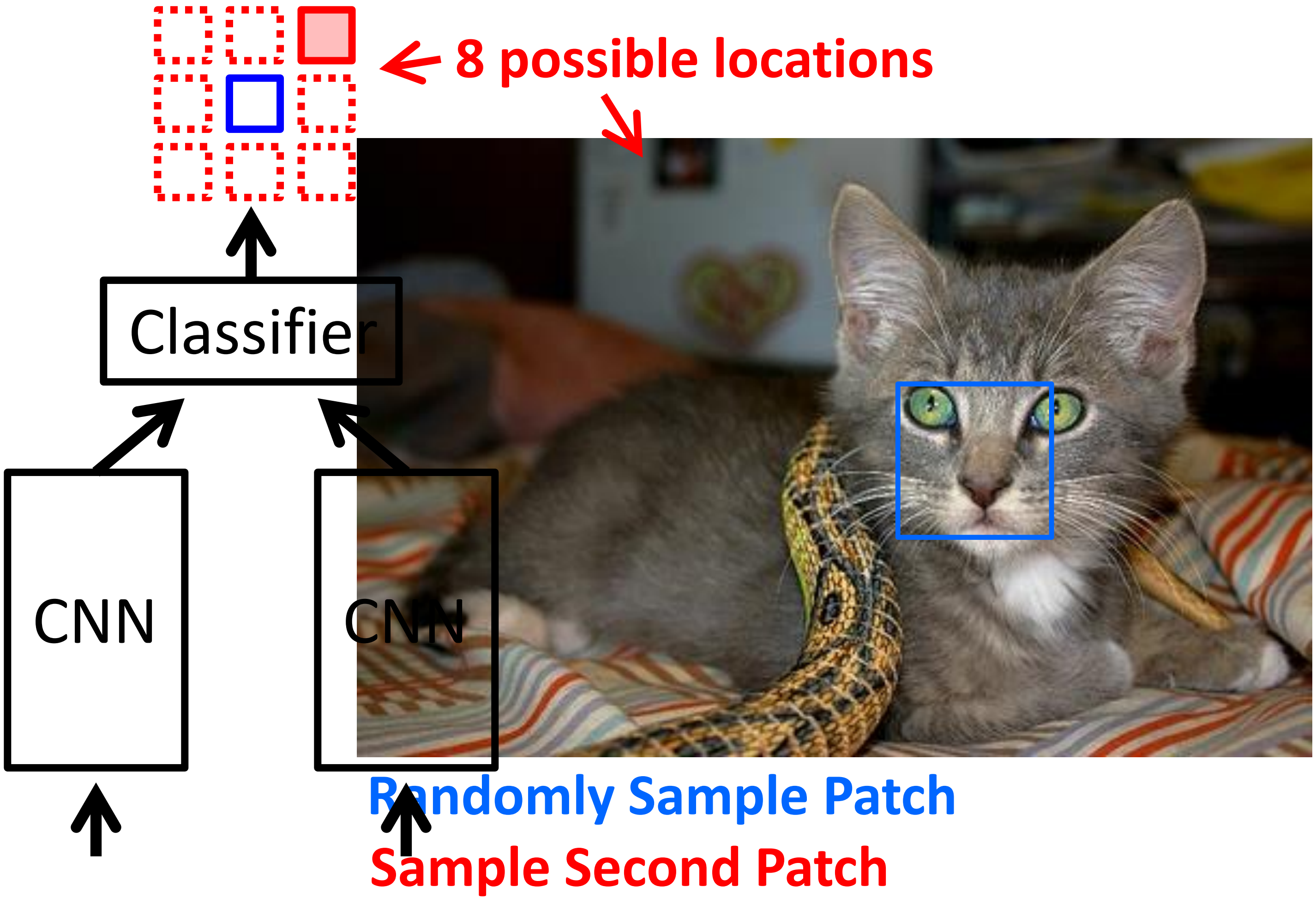
Context Prediction for Images

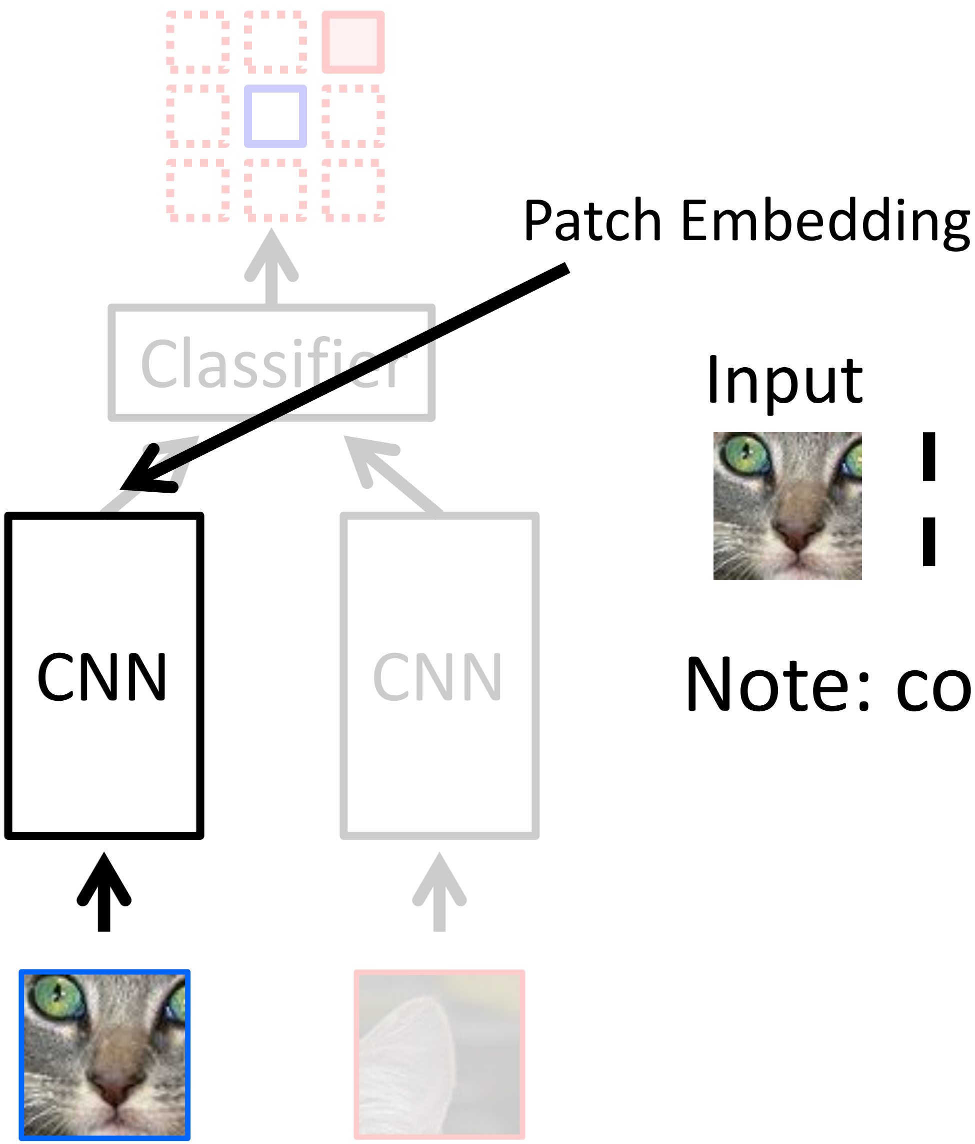


Semantics from a non-semantic task



Relative Position Task

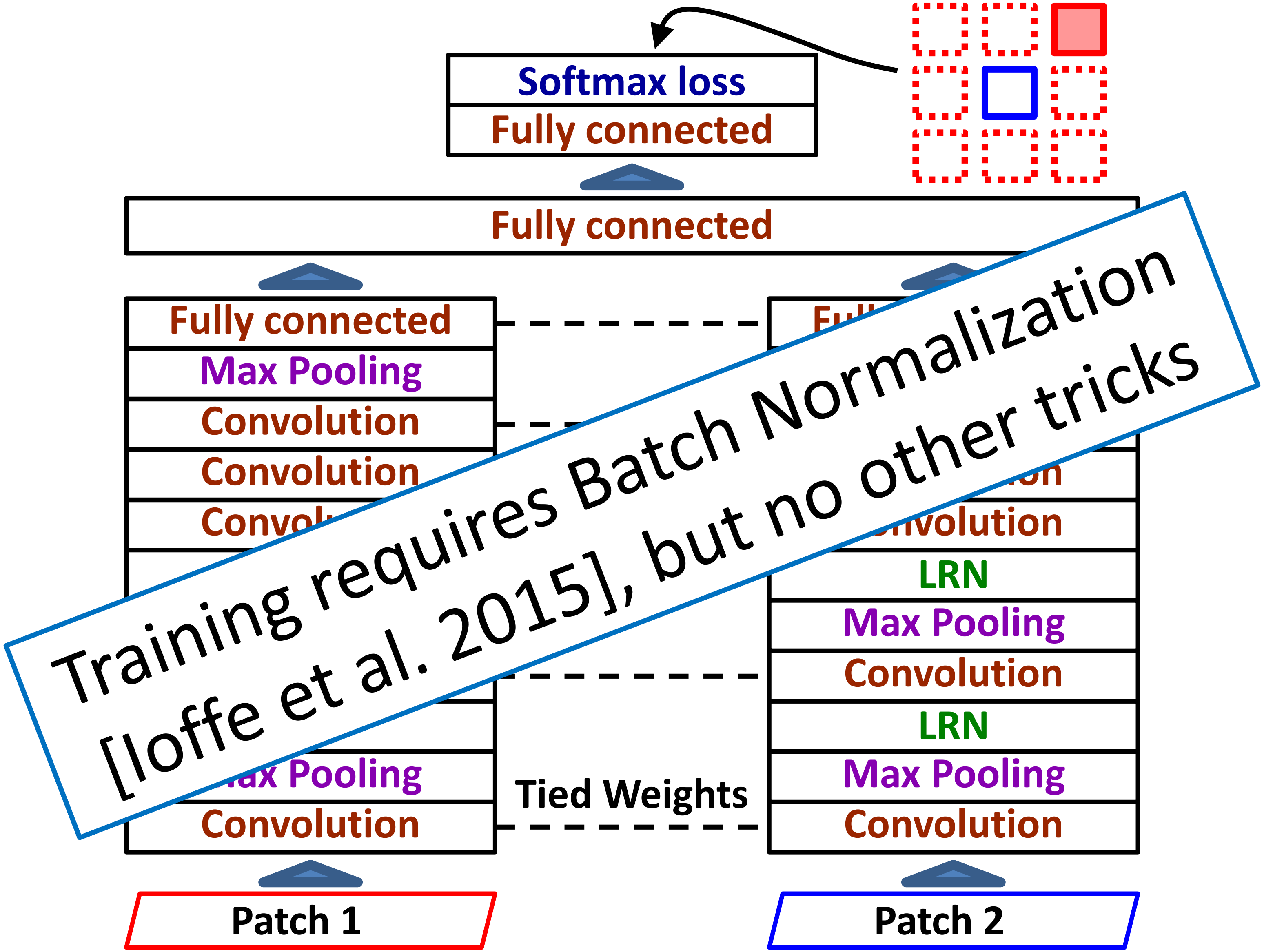




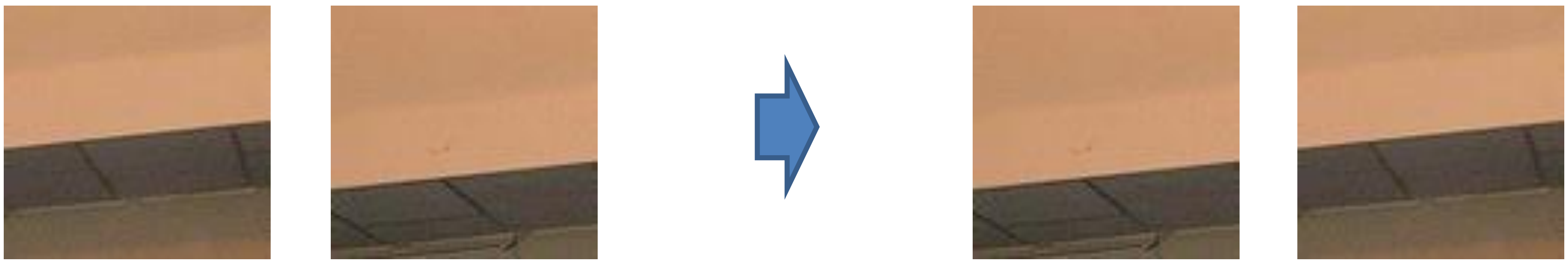
Input | Nearest Neighbors

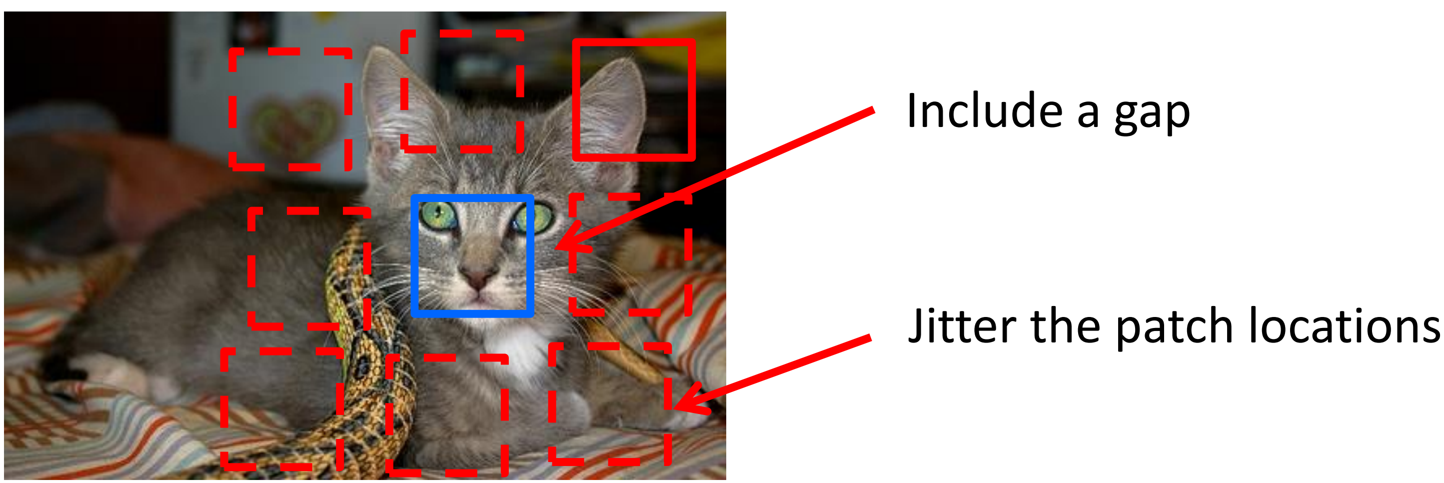
Note: connects ***across*** instances!

Architecture

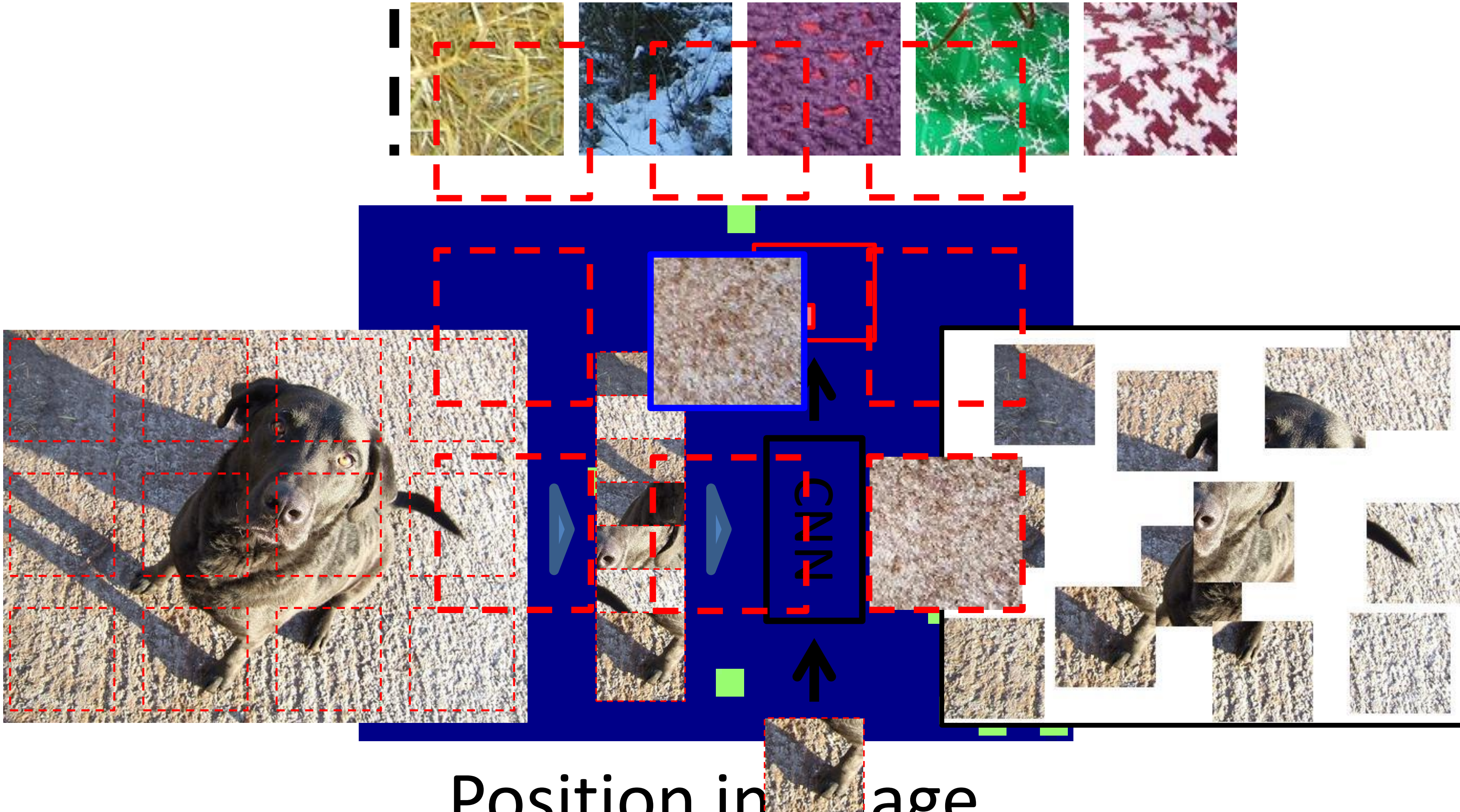


Avoiding Trivial Shortcuts



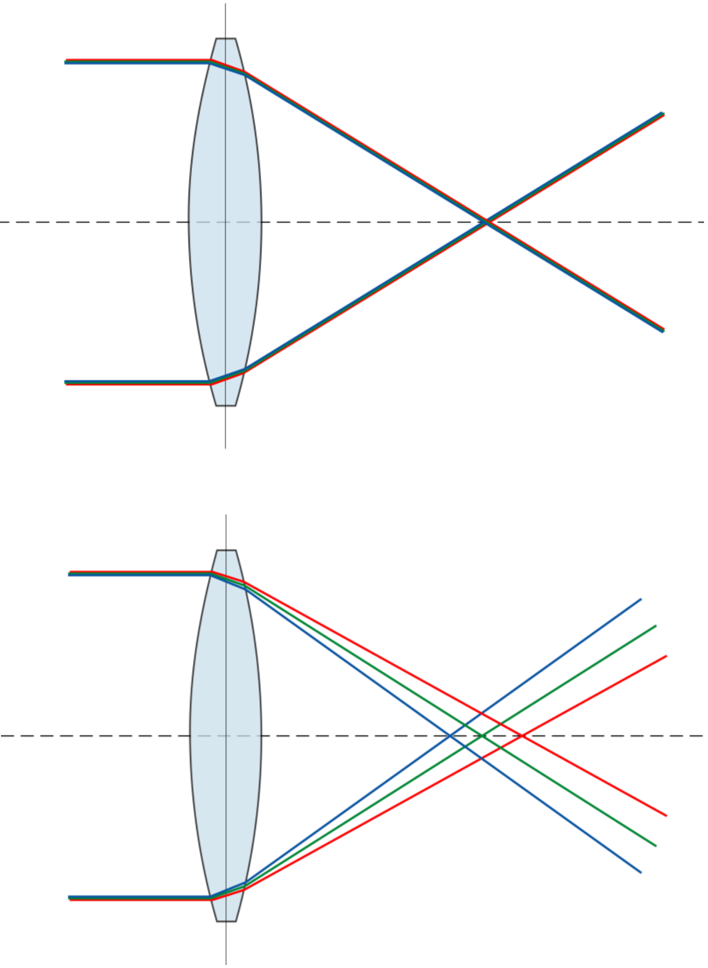


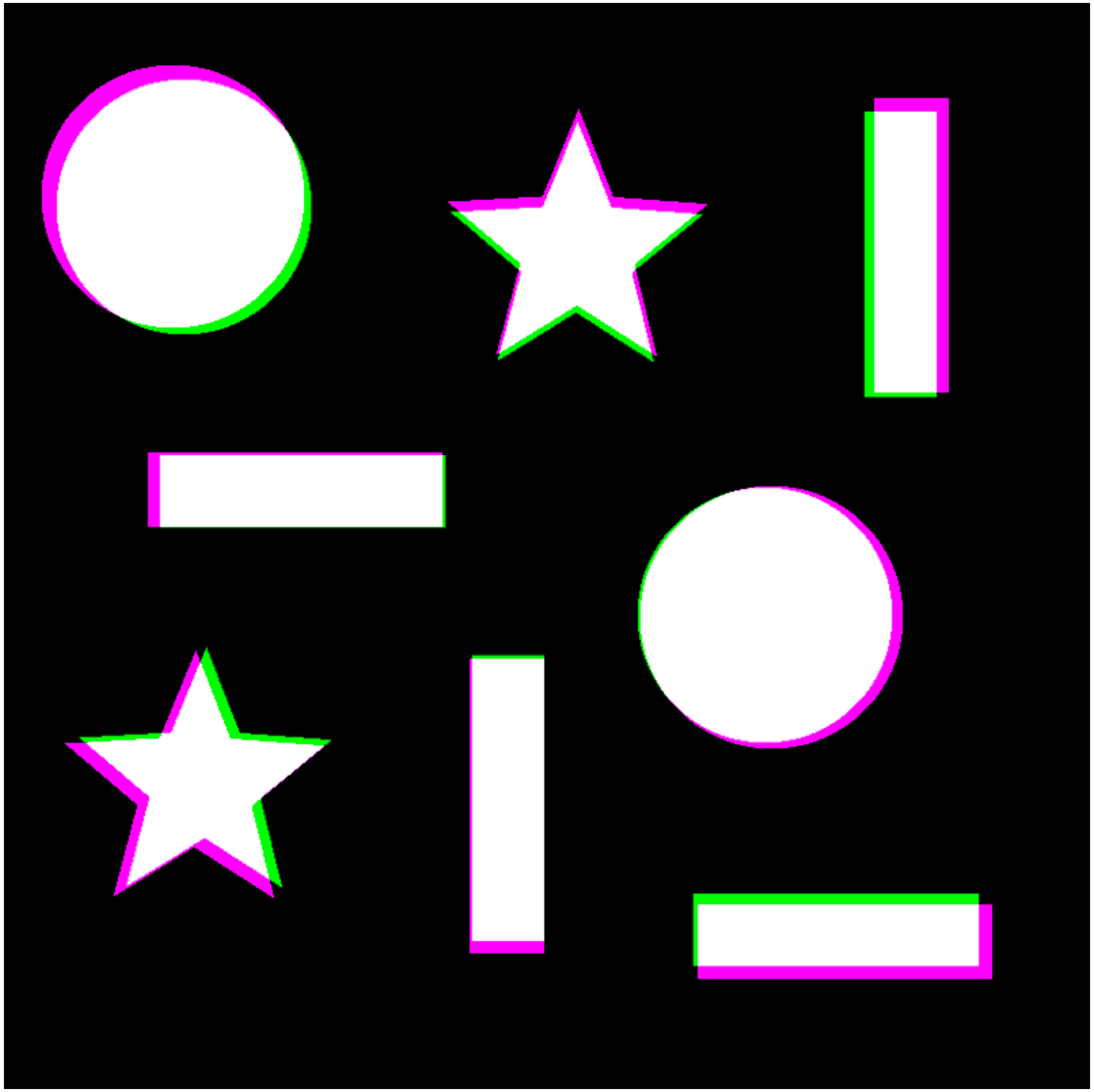
A Not-So “Trivial” Shortcut



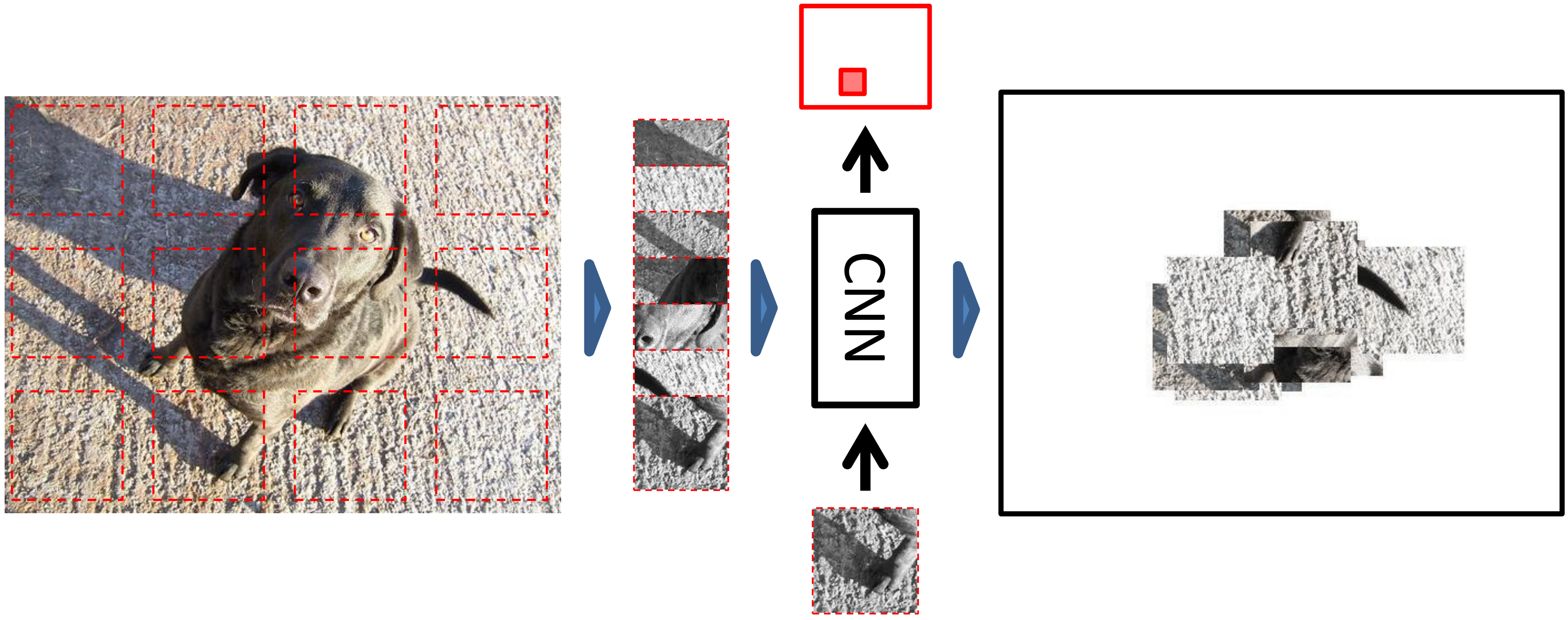
Position in image

Chromatic Aberration

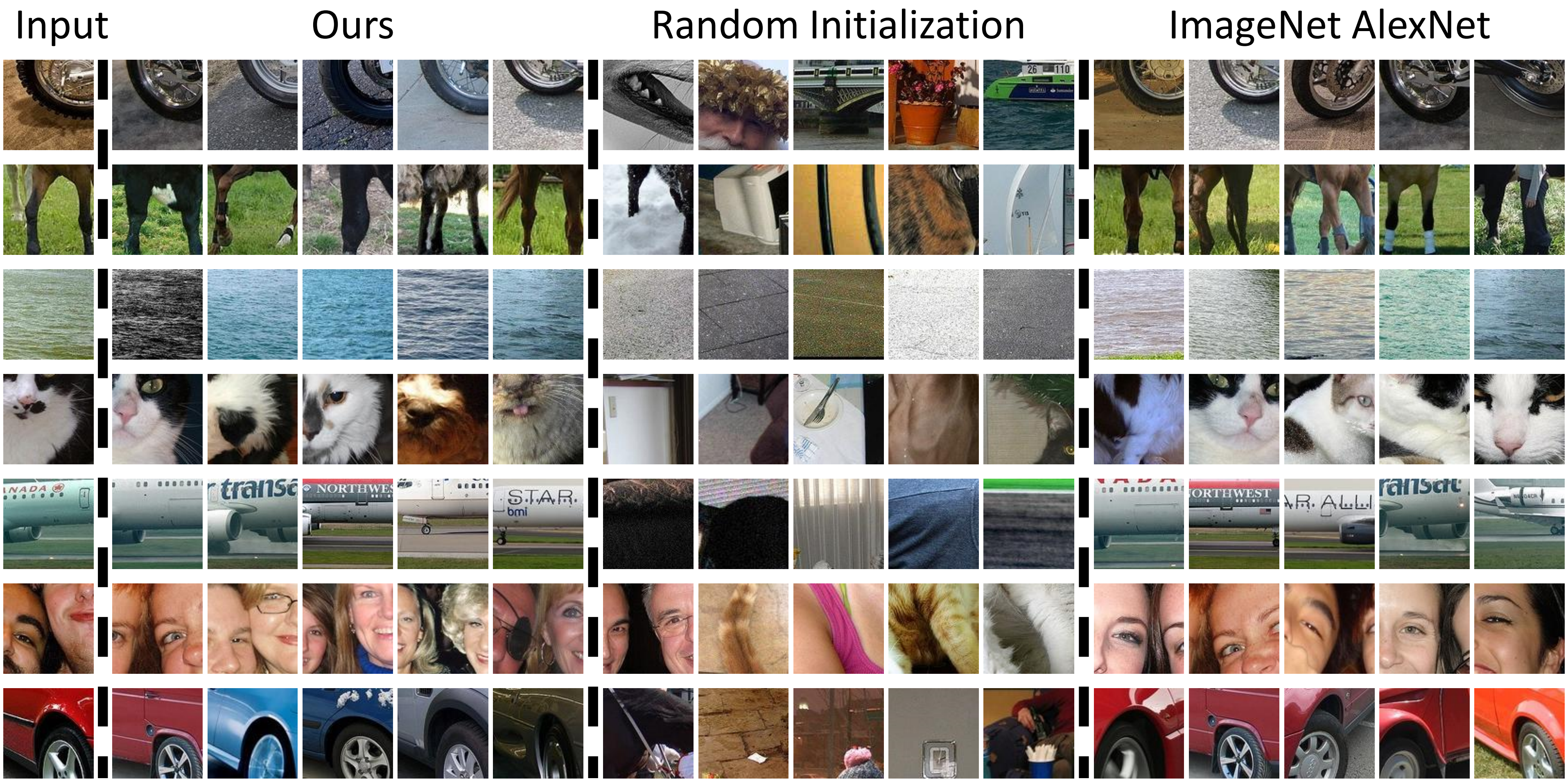




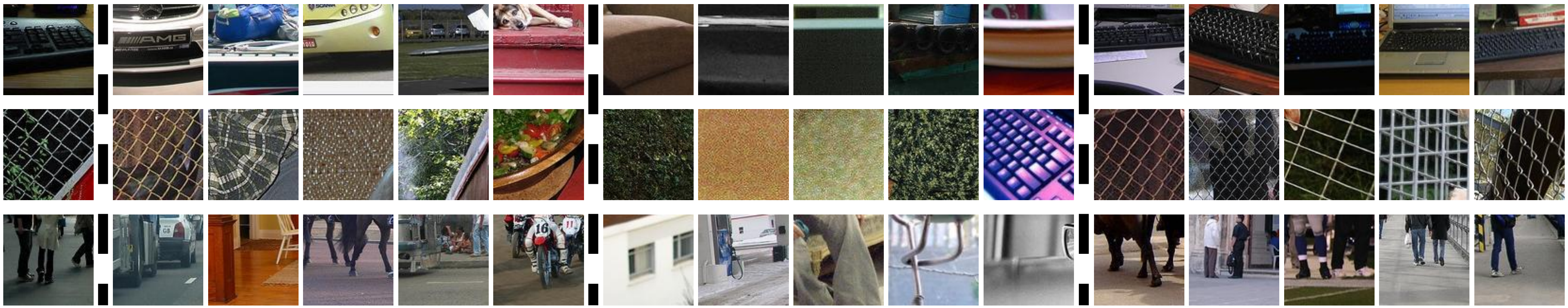
Chromatic Aberration



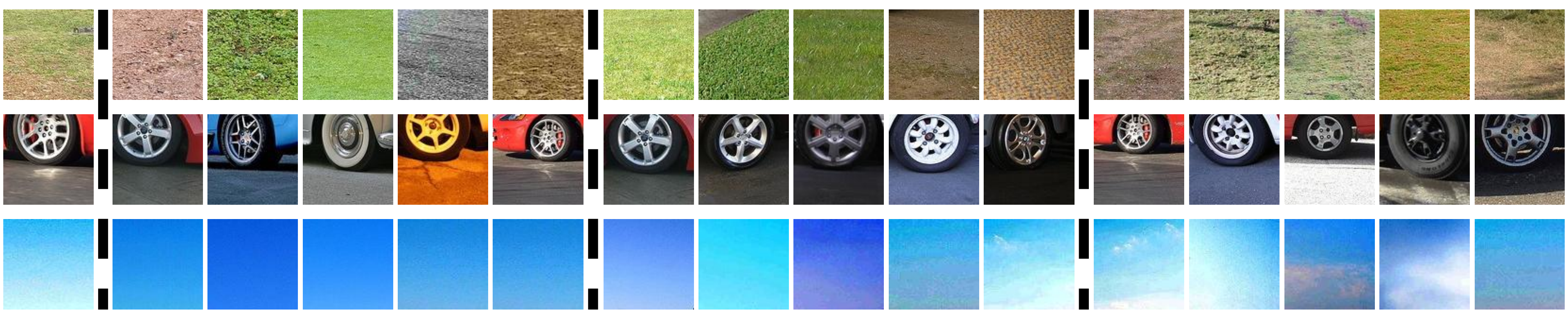
What is learned?



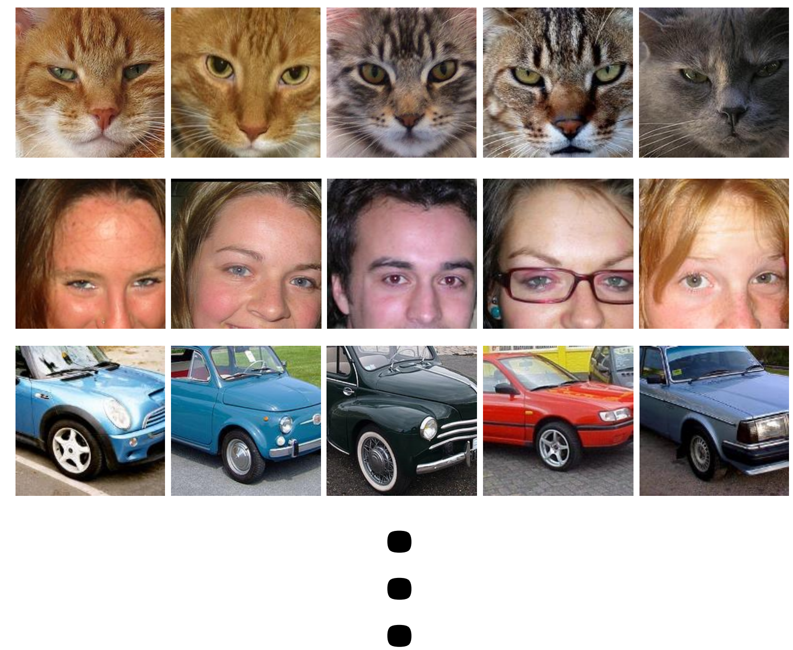
Still don't capture everything



You don't always need to learn!



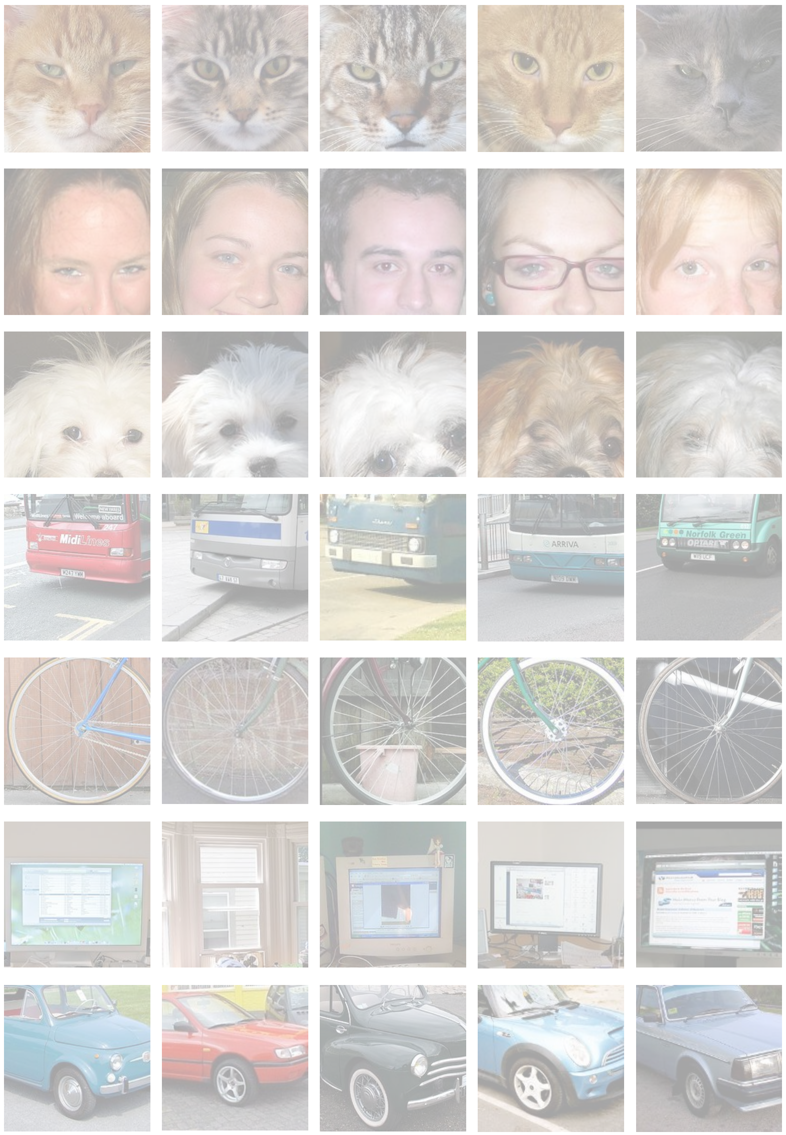
Visual Data Mining

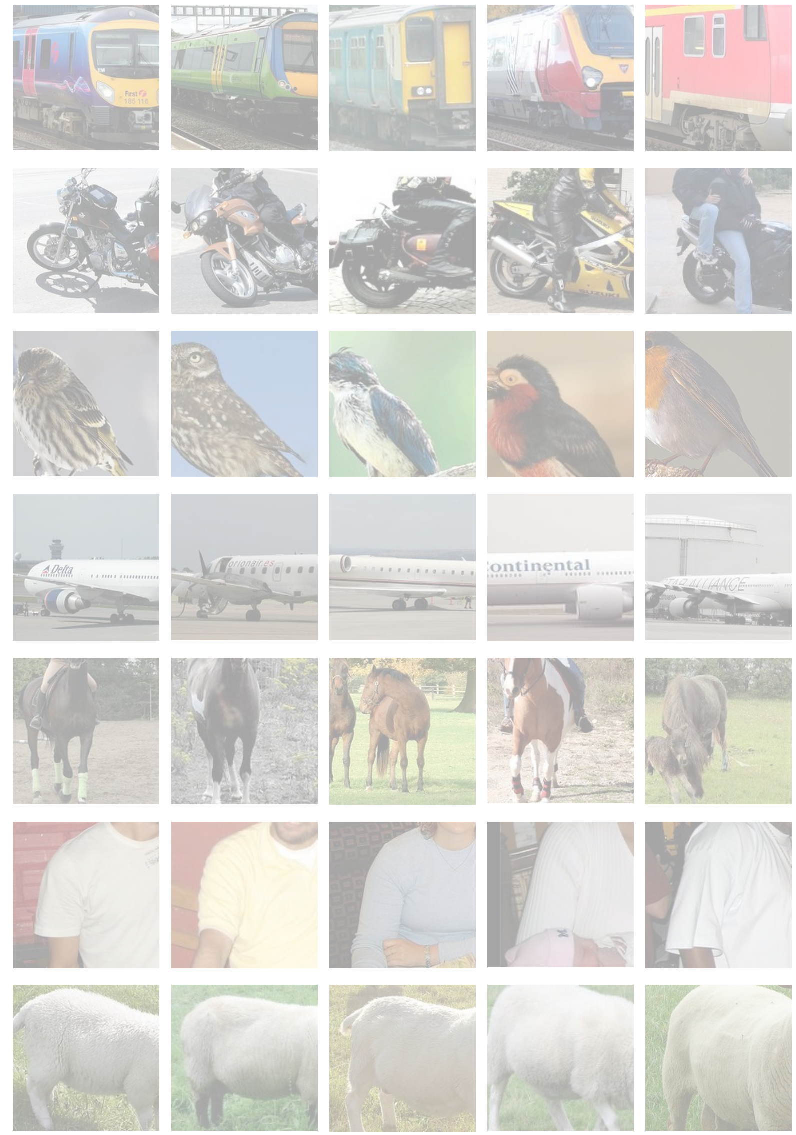


Via Geometric Verification

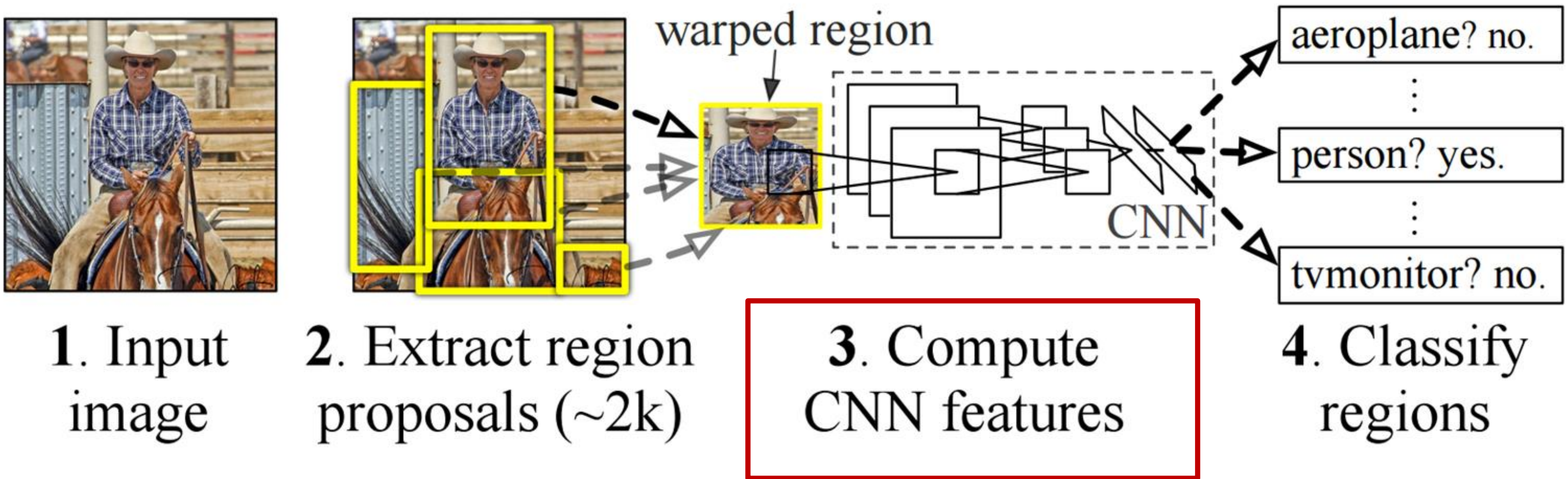
Simplified from [Chum et al 2007]

Mined from Pascal VOC2011





Pre-Training for R-CNN

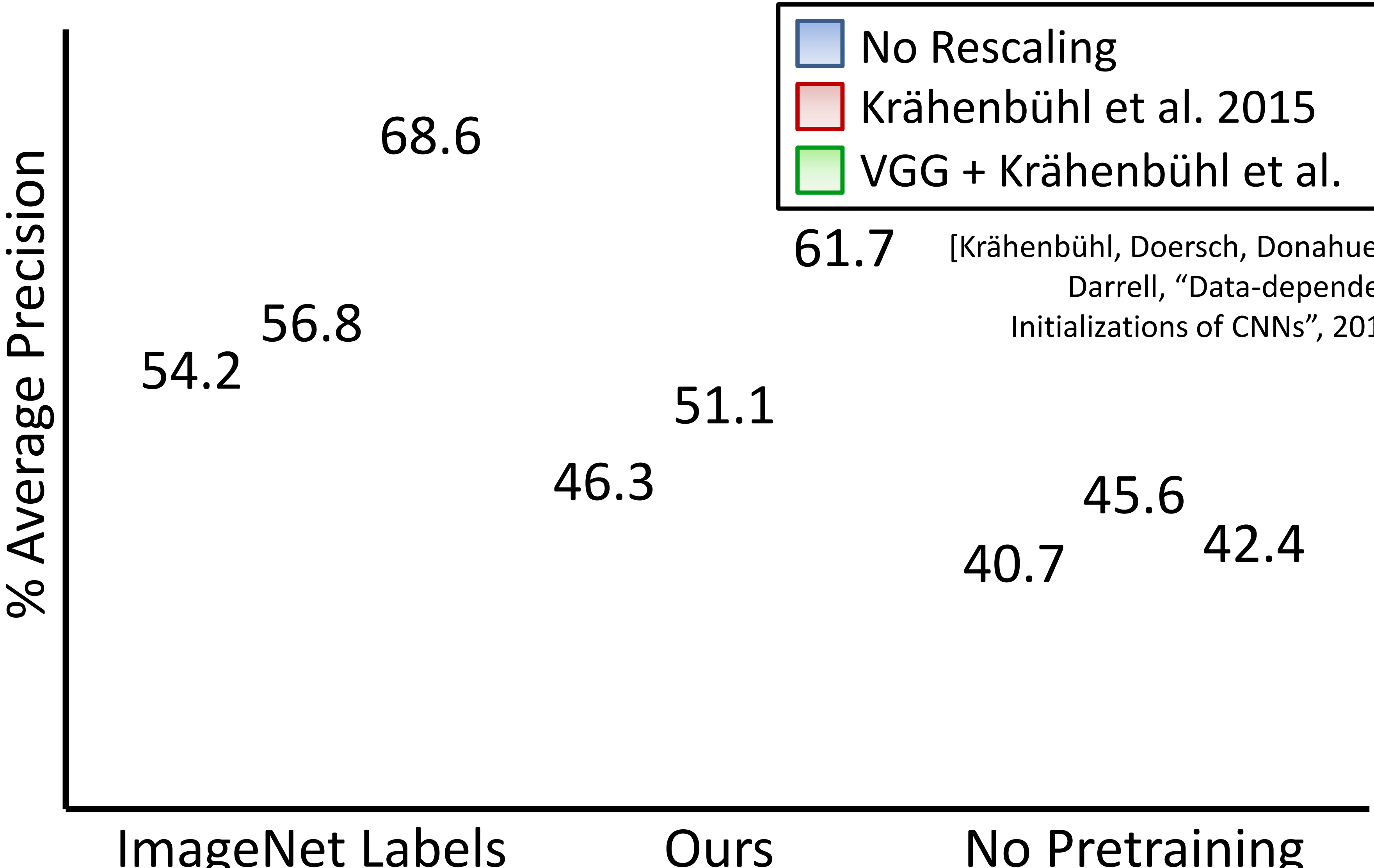


Pre-train on relative-position task, w/o labels

[Girshick et al. 2014]

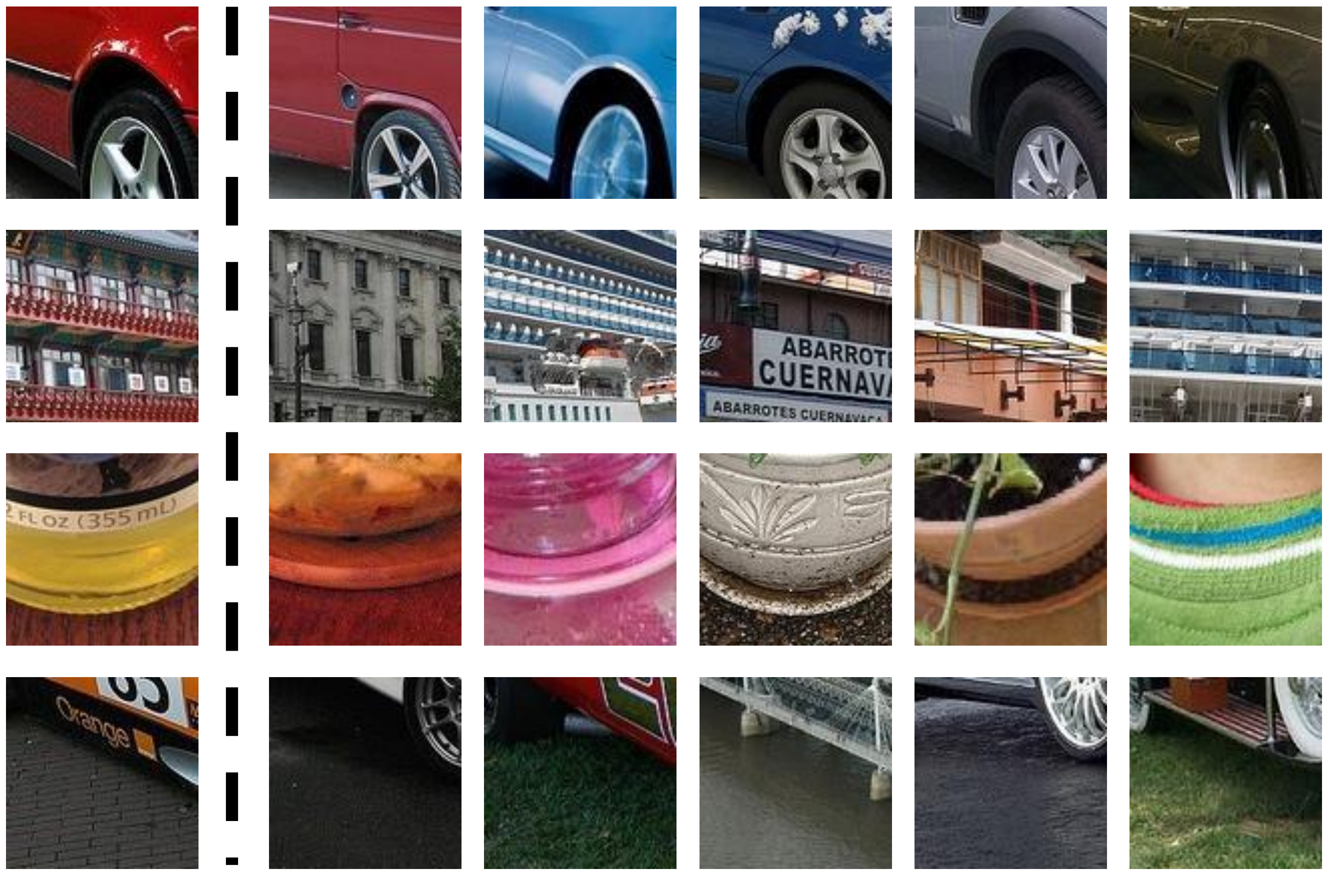
VOC 2007 Performance

(pretraining for R-CNN)

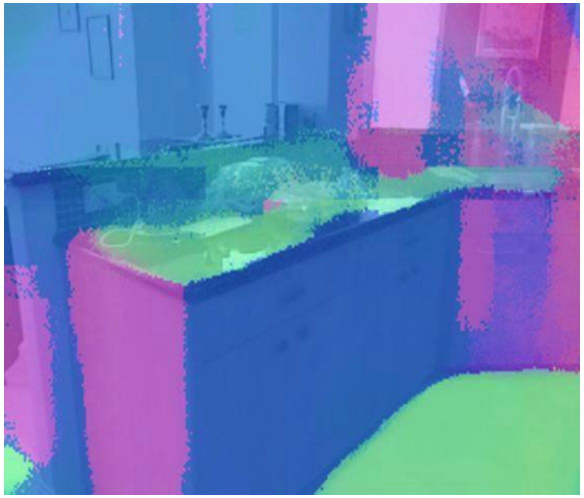


[Krähenbühl, Doersch, Donahue & Darrell, “Data-dependent Initializations of CNNs”, 2015]

Capturing Geometry?



Surface-normal Estimation

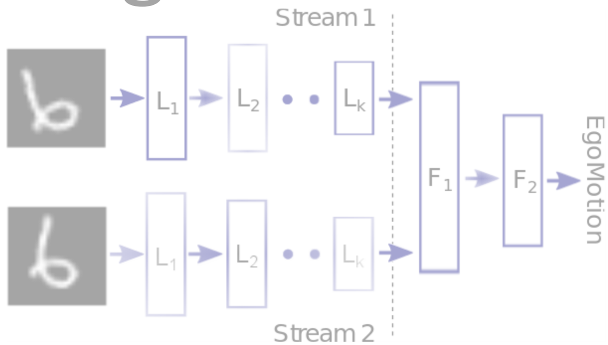


	Error (Lower Better)		% Good Pixels (Higher Better)		
Method	Mean	Median	11.25°	22.5°	30.0°
No Pretraining	38.6	26.5	33.1	46.8	52.5
Ours	**33.2**	21.3	36.0	51.2	57.8
ImageNet Labels	33.3	**20.8**	**36.7**	**51.7**	**58.1**

So, do we need semantic labels?

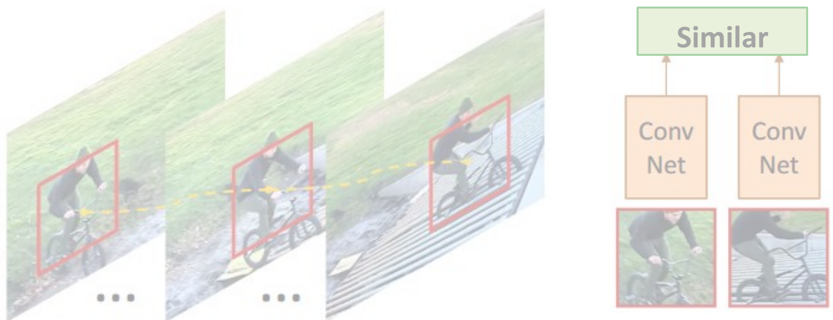
“Self-Supervision” and the Future

Ego-Motion



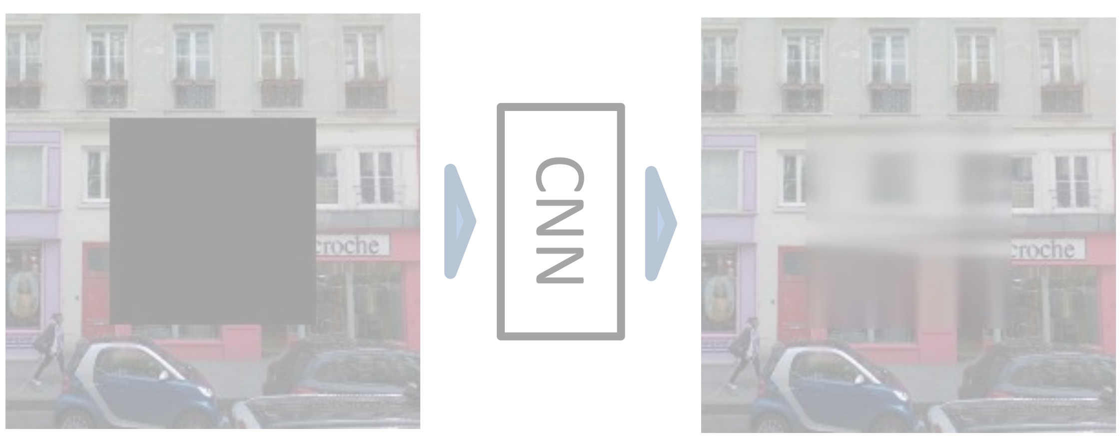
[Agrawal et al. 2015; Jayaraman et al. 2015]

Video



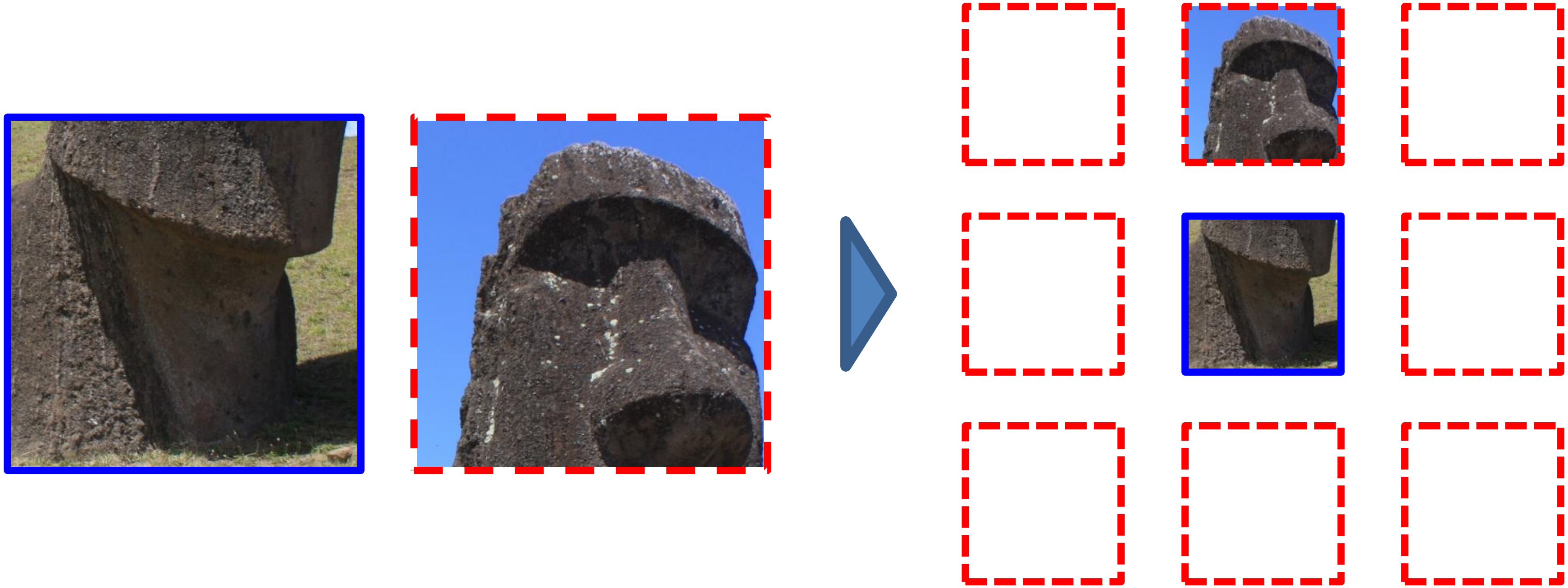
[Wang et al. 2015; Srivastava et al 2015; …]

Context

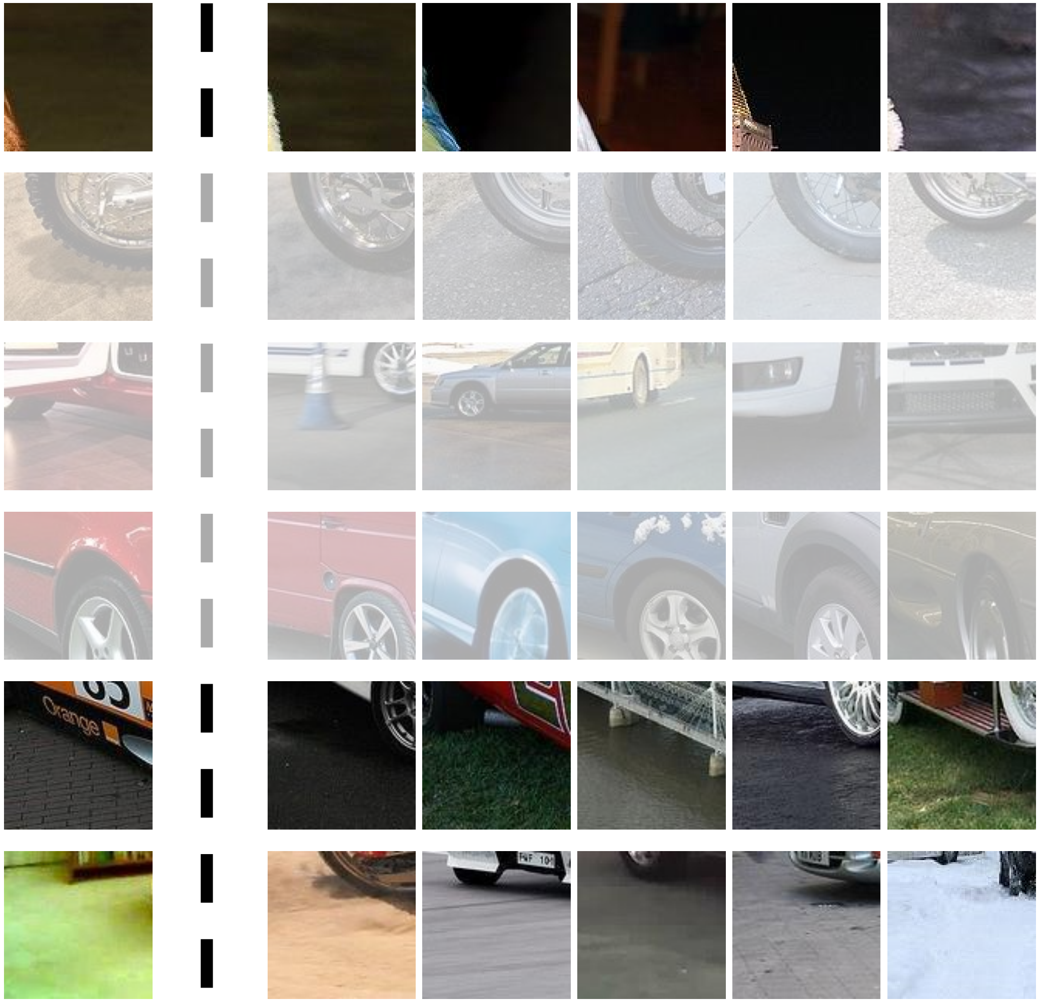


[Doersch et al. 2014; Pathak et al. 2015; Isola et al. 2015]

Thank you!

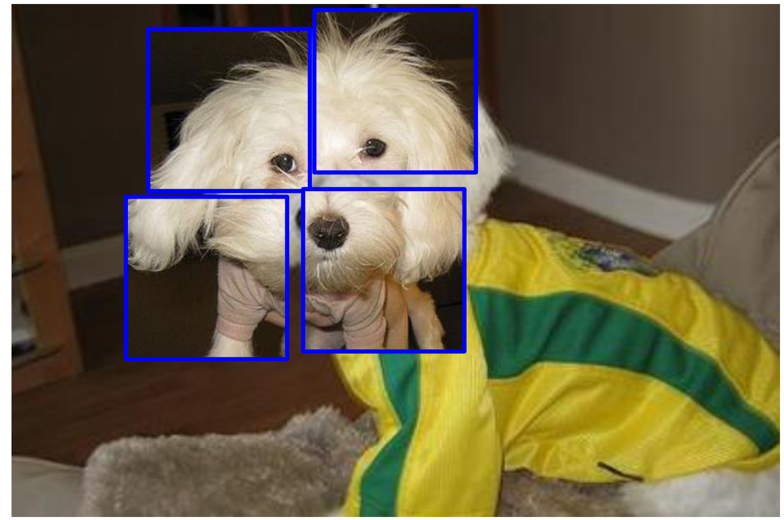


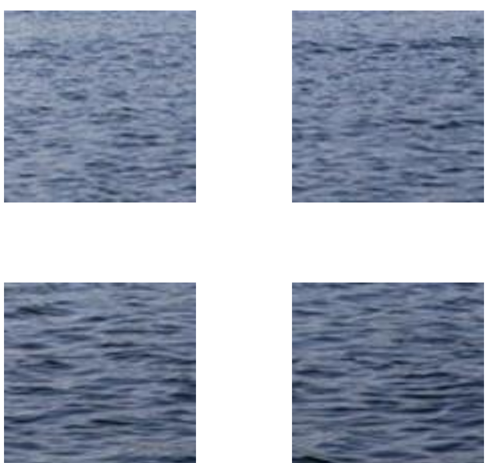
Visual Data Mining?

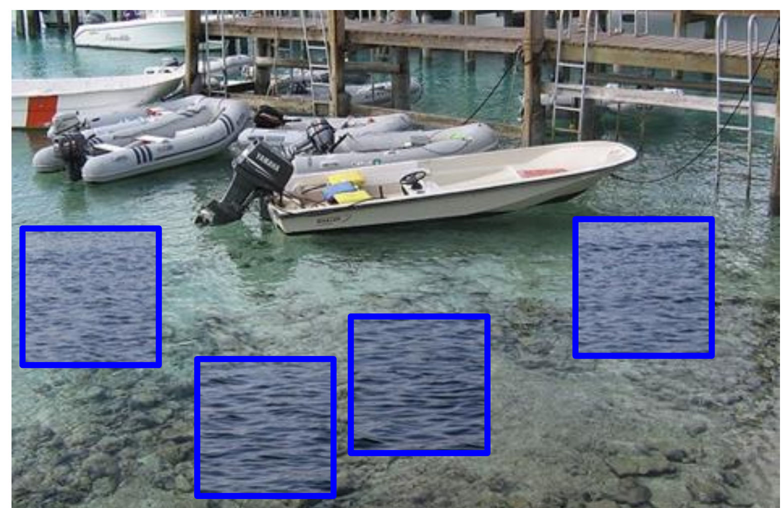


Geometric Verification

Like [Chum et al. 2007], but simpler







Geometric Verification

Like [Chum et al. 2007], but simpler

Learning Spatiotemporal Features with 3D Convolutional Networks



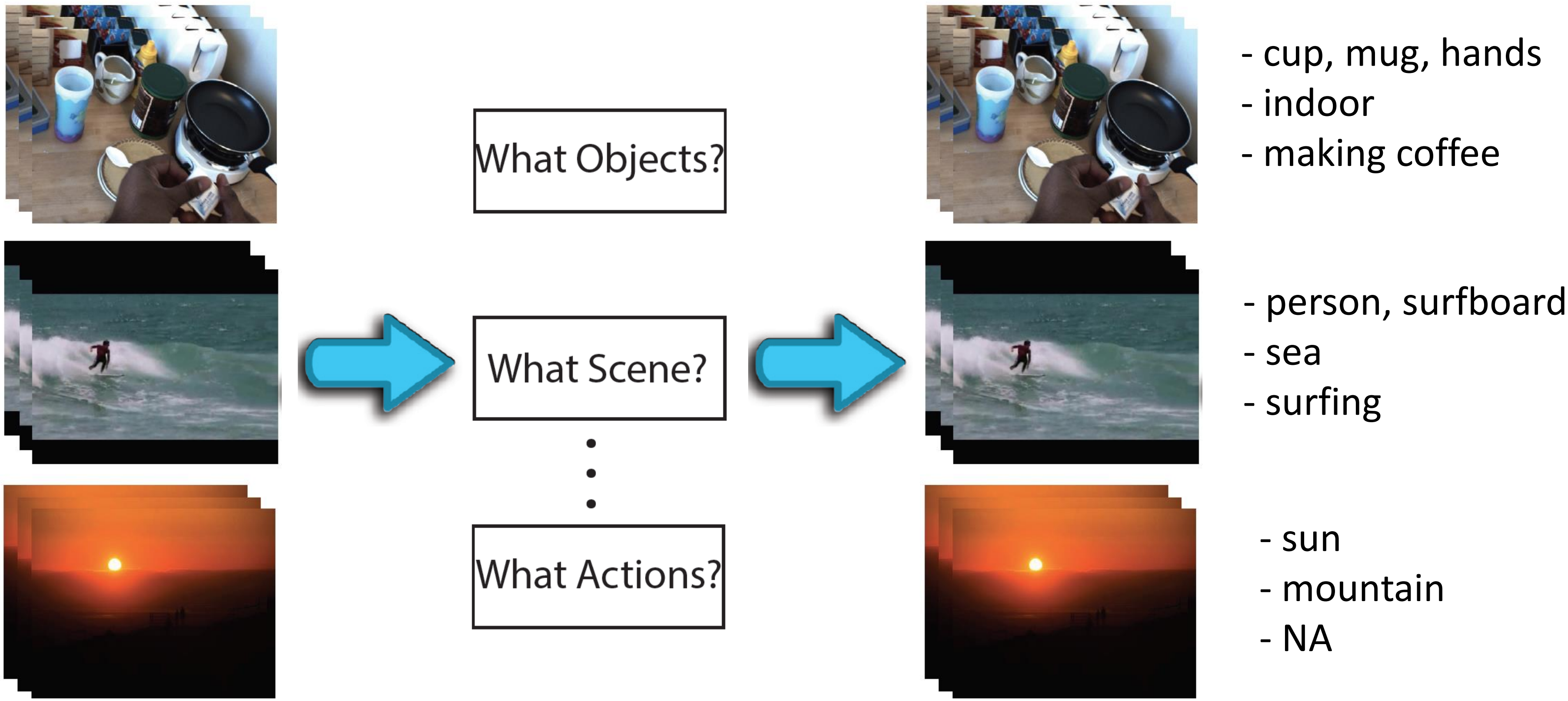


Du Tran (1,2)
Lubomir Bourdev (2)
Rob Fergus (2,3)
Lorenzo Torresani (1)
Manohar Paluri (2)

(1) Dartmouth College, (2) Facebook AI Research, (3) New York University

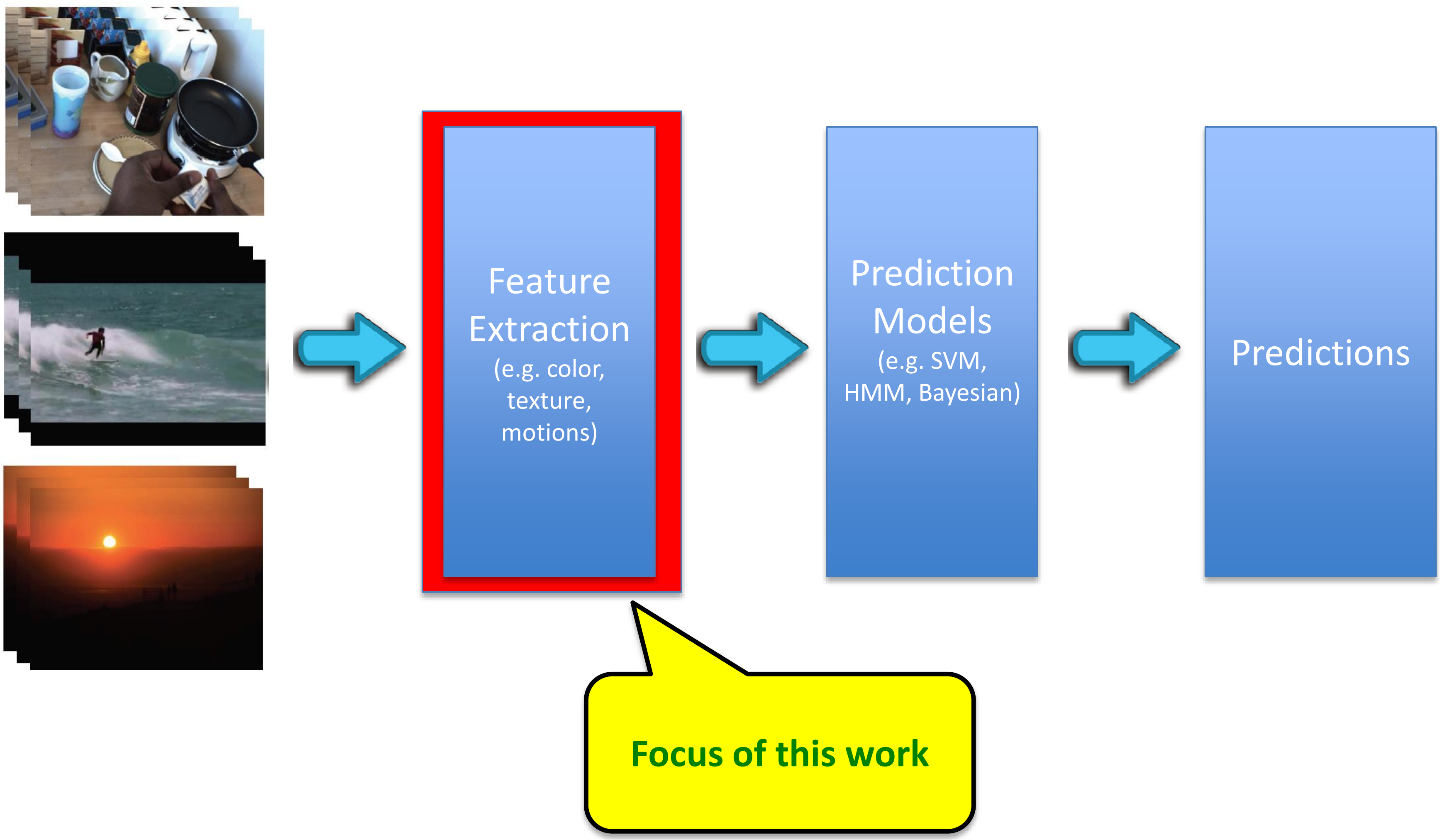
Introduction

Video Understanding problem:



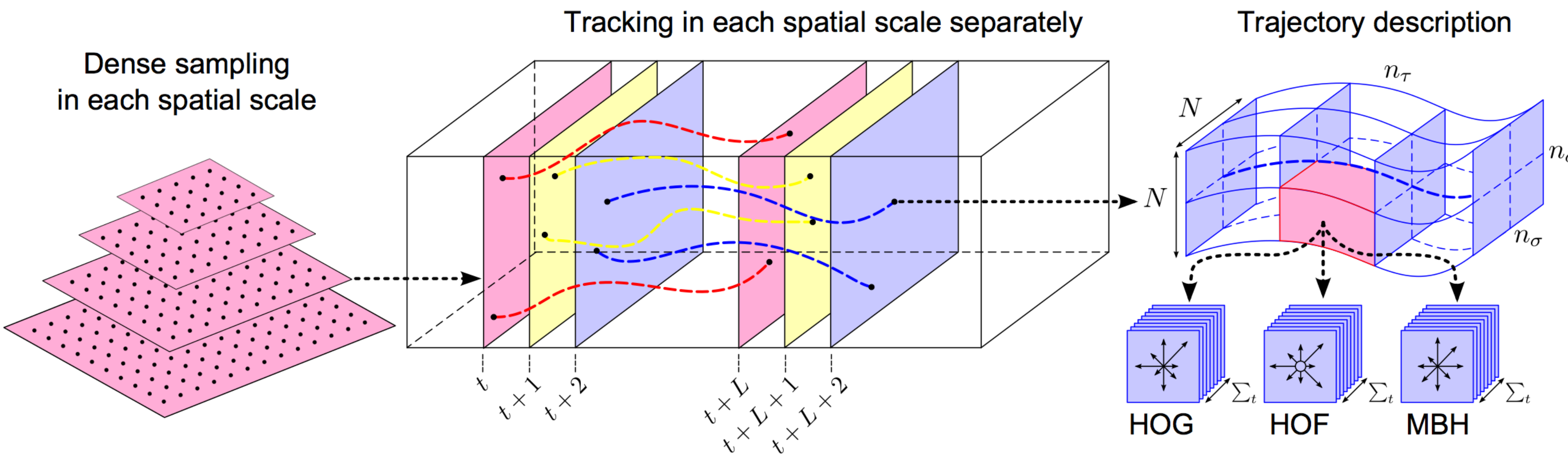
Many potential applications due to a large-growing number of internet videos

Traditional Computer Vision Pipeline



Current Best Video Features

- Improved Dense Trajectories (iDT)



Wang et al. IJCV'13

Pros:
- Don't need to learn
- Don't need large-scale training data

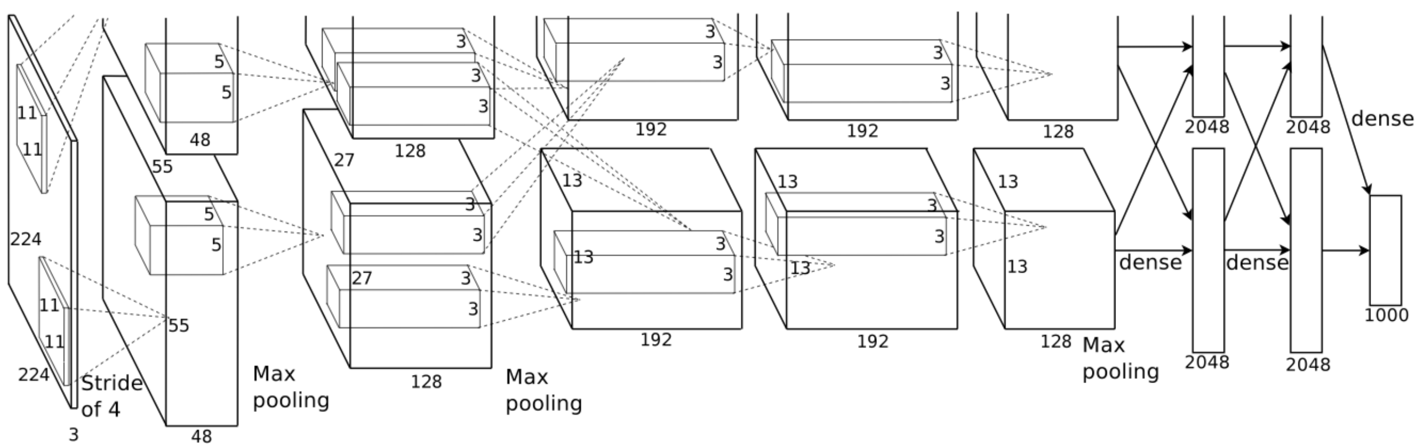
Cons:
- Highly hand-crafted
- Computational intensive
- Hard to parallelize

What If We Have Big Data?

- Learn features directly from data (no more human biases)
  - Normally built on deep learning, e.g. deep features
- Does it work?
  - Showed to work well for images [Donahue et al. ICML'14]
- How about videos?

Deep Image-based Features

IMAGENET Russakovsky et al. IJCV’15



Krizhevsky et al. NIPS’12

The marriage of Big Data and Good Deep Learning Models

- Fully-supervised trained on large-scale dataset.
- Activations are used as features for transferring tasks.

Can Image-based Features Applied to Videos?

- A video is a sequence of images?

Image feature

Image feature

Image feature

Video feature

Any problems?

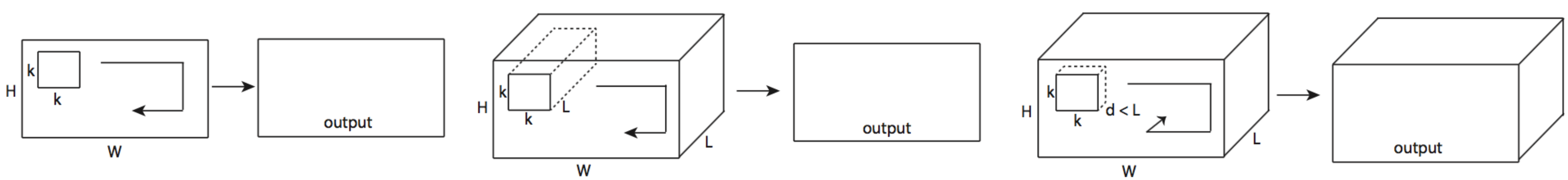
No explicit motion modeling

What is right for jointly modeling appearance & motion?

3D ConvNets instead of 2D ConvNets

2D ConvNet vs. 3D ConvNet

- Basic operations: 2D vs. 3D convolution
- Most of current work
  - Use 2D convolution on images or videos
  - Cannot model temporal information (motions)

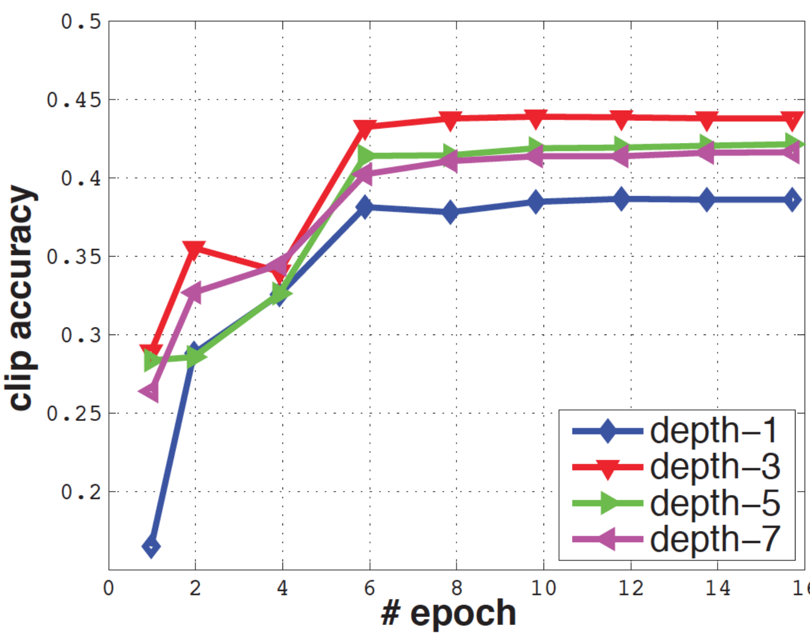


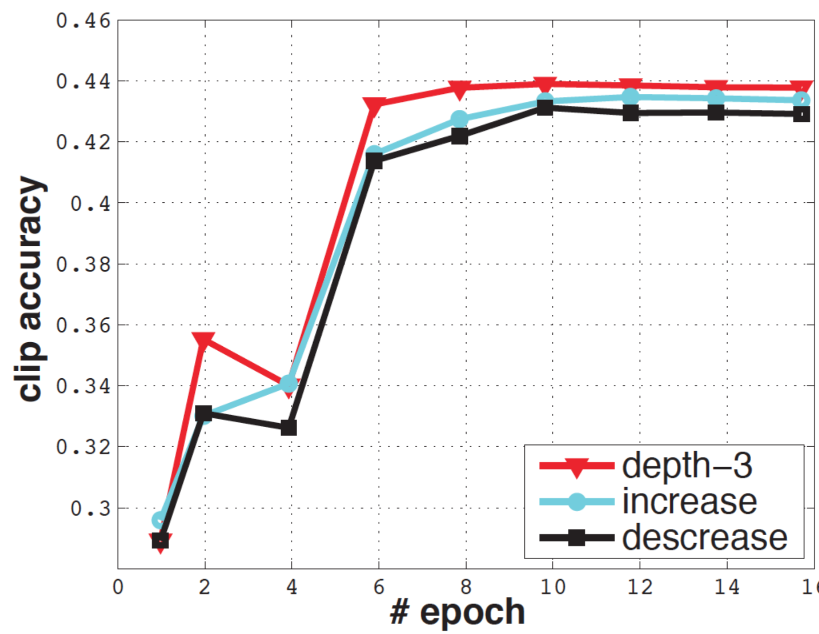
(a) 2D convolution (b) 2D convolution on multiple frames (c) 3D convolution

- We propose to use 3D ConvNets for video feature learning

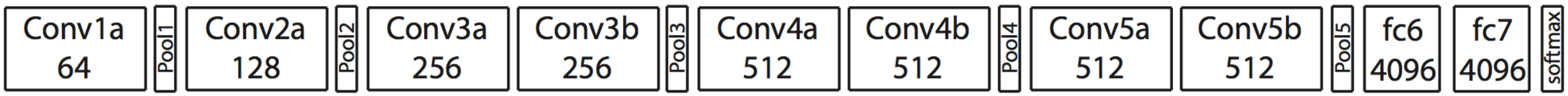
What is a Good Architecture for 3D ConvNets?

- Dataset: UCF101 (13K videos of 101 actions)
- Use similar architecture, varying kernel temporal depth





Learning Video Features with C3D

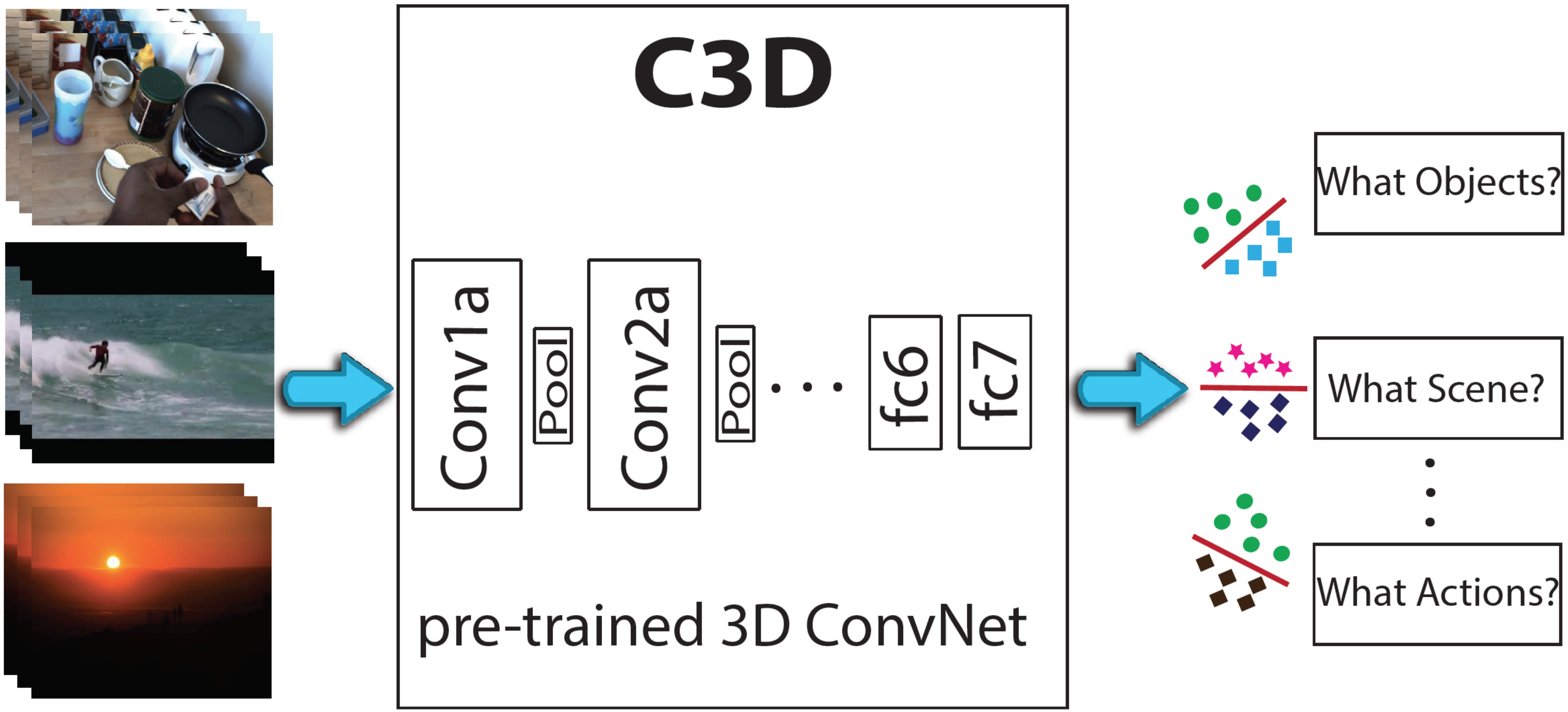


- C3D Architecture
  - 8 convolution, 5 pool, 2 fully-connected layers
  - 3x3x3 convolution kernels
  - 2x2x2 pooling kernels
- Dataset: Sports-1M [Karpathy et al. CVPR'14]
  - 1.1M videos of 487 different sport categories
  - Train/test splits are provided

Sport Classification Results

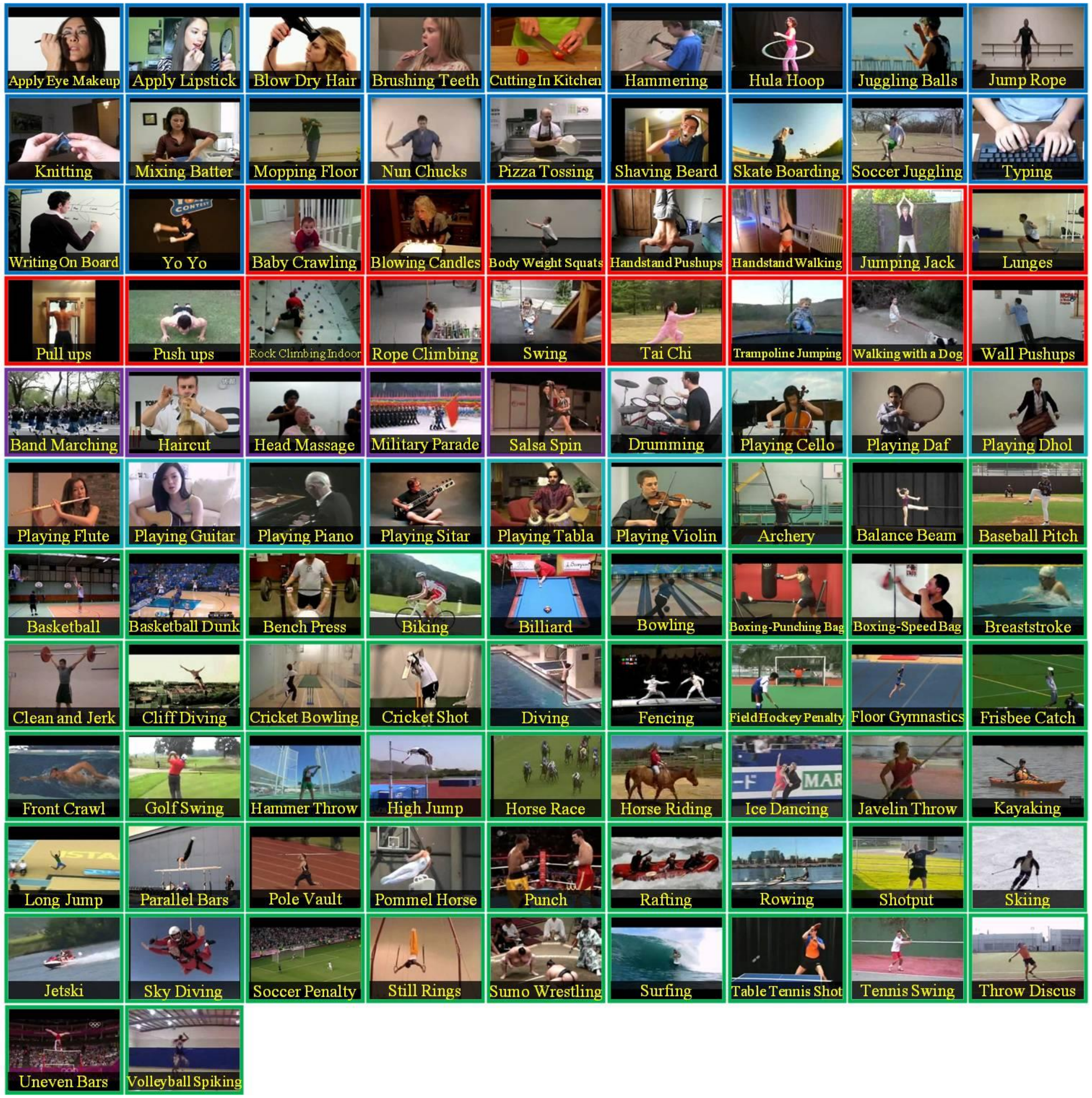
Method	Number of Nets	Clip hit@1	Video hit@1	Video hit@5
Deep Video's Single-Frame + Multires [19]	3 nets	42.4	60.0	78.5
Deep Video's Slow Fusion [19]	1 net	41.9	60.9	80.2
C3D (trained from scratch)	1 net	44.9	60.0	84.4
C3D (fine-tuned from I380K pre-trained model)	1 net	**46.1**	**61.1**	**85.2**

C3D as Generic Features



Simple recipe: C3D + linear SVM = good performance

Action Recognition

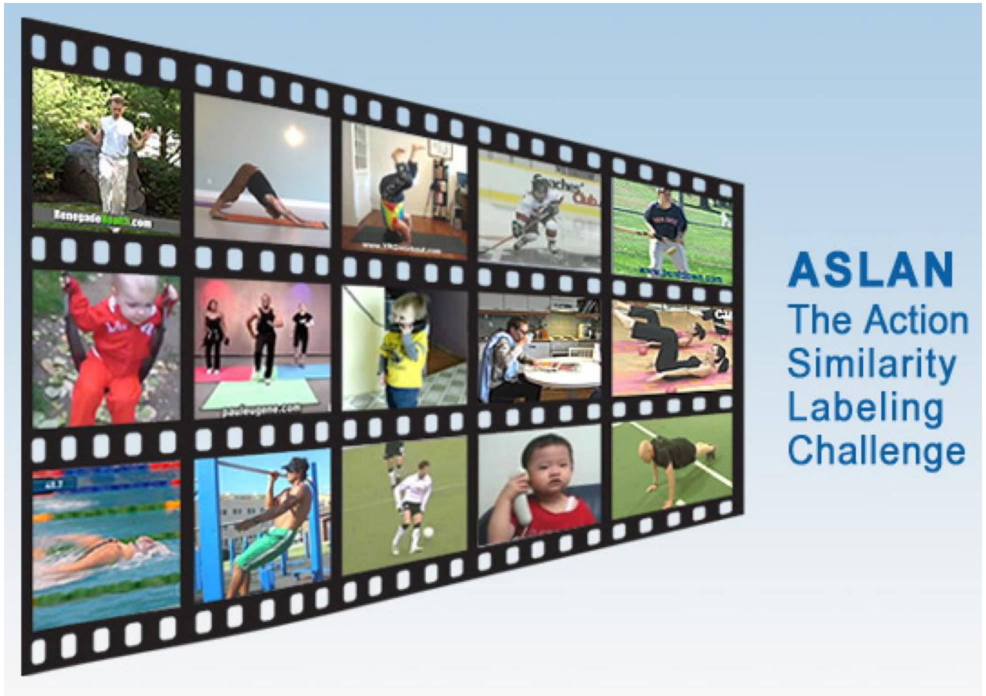


UCF101

Action Recognition Results

	Method	Accuracy (%)
Baselines	Imagenet	68.8
	iDT	76.2
Use raw pixel inputs	Deep networks [19]	65.4
	Spatial stream network [36]	72.6
	LRCN [7]	71.1
	LSTM composite model [39]	75.8
	C3D (1 net)	82.3
	C3D (3 nets)	**85.2**
Use optical flows	iDT with Fisher vector [31]	87.9
	Temporal stream network [36]	83.7
	Two-stream networks [36]	88.0
	LRCN [7]	82.9
	LSTM composite model [39]	84.3
	Multi-skip feature stacking [26]	89.1
	C3D (3 nets) + iDT	**90.4**

Action Similarity Labeling

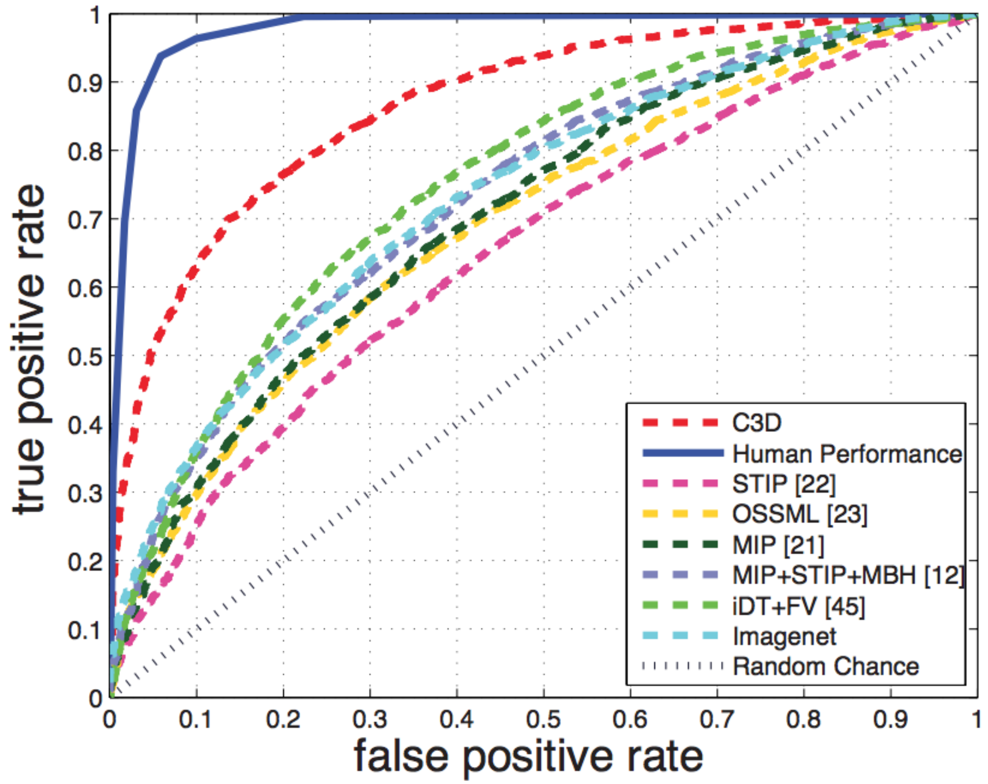


TASK: Given a pair of clips, predict same or different actions

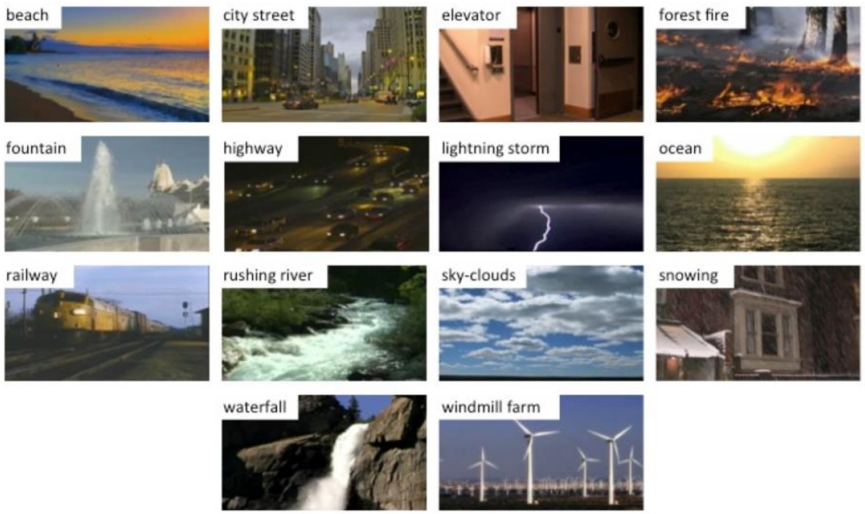
Very challenging evaluation setting: train and test on different categories of actions

ASLAN Results

Method	Features	Model	Acc.	AUC
[22]	STIP	linear	60.9	65.3
[23]	STIP	metric	64.3	69.1
[21]	MIP	metric	65.5	71.9
[12]	MIP+STIP+MBH	metric	66.1	73.2
[45]	iDT+FV	metric	68.7	75.4
Baseline	Imagenet	linear	67.5	73.8
Ours	**C3D**	linear	**78.3**	**86.5**



Dynamic Scene Classification

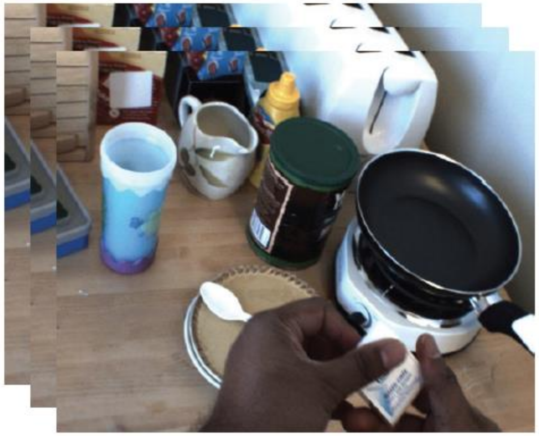


YUPENN

Maryland

Dataset	[5]	[41]	[9]	[10]	Imagenet	**C3D**
Maryland	43.1	74.6	67.7	77.7	**87.7**	**87.7**
YUPENN	80.7	85.0	86.0	96.2	96.7	**98.1**

Object Classification



Egocentric object dataset

Dataset Task	Object object recognition
Method Result	[31] 12.0
C3D Δ	**22.3** 10.3

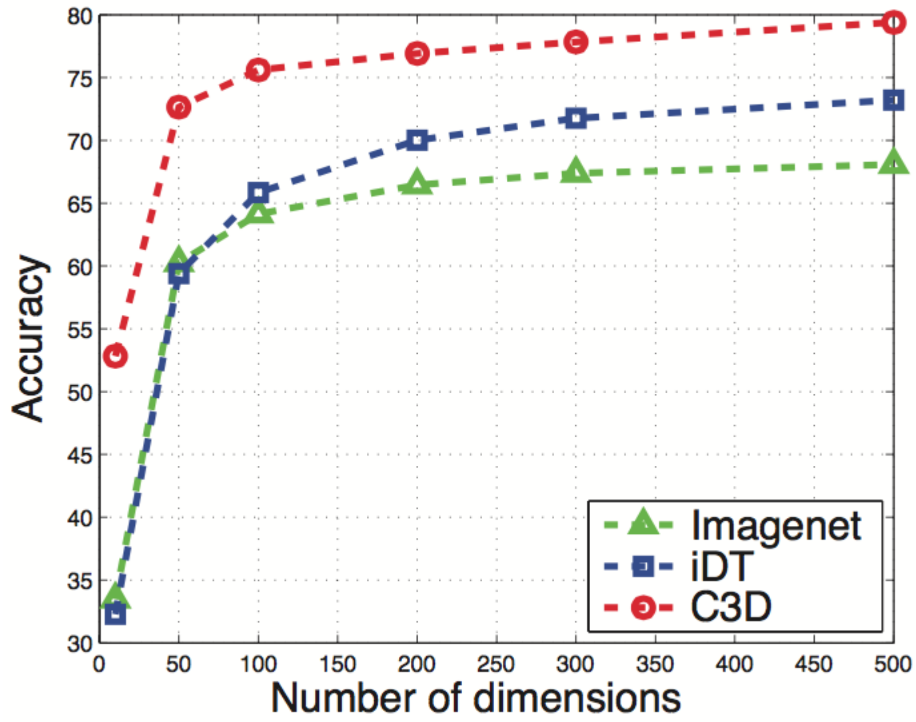
Result Summary

C3D performance compared with current methods

Dataset Task	Sport1M action recognition	UCF101 action recognition	ASLAN action similarity labeling	YUPENN scene classification	UMD scene classification	Object object recognition
Method Result	[19] 80.2	[39]([26]) 75.8 (89.1)	[31] 68.7	[10] 96.2	[10] 77.7	[32] 12.0
C3D Δ	**85.2** 5.0	**85.2 (90.4)** 9.4 (1.3)	**78.3** 9.6	**98.1** 1.9	**87.7** 10.0	**22.3** 10.3

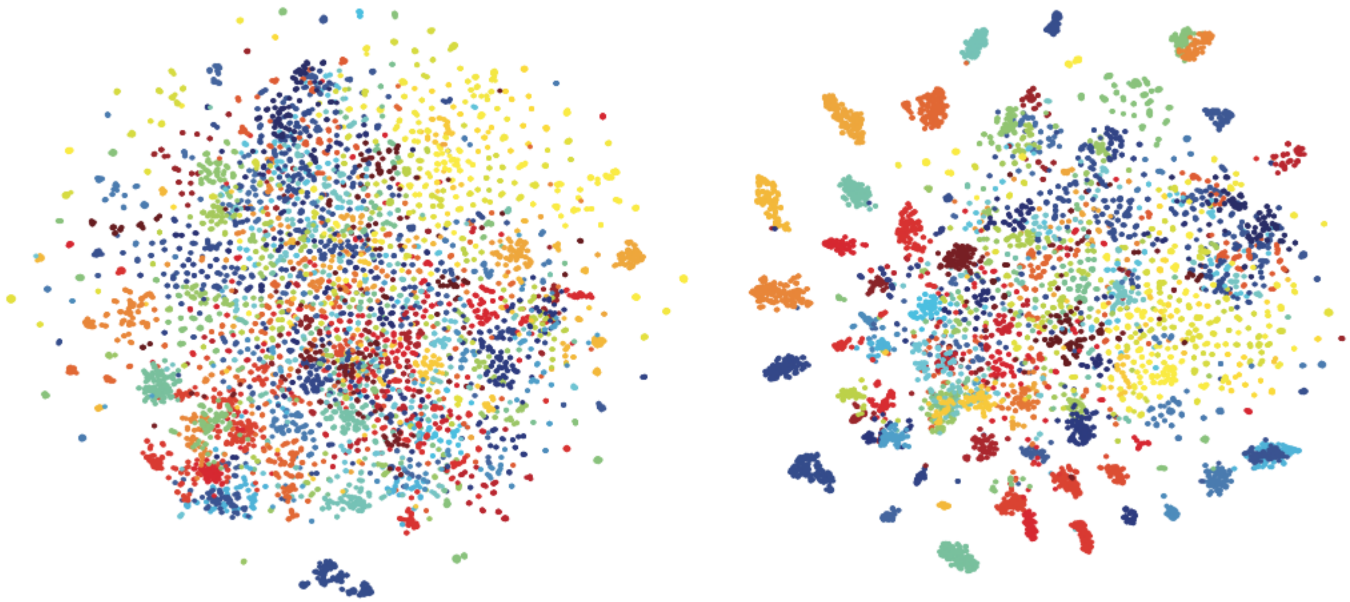
Consistently outperforms state-of-the-art methods on 4 different tasks and 6 different datasets

C3D is Compact



- 10-20% better than Imagenet and iDT at low dimension
- Obtains 52.8% using only 10-dim (random chance is less than 0.96%)

Qualitative Comparison



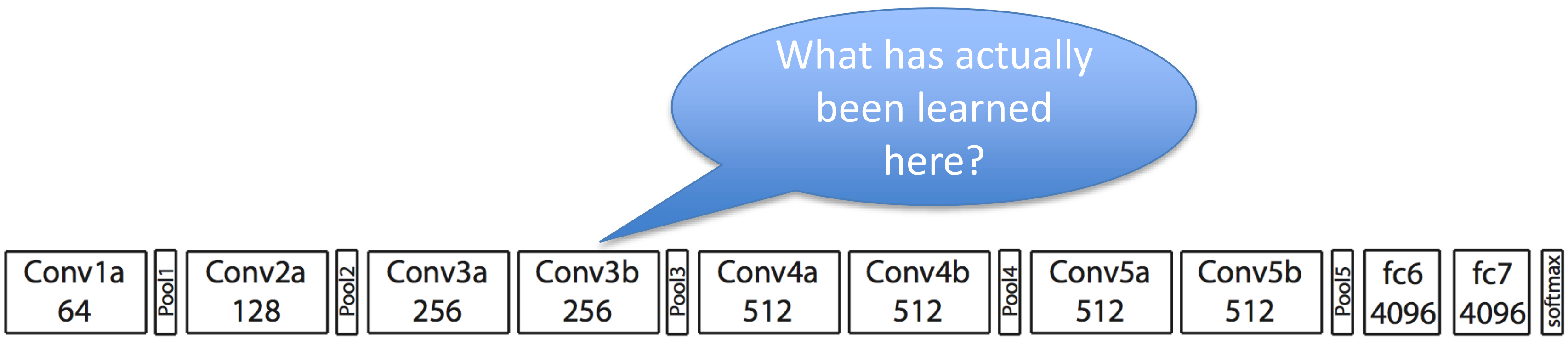
Imagenet C3D

C3D is Efficient

- Extract features on full UCF101
- 91x faster than iDT
- 276x faster than optical-flow-based methods

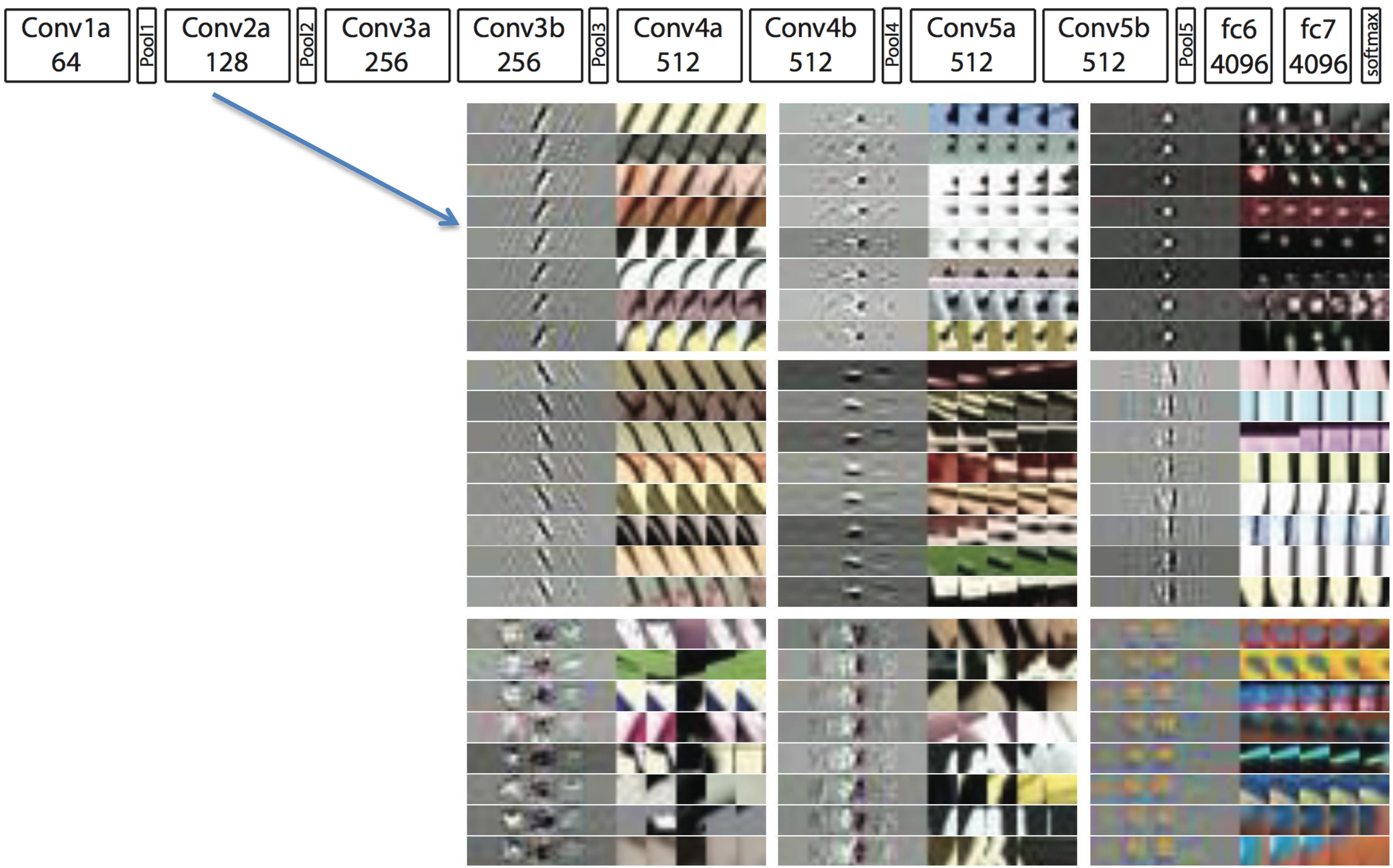
Why does C3D works so well?

- What does C3D learn at internal layers?



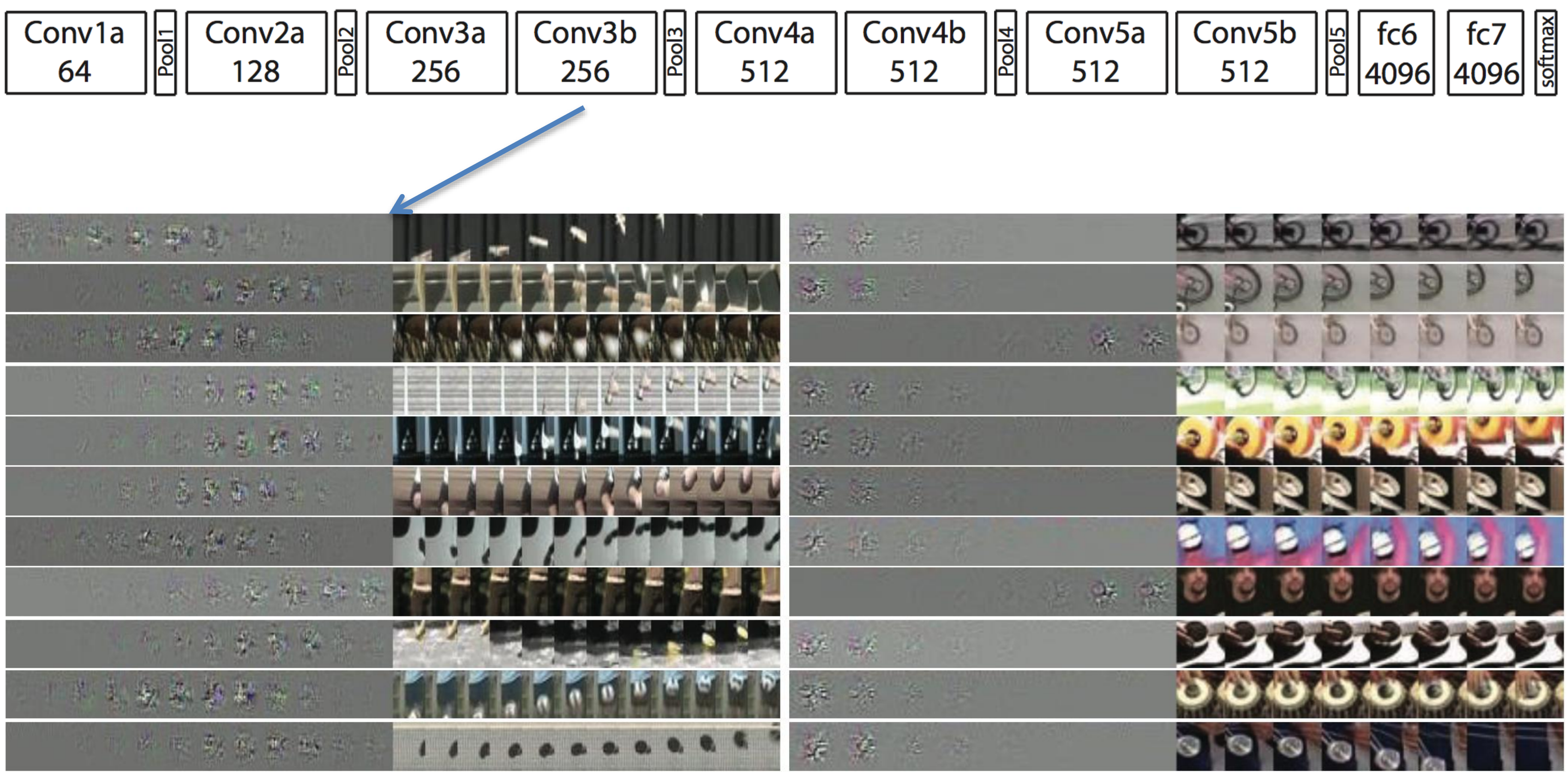
- Use Deconvolution method [Zeiler & Fergus ECCV'14] to visualize C3D learned features of some internal layers.

Deconvolutions of conv2a



Deconvolutions of conv3b

Deconvolutions of conv3b



Conclusions

- 3D ConvNet is well-suited for spatiotemporal feature learning.
- C3D is a good architecture for 3D ConvNet
- C3D is a good generic video features
  - Accurate
  - Compact
  - Efficient to compute
  - Easy to use

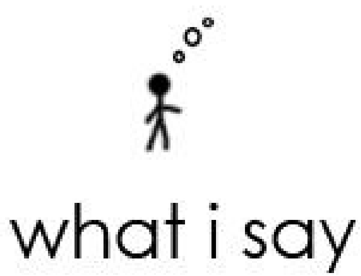
Source code & models are available at http://vlg.cs.dartmouth.edu/c3d

Thank you

- Q&A
- Demo

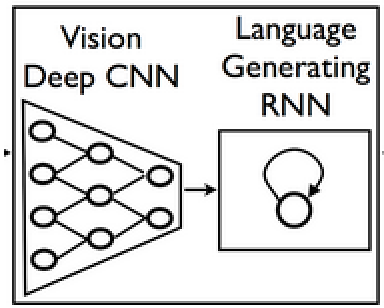
Words, Pictures, and Common Sense a.k.a. Learning by Playing

Devi Parikh
Virginia Tech

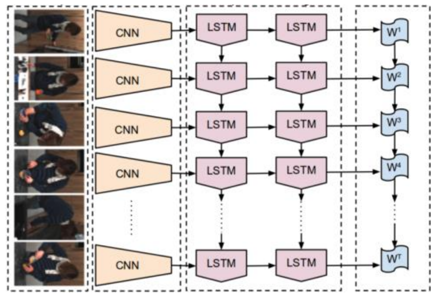


Slide credit: Devi Parikh

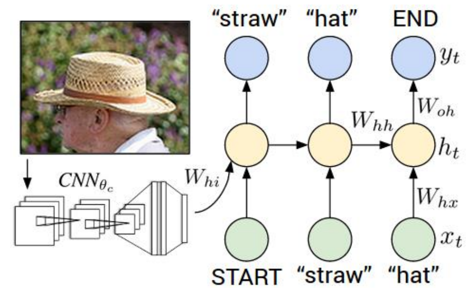
Image captioning is receiving a lot of attention



Vinyals et al., 2015

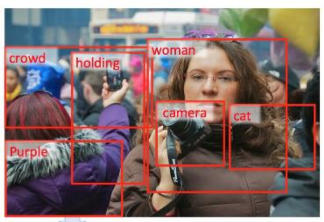


Donahue et al., 2015

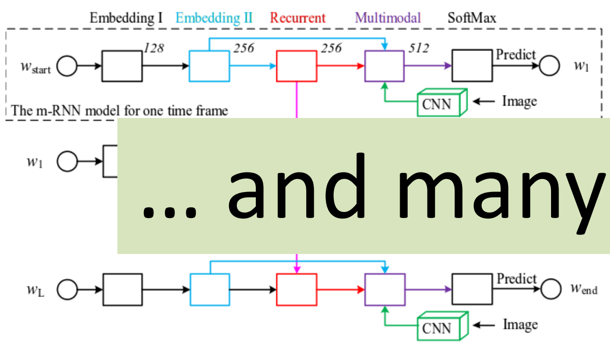


Karpathy and Fei-Fei, 2015

Hodosh et al., 2013



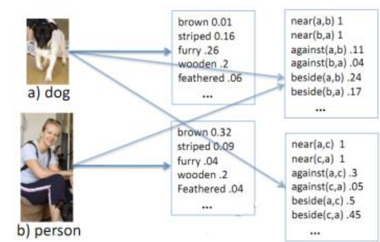
Fang et al., 2015



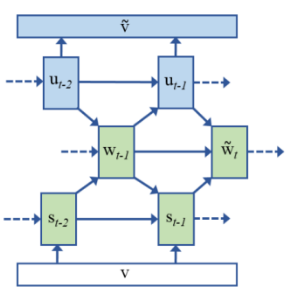
Mao et al., 2015

... and many more

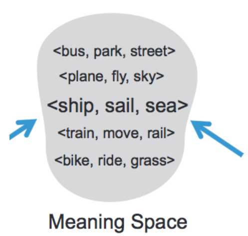
Ordonez et al., 2011



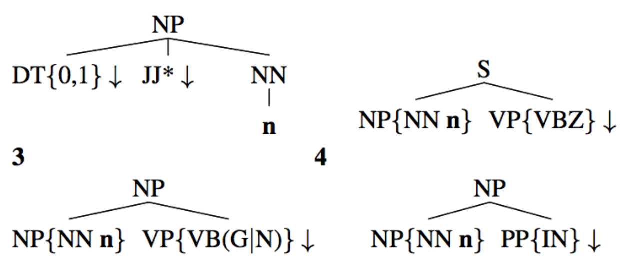
Kulkarni et al., 2011



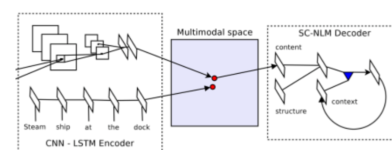
Chen and Zitnick, 2015



Farhadi et al., 2010



Mitchell et al., 2012



Kiros et al., 2015

"man in blue wetsuit is surfing on wave."

Karpathy and Fei-Fei, CVPR 2015

Slide credit: Devi Parikh

A group of young people playing a game of frisbee.

Vinyas et al., CVPR 2015

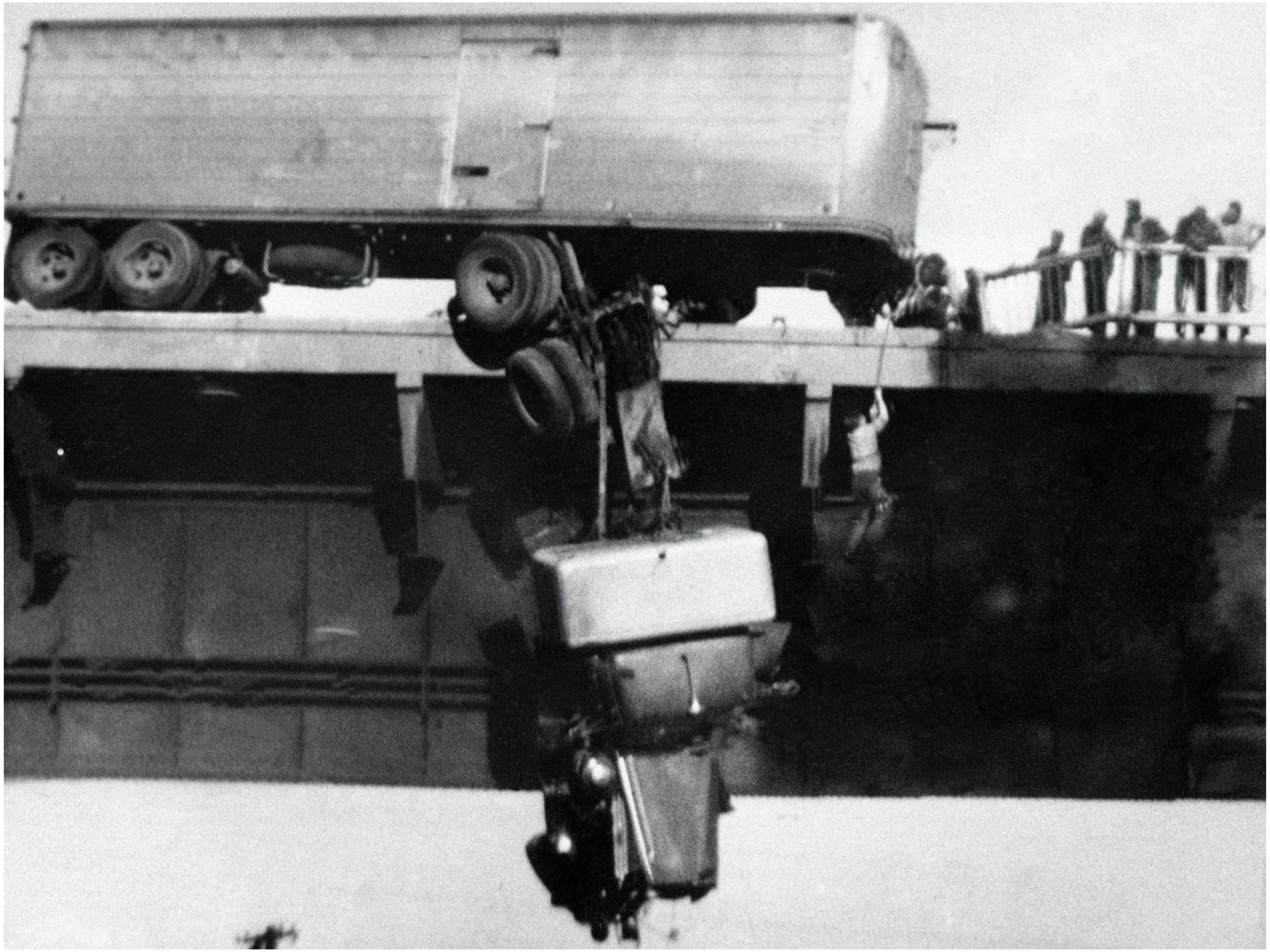
a car is parked
in the middle
of nowhere .

Kiros et al., TACL 2015

a pot of broccoli on a stove

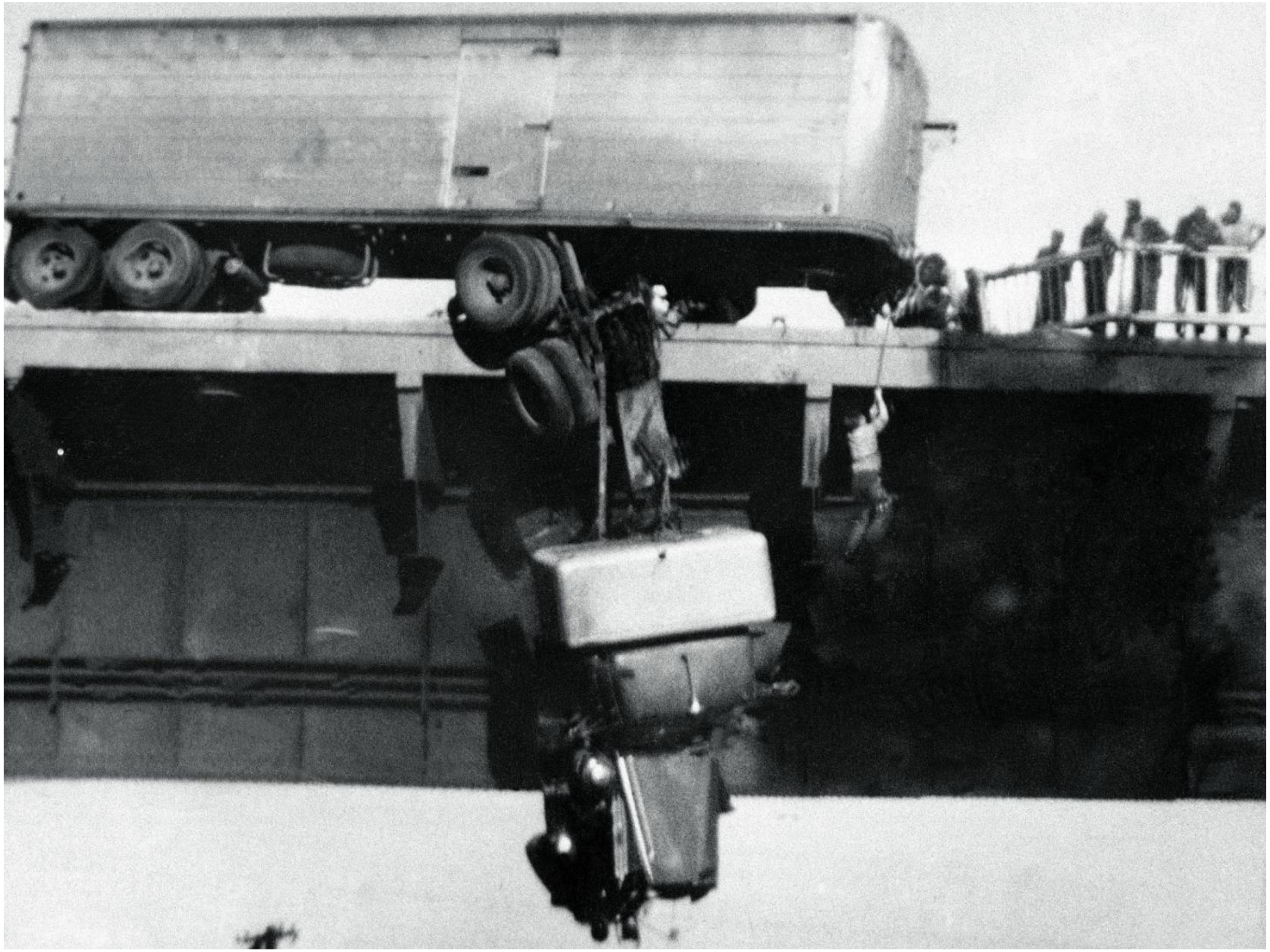
Fang et al. CVPR 2015

A man is rescued from his truck that is hanging dangerously from a bridge.



Slide credit: Larry Zitnick

A man is *rescued* from his truck that is hanging *dangerously* from a bridge.



Slide credit: Devi Parikh

Learning Common Sense

- Text
  - Reporting bias

Slide credit: Devi Parikh

Reporting bias in text

Word	*Teraword*	*Knext*
spoke	11,577,917	244,458
laughed	3,904,519	169,347
murdered	2,843,529	11,284
inhaled	984,613	4,412
breathed	725,034	34,912

Word	*Teraword*	*Knext*
hugged	610,040	10,378
blinked	390,692	20,624
was late	368,922	31,168
exhaled	168,985	3,490
was punctual	5,045	511

[Gordon et al. 2013]

Slide credit: Devi Parikh

Reporting bias in text

Word	*Teraword*	*Knext*	*Word*	*Teraword*	*Knext*
spoke	11,577,917	244,458	hugged	610,040	10,378
laughed	3,904,519	[illegible]	[illegible]	390,692	20,624
murdered	2,843,529	11,284	was late	368,922	31,168
inhaled	984,613	4,412	exhaled	168,985	3,490
breathed	725,034	34,912	was punctual	5,045	511

inhale:exhale = 6:1

[Gordon et al. 2013]

Slide credit: Devi Parikh

Reporting bias in text

murder:exhale = 17:1

Word	Teraword	Knext	Word	Teraword	Knext
spoke	11,577,917	244,458	hugged	610,040	10,378
laughed	3,904,519	169,347	blinked	390,692	20,624
murdered	2,843,529	11,284	was late	368,922	31,168
inhaled	984,613	4,412	exhaled	168,985	3,490
breathed	725,034	34,912	was punctual	5,045	511

[Gordon et al. 2013]

Slide credit: Devi Parikh

Reporting bias in text

Body Part	*Teraword*	*Knext*
Head	18,907,427	1,332,154
Eye(s)	18,455,030	1,090,640
Arm(s)	6,345,039	458,018
Ear(s)	3,543,711	230,367
Brain	3,277,326	260,863

Body Part	*Teraword*	*Knext*
Liver	246,937	10,474
Kidney(s)	183,973	5,014
Spleen	47,216	1,414
Pancreas	24,230	1,140
Gallbladder	17,419	1,556

[Gordon et al. 2013]

Slide credit: Devi Parikh

Reporting bias in text

Body Part	*Teraword*	*Knext*
Head	18,907,427	1,332,154
Eye(s)	18,455,030	1,090,640
Arm(s)	[illegible]	[illegible]
Ear(s)	3,543,711	230,367
Brain	3,277,326	260,863

Body Part	*Teraword*	*Knext*
Liver	246,937	10,474
Kidney(s)	183,973	5,014
[illegible]	[illegible]	[illegible]
Pancreas	24,230	1,140
Gallbladder	17,419	1,556

People have heads:gallbladders = 1085:1

[Gordon et al. 2013]

Slide credit: Devi Parikh

Do birds fly?

bird

/bɜːd/

noun

1. any warm-blooded egg-laying vertebrate of the class *Aves*, characterized by a body covering of feathers and forelimbs modified as wings. Birds vary in size between the ostrich and the humming bird *related adjectives* avian ornithic

Slide credit: Devi Parikh

Do birds fly?

penguin

[**peng**-gwin, **pen**-]

Spell Syllables

Examples Word Origin

noun, *Ornithology*

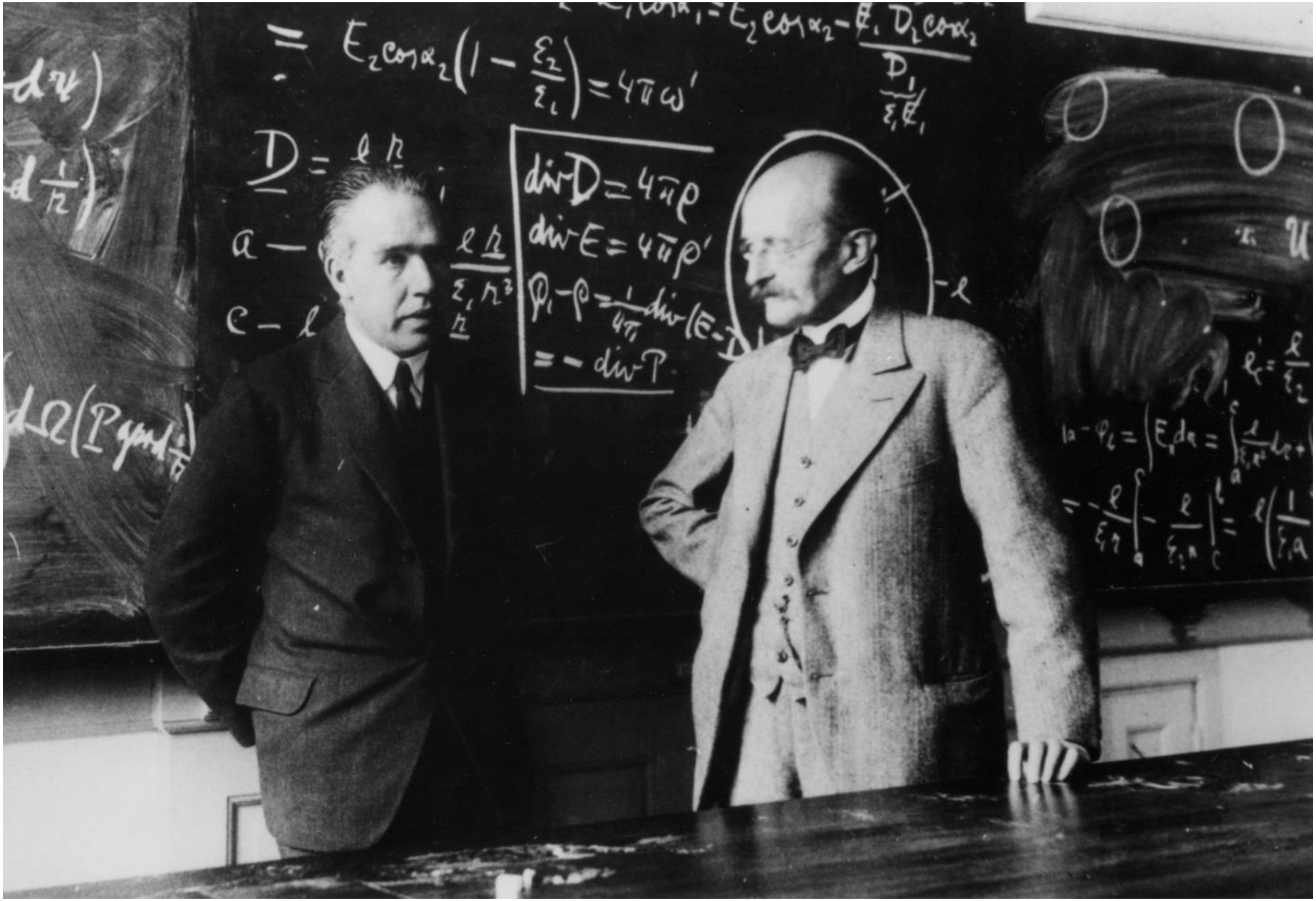
1. any of several flightless, aquatic birds of the family Spheniscidae, of the Southern Hemisphere, having webbed feet and wings reduced to flippers.

Slide credit: Devi Parikh

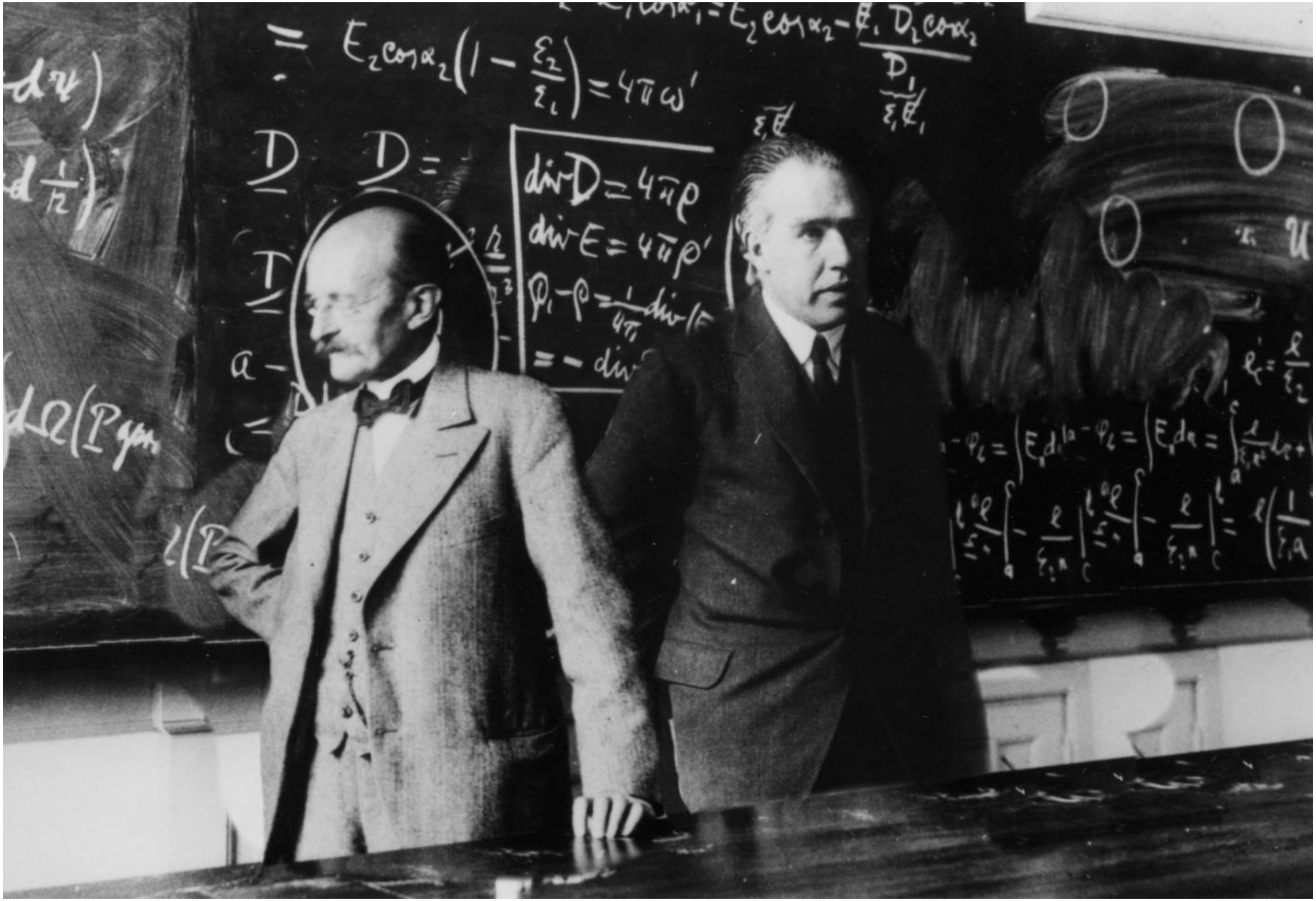
Learning Common Sense

- Text
  - Reporting bias
- From structure in our visual world?

Two professors converse in front of a blackboard.

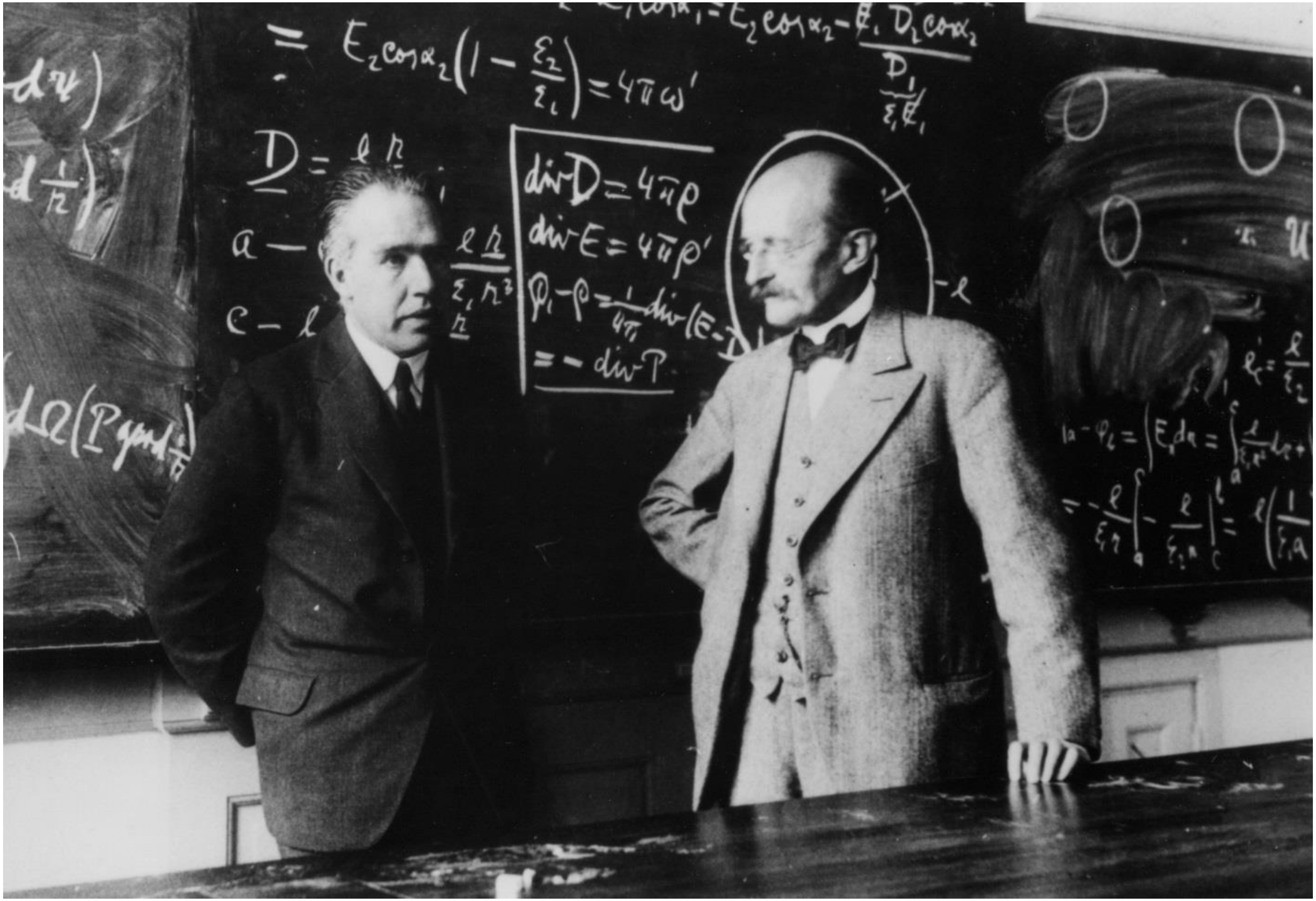


Two professors stand in front of a blackboard.



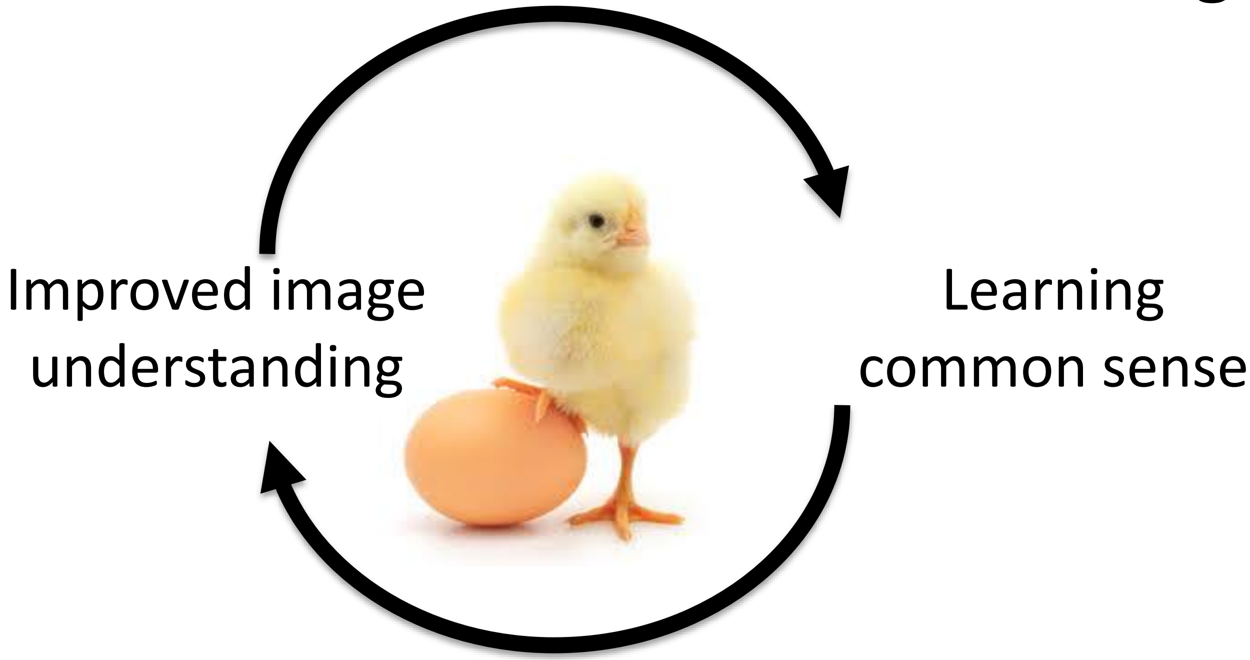
Slide credit: Larry Zitnick

Two professors converse in front of a blackboard.

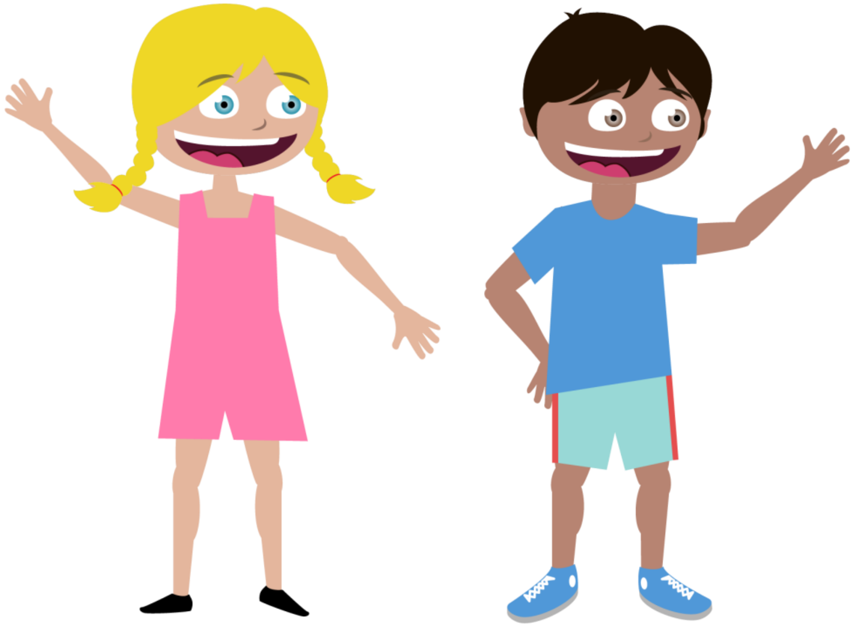


Challenges

- Lacking visual density
- Annotations are expensive
- Computer vision doesn't work well enough



Is photorealism necessary?

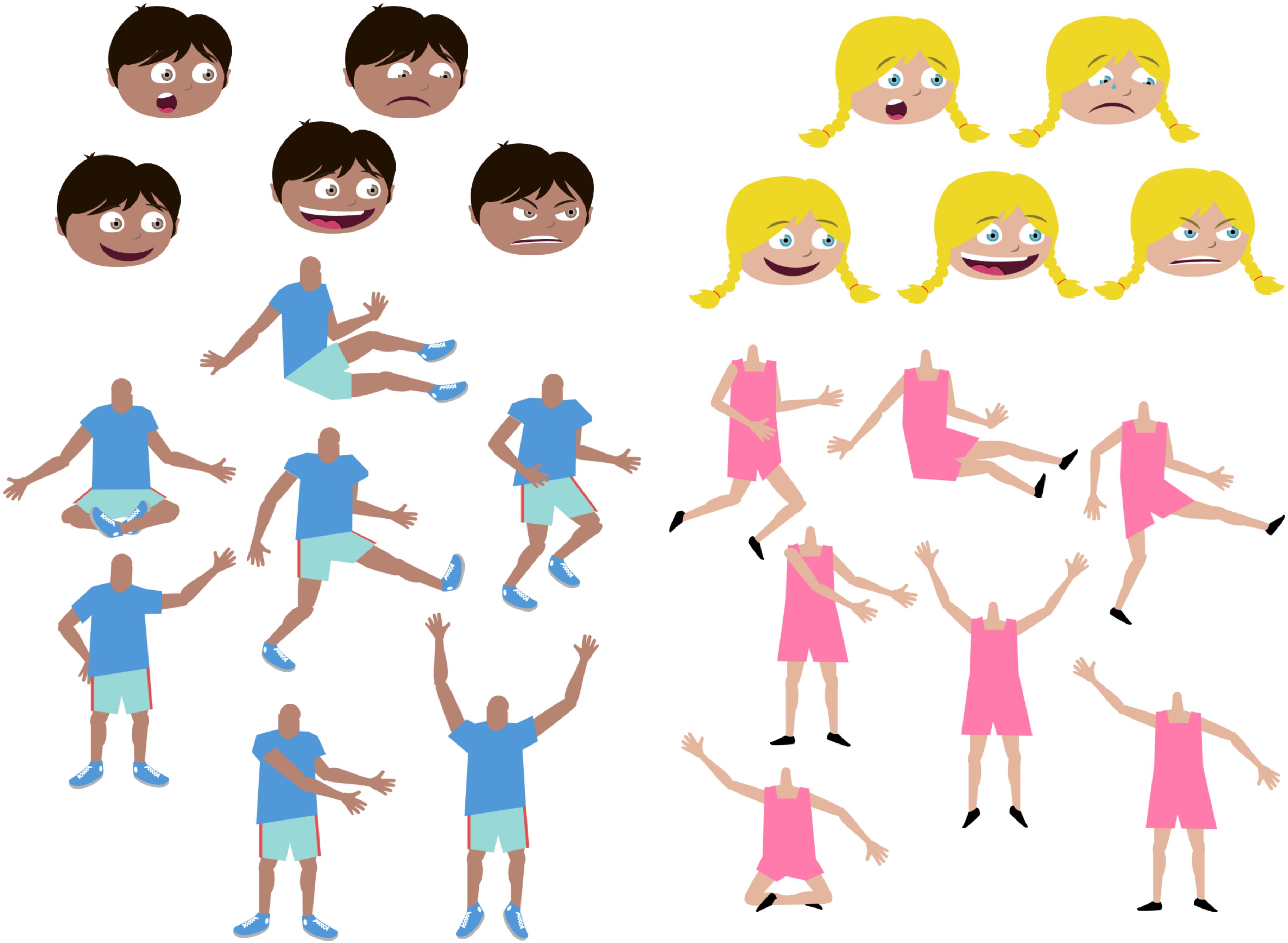


Jenny

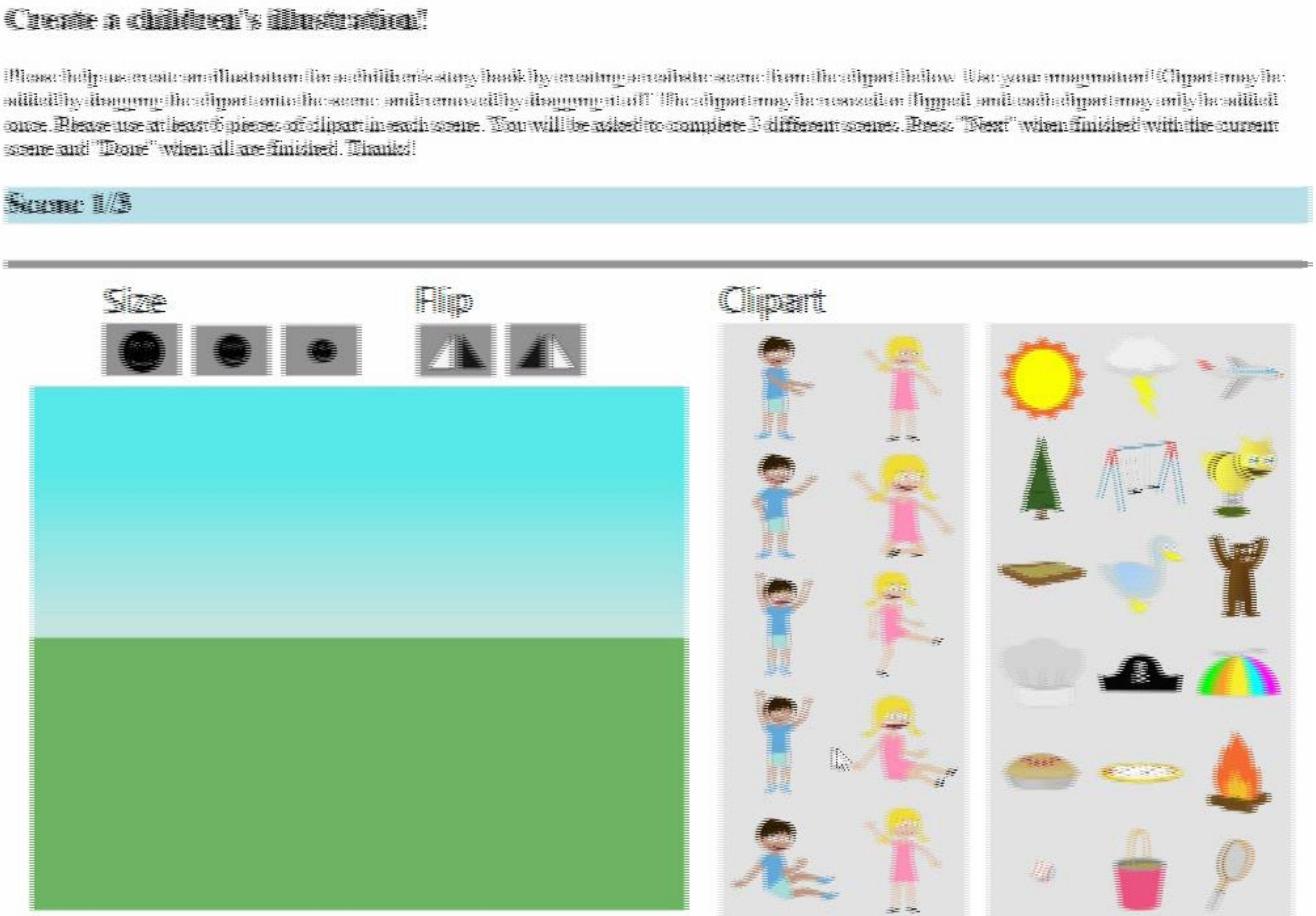
Mike

Slide credit: Larry Zitnick

Slide credit: Larry Zitnick



Slide credit: Larry Zitnick



2x

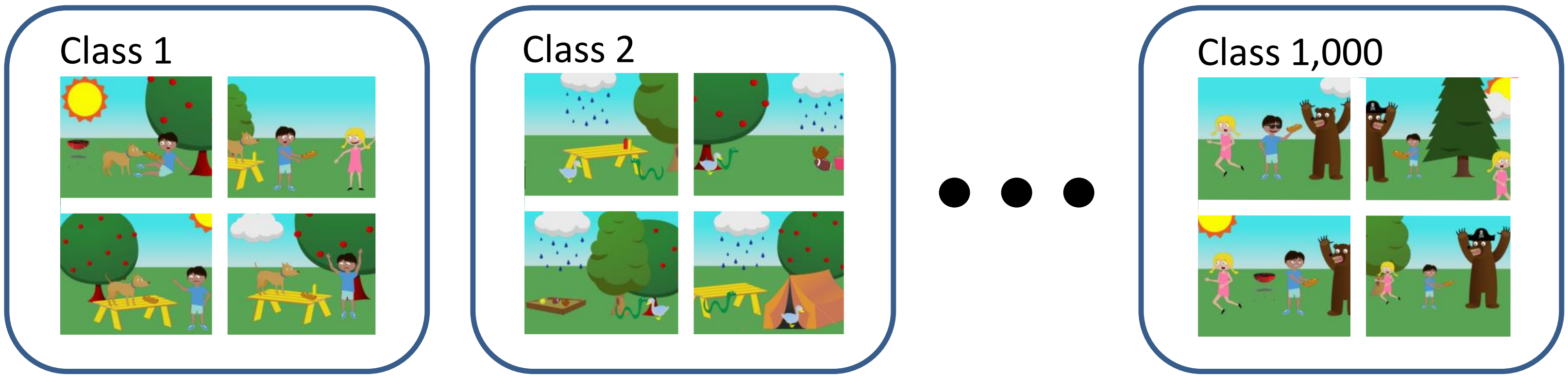
Slide credit: Devi Parikh

Mike fights off a bear by giving him a hotdog while Jenny runs away.

Slide credit: Larry Zitnick

Dataset

1,000 classes of semantically similar scenes:



1,000 classes x 10 scenes per class = 10,000 scenes

[Zitnick and Parikh, CVPR 2013, Oral]

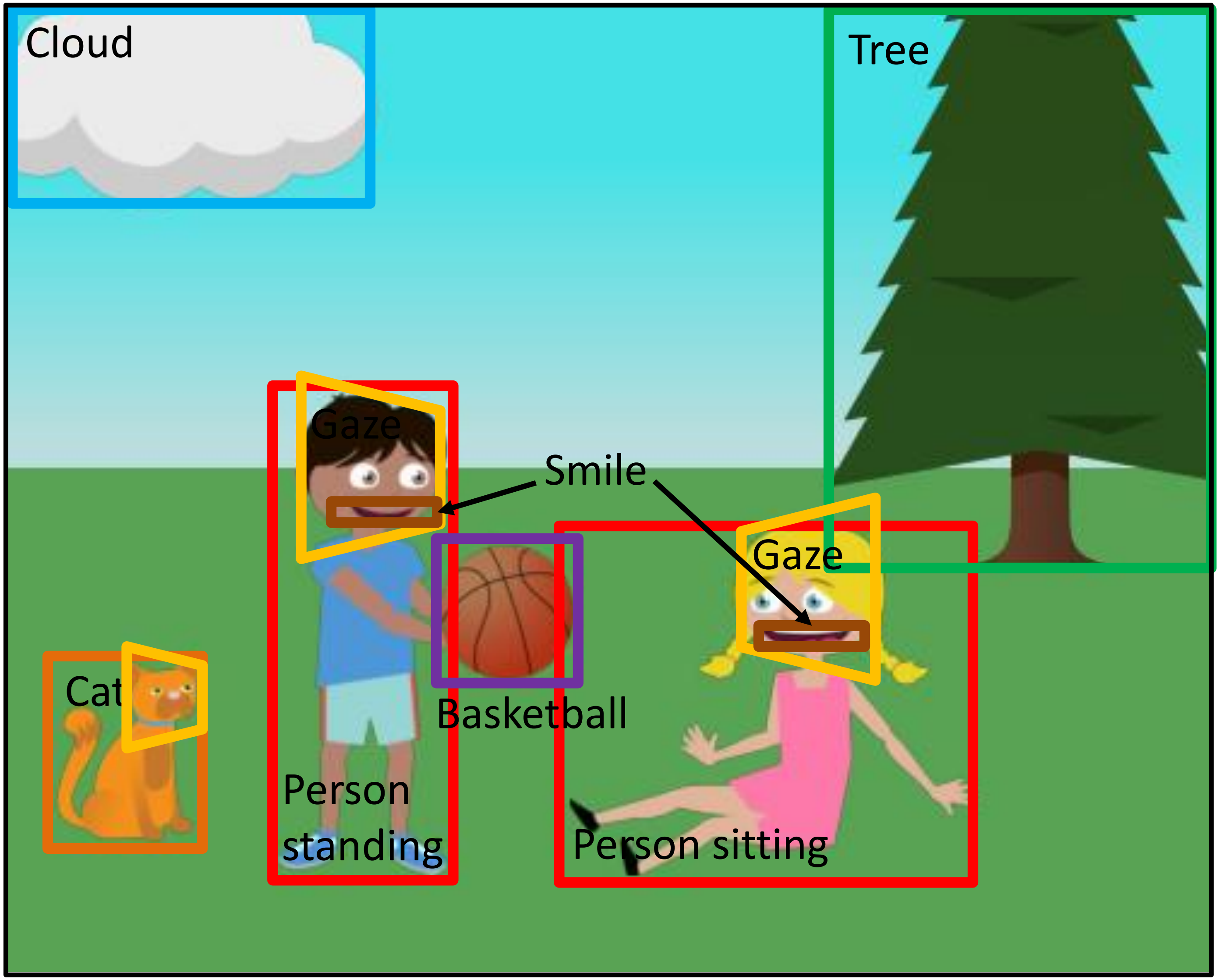
Dataset online

Slide credit: Larry Zitnick

Visual Features

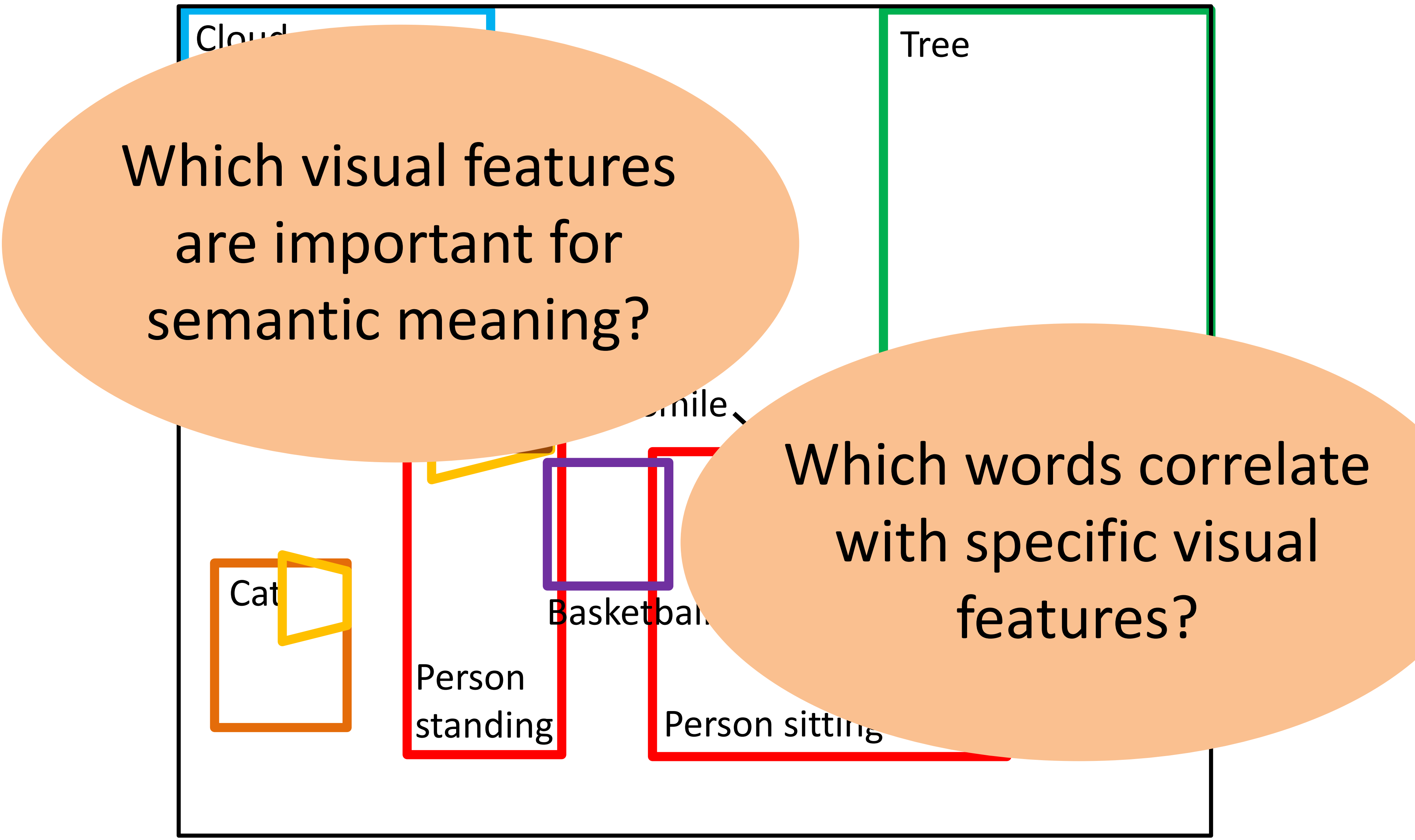
Slide credit: Larry Zitnick

Visual Features



Slide credit: Larry Zitnick

Visual Features



Generate Scenes

Input: Jenny is catching the ball. Mike is kicking the ball. The table is next to the tree.

Tuples: <<Jenny>,<catch>,<ball>> <<Mike>,<kick>,<ball>> <<table>,<be>,<>>

Automatically Generated

Human Generated

[Zitnick, Parikh and Vanderwende, ICCV 2013]

Generate Scenes

Jenny was mad and tried to kick Mike.
<<Jenny>, <be mad>, <>>
<<Jenny>, <try>, <kick>>
<<Jenny>, <kick>, <Mike>>

Mike and Jenny are wearing hats.
<<Mike>, <wear>, <hat>>
<< Jenny>, <wear>, <hat>>

Mike is holding a baseball bat.
<<Mike>, <hold>, <bat>>

Mike is standing in front of the table.
<<Mike>, <stand in_front_of>, <table>>

Jenny is sad because she wants the ball.
<<Jenny>, <be sad>, <>>
<<she>, <want>, <ball>>

Mike is wearing a blue hat with a star.
<<Mike>, <wear>, <hat>>
<<hat>, <with>, <star>>

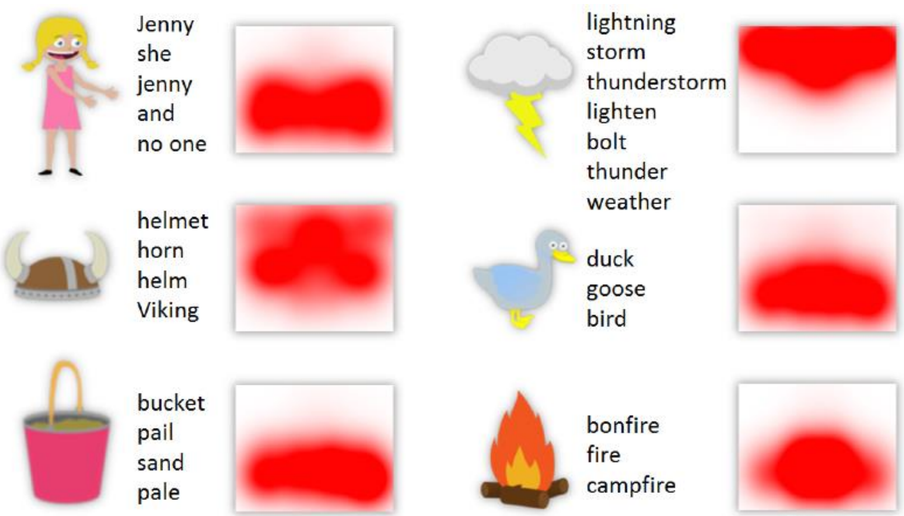
Jenny is happy to see Mike.
<<Jenny>, <be happy>, <>>
<<Jenny>, <see>, <Mike>>

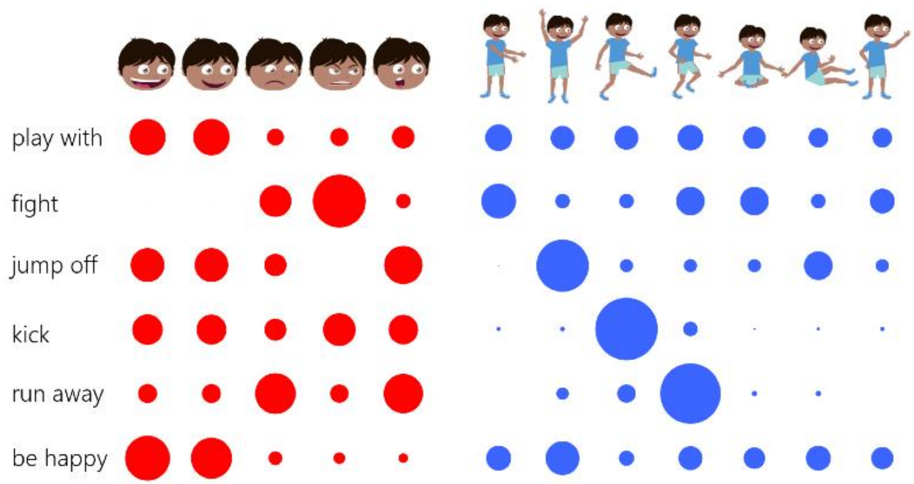
Mike is eating a burger
<<Mike>, <eat>, <burger>>

The cat is watching Jenny and Mike.
<<cat>, <watch>, <Jenny>>
<<cat>, <watch>, <Mike>>

Jenny wants Mike to share the bat.
<<Jenny>, <want>, <share>>
<<Mike>, <share>, <bat>>

Jenny is on the swings.
<<Jenny>, <be>, <>>

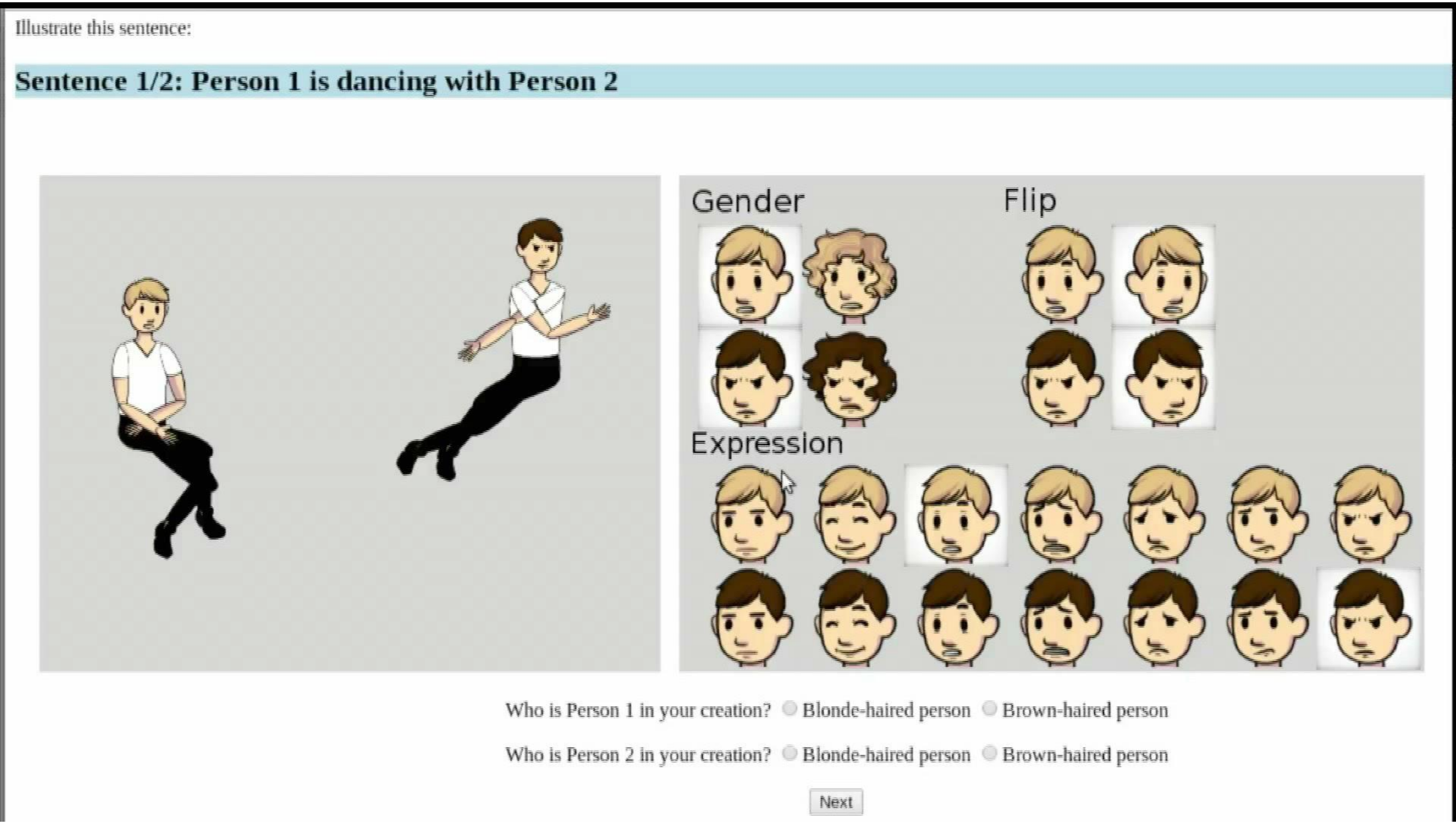




[Zitnick, Parikh and Vanderwende, ICCV 2013]

Slide credit: Devi Parikh

Learning Fine-grained Interactions

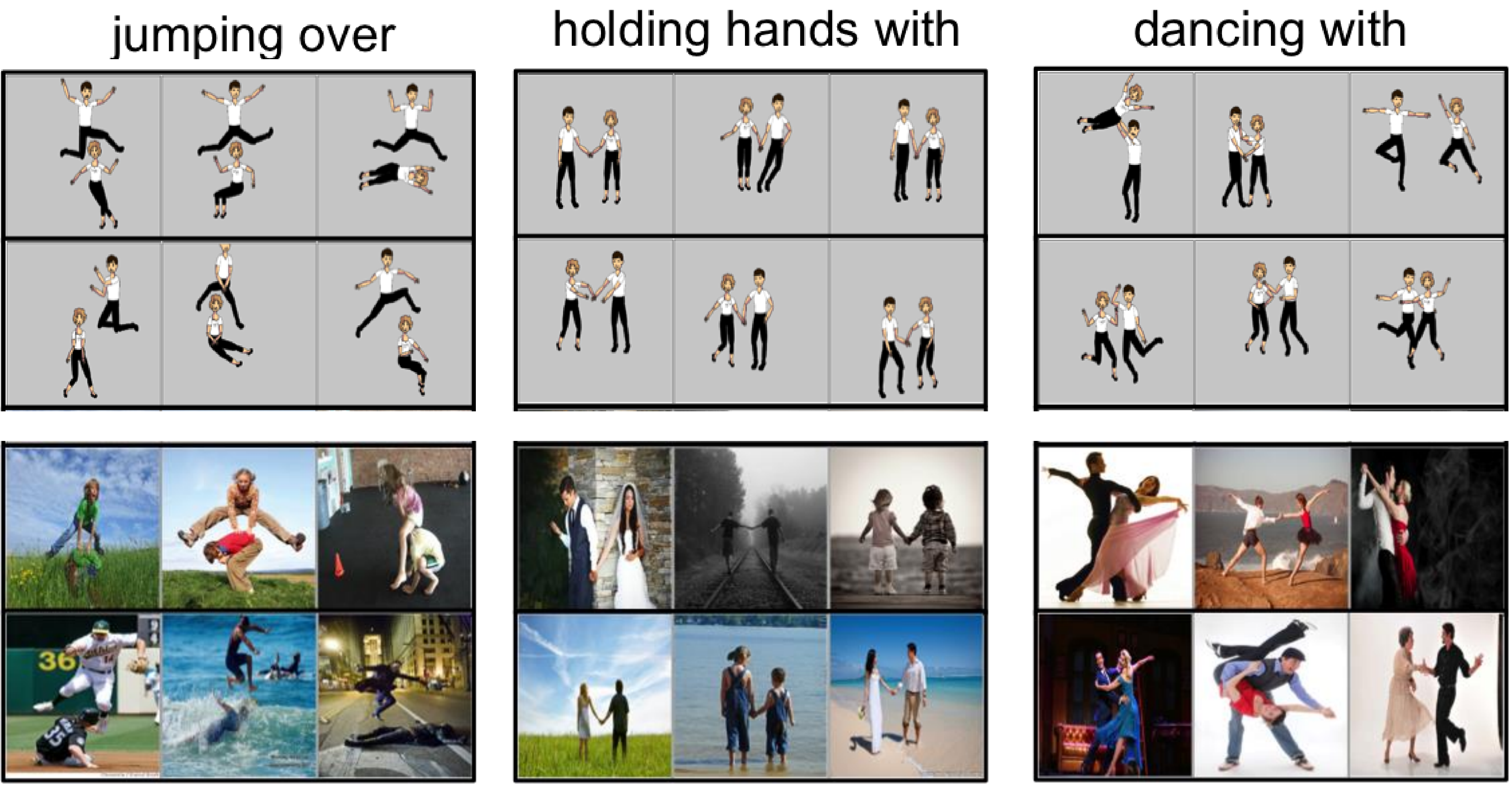


3x

[Antol, Zitnick and Parikh, ECCV 2014]

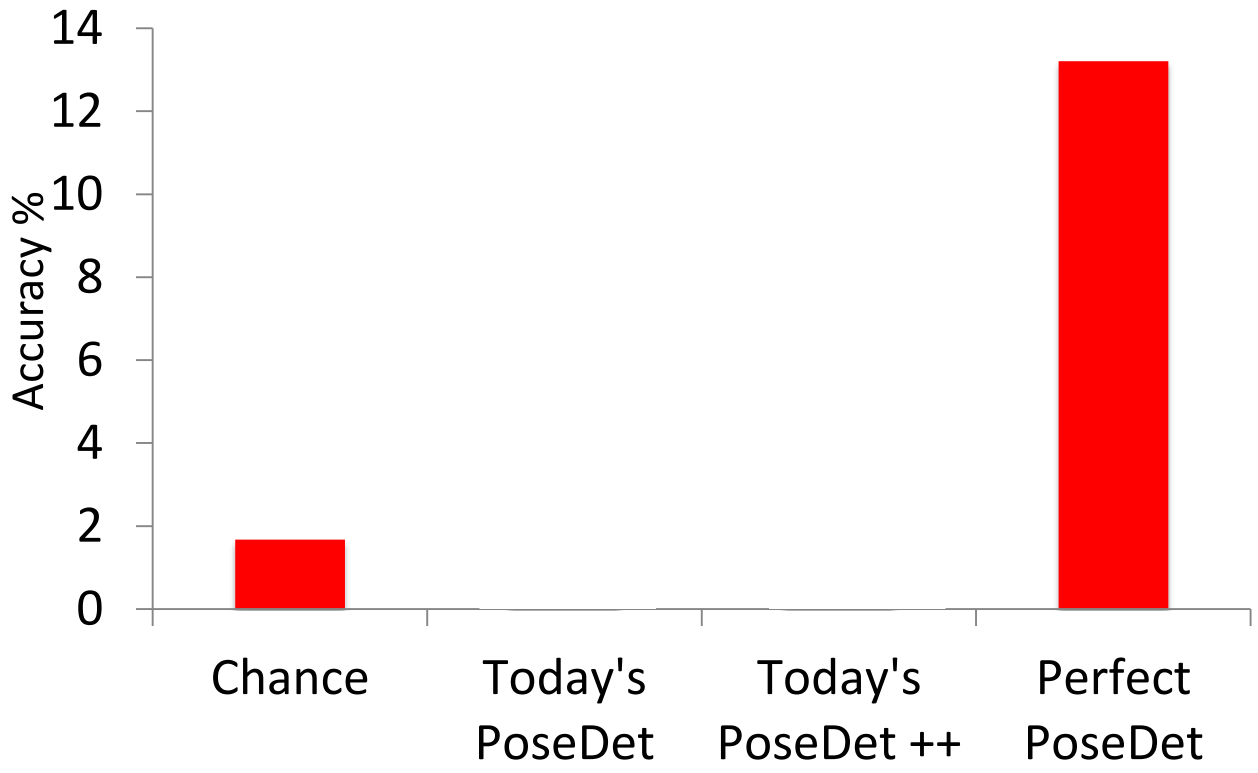
Slide credit: Devi Parikh

Learning Fine-grained Interactions



Train on clipart, test on real

Results: 60 categories



[Antol, Zitnick and Parikh, ECCV 2014]

Slide credit: Devi Parikh

Learning Common Sense

- Assess plausibility of relations
  - man holds meal
  - tree grows in table
- Plausibility: similarity to other relations we know are plausible
  - person holds sandwich
  - man eats pizza
  - ...
- Textual and visual similarity

Slide credit: Devi Parikh

Slide credit: Devi Parikh

Results

Given *any* tuple, can assess its plausibility

	Average Precision	Rank Correlation
Text alone		
Visual alone		
Text + visual		

[Vedantam, Lin, Batra, Zitnick, and Parikh, ICCV 2015]

Slide credit: Devi Parikh

Online Demo

Online Demo **Home**

Predicting Plausibility of Common Sense Assertions

Based on *Ramakrishna Vedantam**, *Xiao Lin**, *Tanmay Batra*, *C. Lawrence Zitnick*, *Devi Parikh*, **Learning Common Sense Through Visual Abstraction**, *ICCV 2015. *Equal Contribution*

Demo prepared by Arijit Ray.

Enter the Primary, Relation and Secondary Phrases of a Tuple whose plausibility you want to assess:

Test!

Examples

Or, submit your text file

Entered Tuple : man eats cake

Predicted Plausibility Score : 0.3096

Show More Details

Slide credit: Devi Parikh

Fill-in-the-blank:

Mike is having lunch when he sees a bear. _____________.

A. Mike orders a pizza.
B. Mike hugs the bear.
C. Bears are mammals.
D. Mike tries to hide.

Slide credit: Devi Parikh

Fill-in-the-blank

Question

_________________. Mike is wearing a blue cap. Mike is telling Jenny to get off the swing

Options and Generated Scenes

A. There is a tree near a table.

B. The brown dog is standing next to Mike.

C. The sun is in the sky.

D. Jenny is standing dangerously on the swing

Slide credit: Devi Parikh

[Zitnick, Parikh and Vanderwende, ICCV 2013]

Visual Paraphrasing

It is a sunny day.
Mike is sitting with a pizza.
Jenny is playing with a soccer ball.

Mike is eating a pizza.
Jenny is playing soccer.
A cat is eating a hot dog.

Same or different?

Slide credit: Devi Parikh

Results

	Fill-in-the-blanks (FITB) Accuracy (+/- ~0.15)	Visual Paraphrasing (VP) AP (+/- ~0.02)
Random	25.00	33.33

[Lin and Parikh, CVPR 2015]

Slide credit: Devi Parikh

Visual Abstraction For…

- Studying mappings betw
- Zero-sh
- Studyin

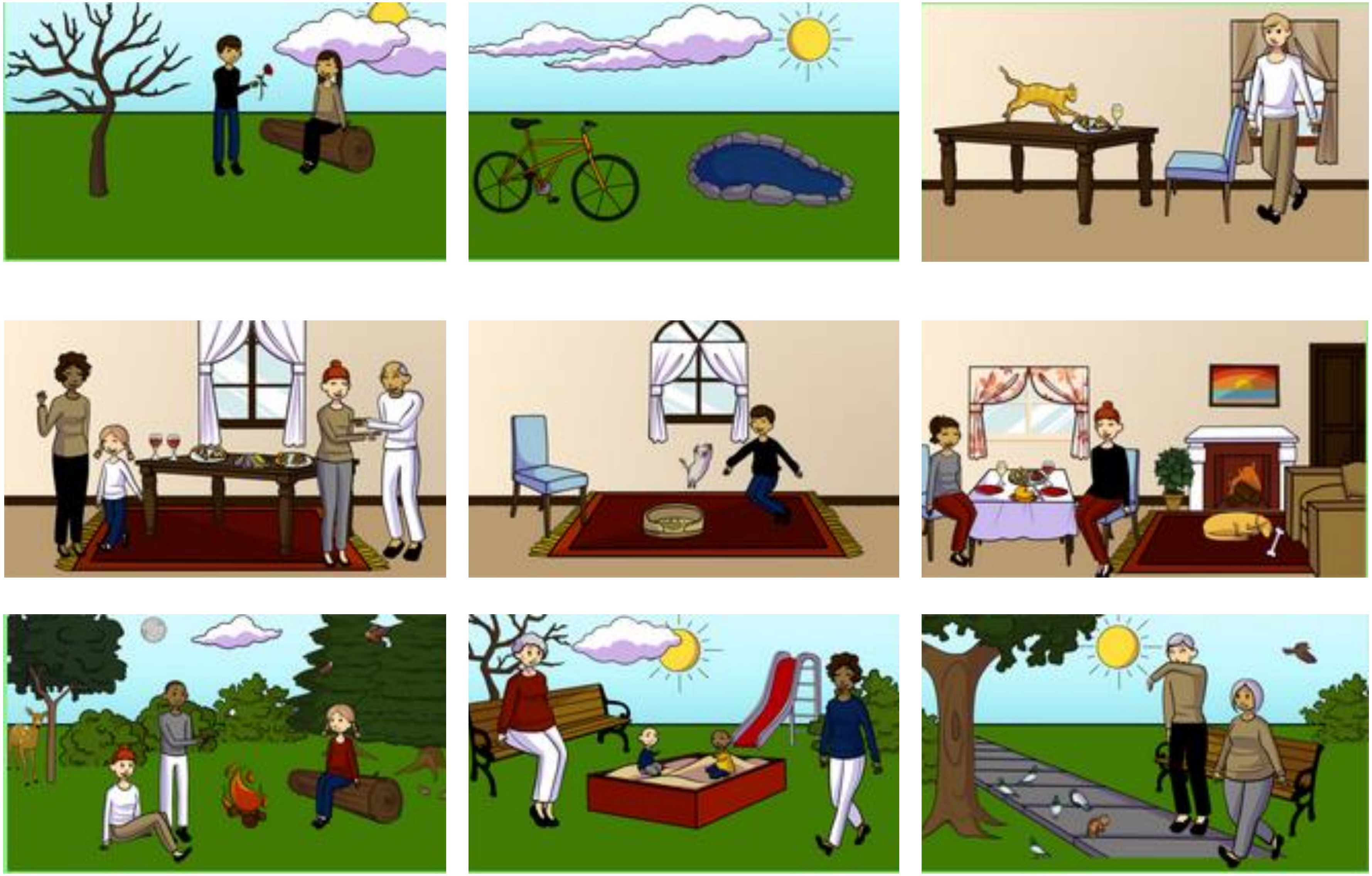
Study high-level image understanding tasks without waiting for lower-level vision tasks to be solved

- Learning common sense knowledge

- Rich annotation modality
  - Ask for d
  - Ask for sc
  - Show sce
  - Perturb a scene and ask for descriptions
  - …

Learning by playing!

Slide credit: Devi Parikh



50k scenes   Available online!

Slide credit: Devi Parikh

Semantic Image Understanding

“Color College Avenue”, Blacksburg, VA, May 2012

Slide credit: Devi Parikh

Semantic Image Understanding

Slide credit: Devi Parikh

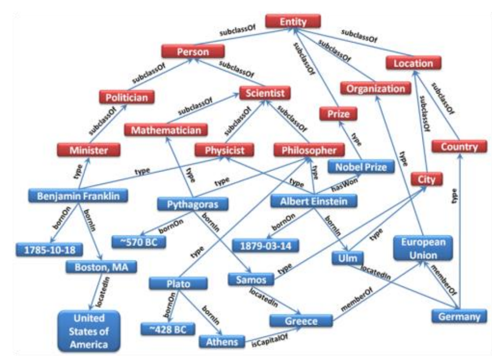
Semantic Image Understanding

Words

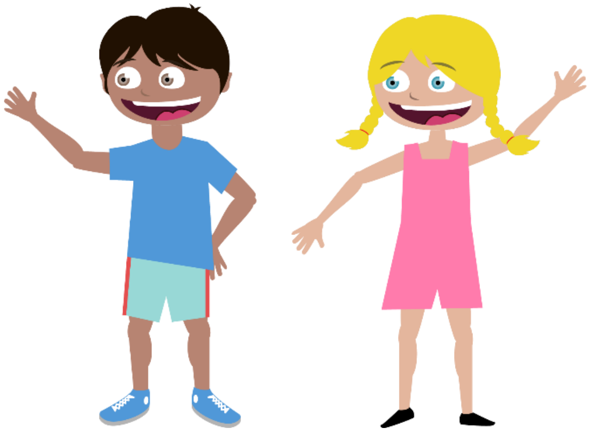
Mike is wearing a blue hat with a star.
<<Mike>, <wear>, <hat>>
<<hat>, <with>, <star>>
Jenny is happy to see Mike.
<<Jenny>, <be happy>, <>>
<<Jenny>, <see>, <Mike>>

Pictures

Reasoning
(Common Sense,
Knowledge Base)



Thank you.



Slide credit: Devi Parikh