# Dual-Path Convolutional Image-Text Embeddings with Instance Loss

Candidate Assessment

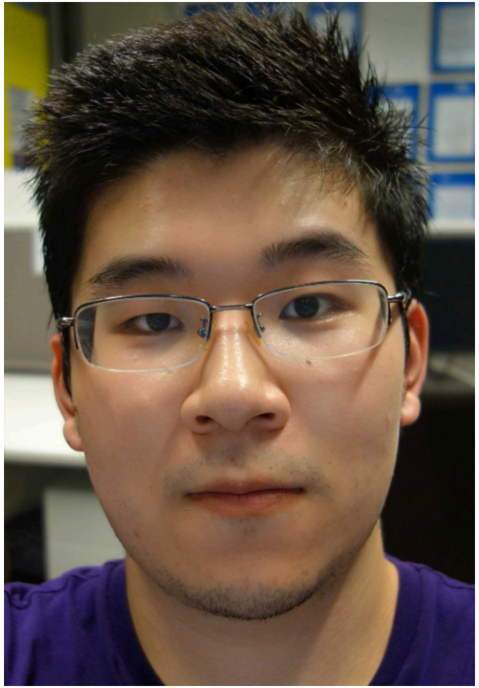
Zhedong Zheng

Centre for Artificial Intelligence
Faculty of Engineering and Information Technology
University of Technology Sydney

# About Me

**Present**

- **3rd year PhD student**
- Advised by Prof. Yi Yang and Dr. Liang Zheng
- Published two top-conference papers and two journal papers



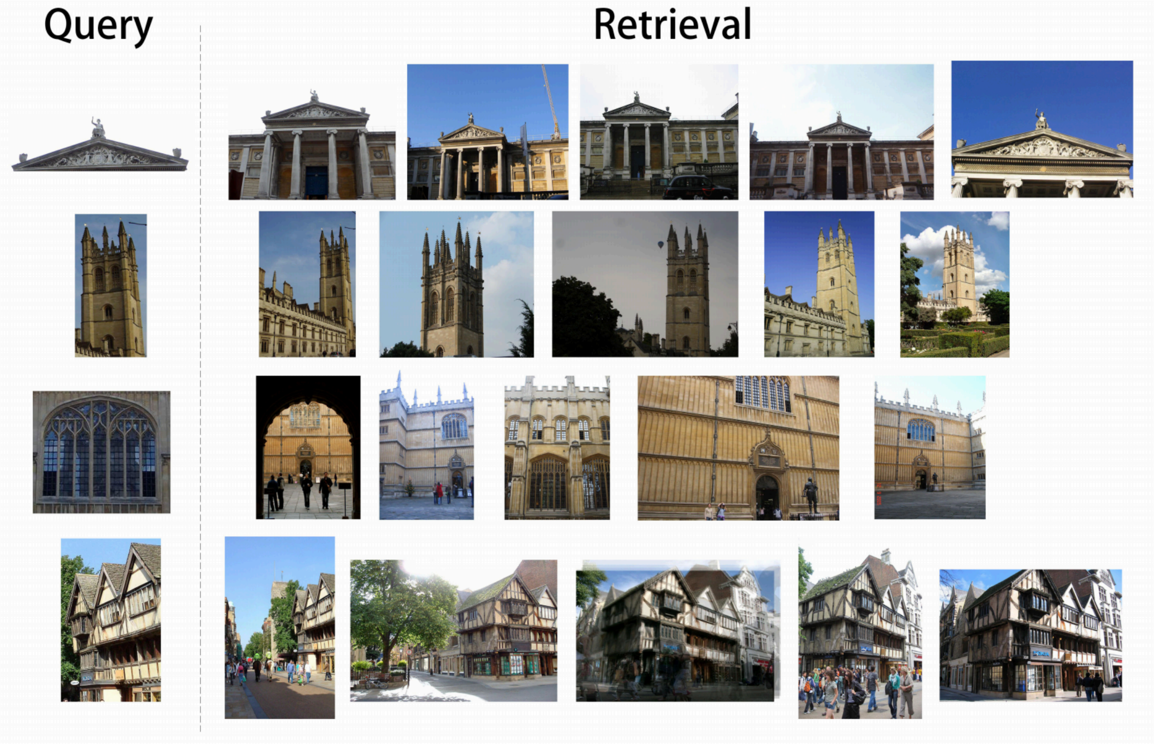
**Research Interests**

Computer Vision, Image Retrieval, Image-text Understanding,

Image Generation, Generative Adversarial Networks

Zhedong.Zheng@student.uts.edu.au

# Single-modal Retrieval



# What is Multi-modal Retrieval ?

"A boy playing basketball in a gym"

"A little girl sits in a plastic swing set ."

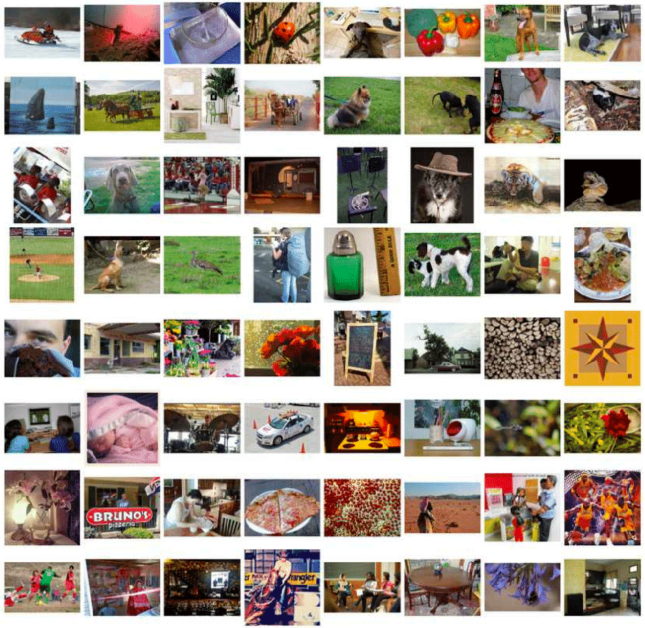
# What is Multi-modal Retrieval ?

1. Brown and white dog yawning .
2. A dog with its mouth opened .
3. Dog yawns
4. The dog 's mouth is open like he is yawning .
5. Closeup of dog in profile with mouth open .

1. The tennis player is wearing a yellow and blue shirt and a blue headband .
2. a tennis player wearing a yellow , white and blue shirt carrying a racquet
3. A tennis player is carrying a tennis racket .
4. A male athlete is wearing a teal sweat-band and a shirt from Nike and is holding a tennis racket .
5. A tennis player in an orange outfit hits a ball .

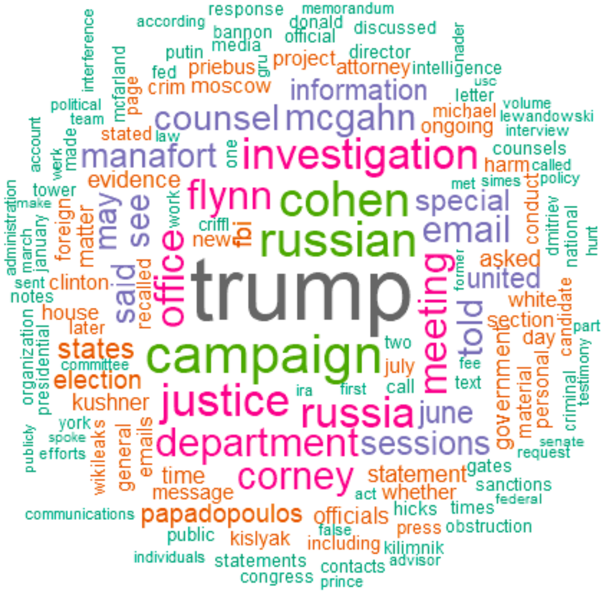
# Main Challenge

Images




?

Sentences



# What should we care about?

# What should we care about?

- Better Features

Are the off-the-shelf features good?

- Faster Inference Speed

RNN needs wait the former output.

- Scalable to Large Datasets

We evaluate our methods on two large-scale datasets.

# What should we care about?

- **Better Features**

Are the off-the-shelf features good?

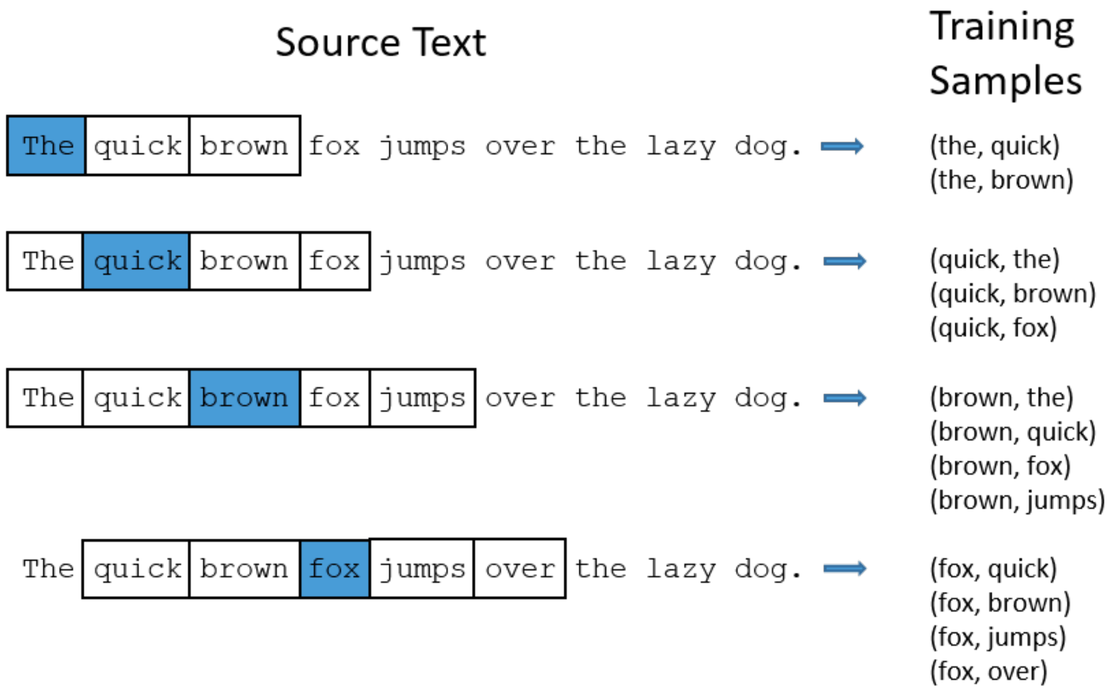
- Fast Inference Speed

RNN needs wait the former output.

- Scalable to Large Datasets

We evaluate our methods on two large-scale datasets.

# Word2vec




T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv: 1301.3781, 2013

# Word2vec may learn similar representation for keywords.

The quick **brown** fox jumps over the lazy dog.

The quick **grey** fox jumps over the lazy dog.

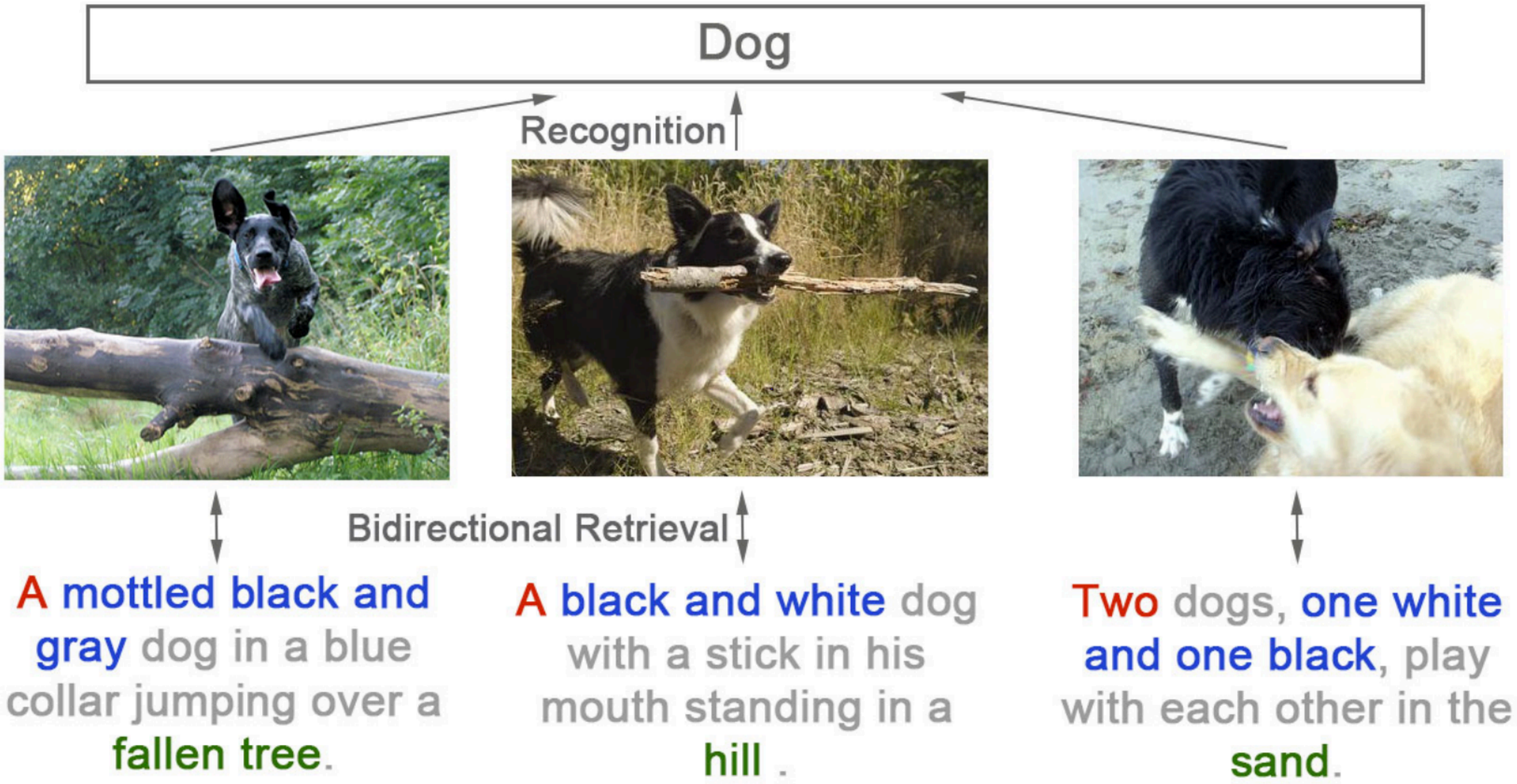
# Word2vec may learn similar representation for keywords.

The quick brown **fox** jumps over the lazy dog.

The quick brown **dog** jumps over the lazy fox.

CNN model trained on ImageNet is not perfect.

# CNN model trained on ImageNet



# What should we care about?

- **Better Features**

Are the off-the-shelf features good? **No.**

- Faster Inference Speed

RNN needs wait the former output.

- Scalable to Large Datasets

We evaluate our methods on two large-scale datasets.

# Instance Loss (Based on an unsupervised assumption)

1. A light brown dog with his tail in the air jumps of a pontoon toward the water .

...

5. a gray and brown dog jumps off a dock into a lake

1.A dog playing with a dog toy as someone tries to pull it from its mouth .

...

5.The photographer is playing tug-of-war with a dog .

1. one man wearing a gray shirt and a backpack with snowy mountains in the backgroud

...

5. A man in a blue shirt sitting on the side of a mountain wea-ring a backpack .

# Instance Loss Definition

**Formulation.** For two modalities, we formulate two classification objectives as follows,

Shared Classifier

$$P_{visual} = softmax(\boxed{W_{share}^T} f_{img}), \tag{4.5}$$

$$L_{visual} = -\log(P_{visual}(c)), \tag{4.6}$$

$$P_{textual} = softmax(\boxed{W_{share}^T} f_{text}), \tag{4.7}$$

$$L_{textual} = -\log(P_{text}(c)), \tag{4.8}$$

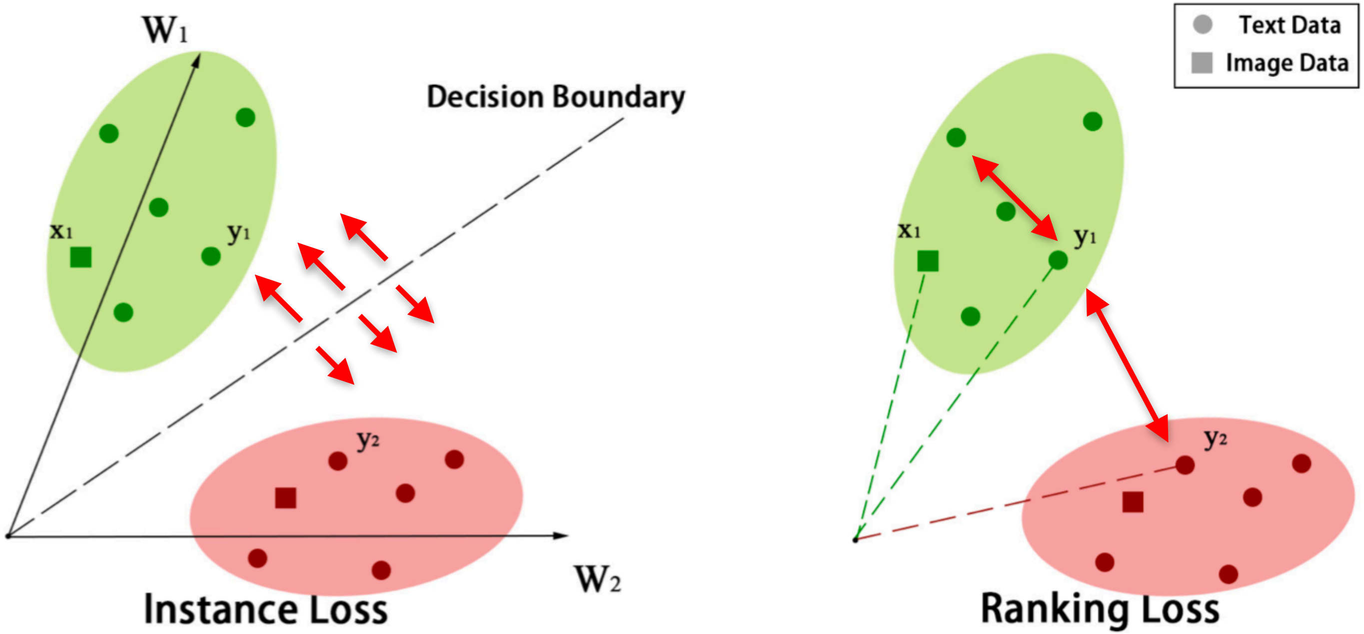
where $f_{img}$ and $f_{text}$ are image and text features defined in Eq. 4.1 and Eq. 4.3, respectively. $W_{share}$ is the parameter of the final fully connected layer (Fig. 4.1).

# Ranking Loss Definition

$$L_{rank} = \overbrace{max[0, \alpha - (D(f_{I_a}, f_{T_a}) - D(f_{I_a}, f_{T_n}))]}^{image\ \ anchor} + \underbrace{max[0, \alpha - (D(f_{T_a}, f_{I_a}) - D(f_{T_a}, f_{I_n}))]}_{text\ \ anchor},$$

$$L = \lambda_1 L_{rank} + \lambda_2 L_{visual} + \lambda_3 L_{textual},$$

# Instance Loss + Ranking Loss



# What should we care about?

- **Better Features**

Are the pretext tasks good? **No**

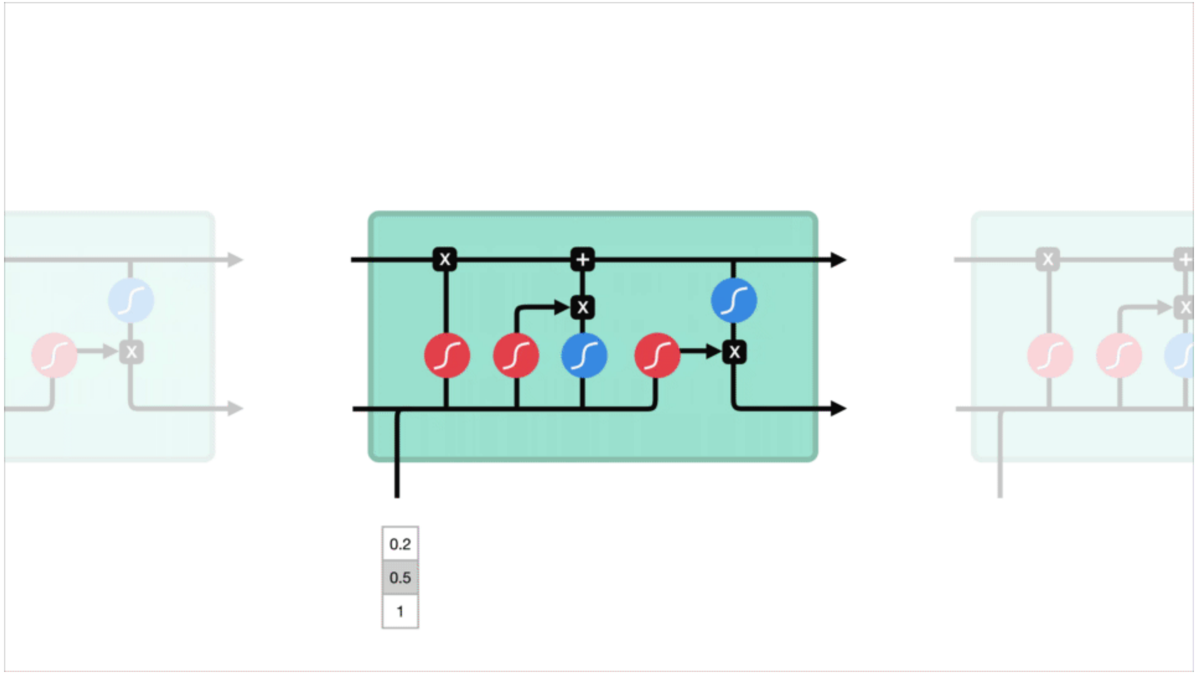
- **Faster Inference Speed**

RNN needs wait the former output.

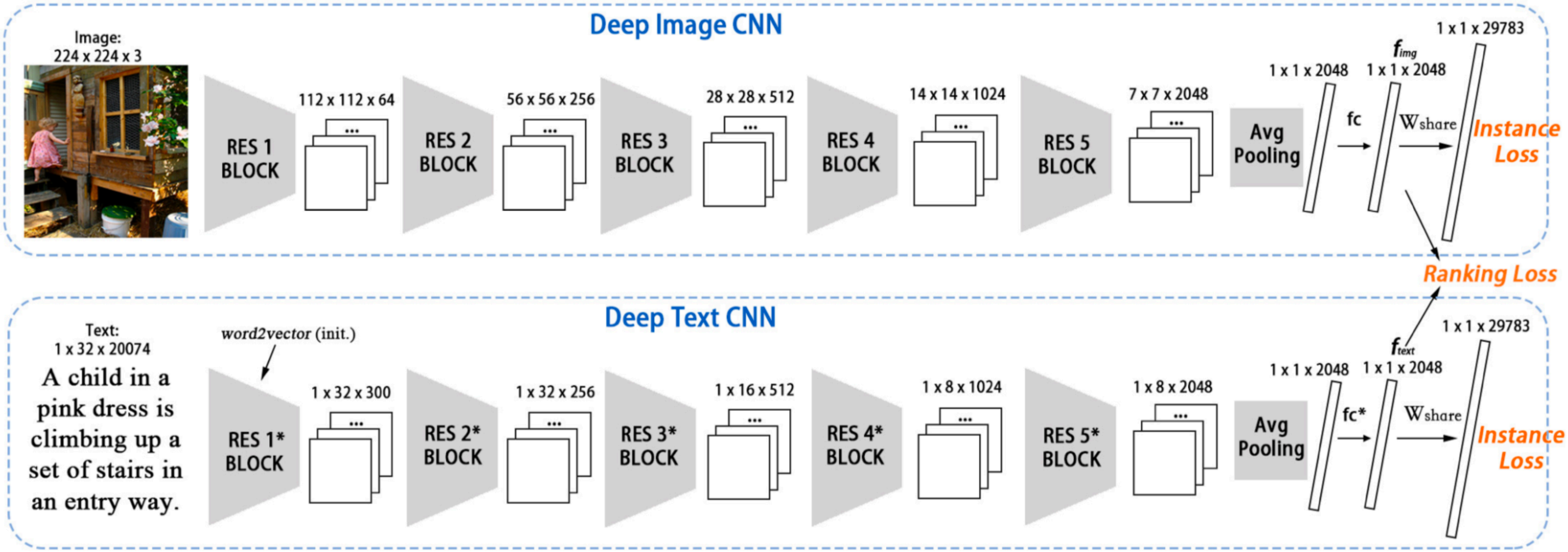
- Scalable to Large Datasets

We evaluate our methods on two large-scale datasets.

# CNN+RNN



# **CNN+CNN**: Dual-path Convolutional Neural Network



# **CNN+CNN**: Dual-path Convolutional Neural Network

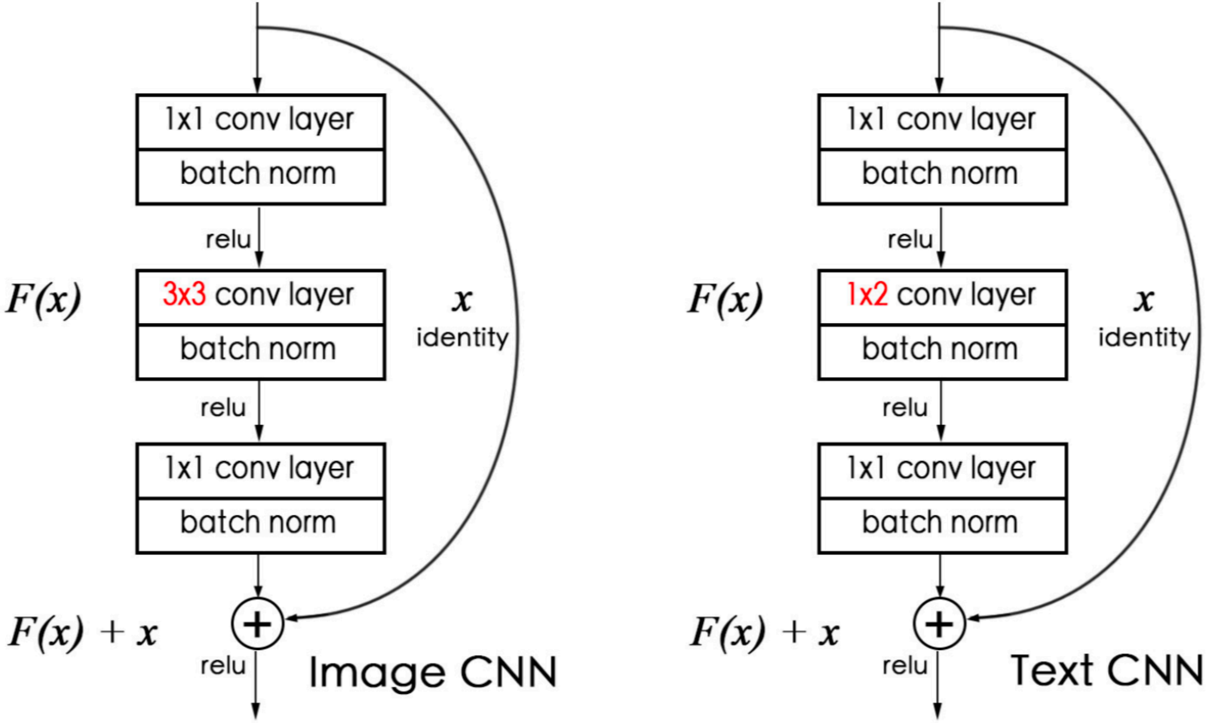
A child in a pink dress is climbing upon a set of stairs in an entry way.

224 x 224 x 3

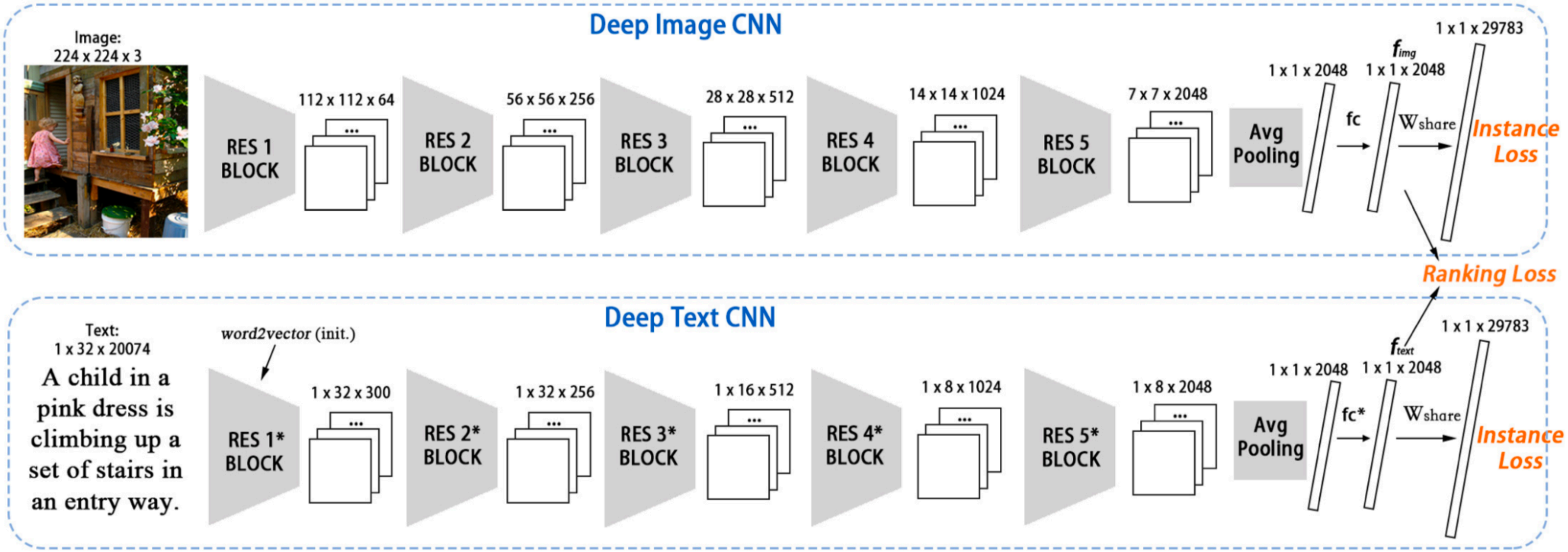
1 x Length x Dictionary Size

1 x 32 x 20074

# **CNN+CNN**: Dual-path Convolutional Network



# **CNN+CNN**: Dual-path Convolutional Neural Network




End-to-End Training: From Raw Input to the Final Objectives

# What should we care about?

- **Better Features**

Are the pretext tasks good? **No**

- **Fast Inference Speed**

RNN needs wait the former output.

- **Scalable to Large Datasets**

We evaluate our methods on two large-scale datasets.

# Experiment

# Datasets

- **Flickr30k:**

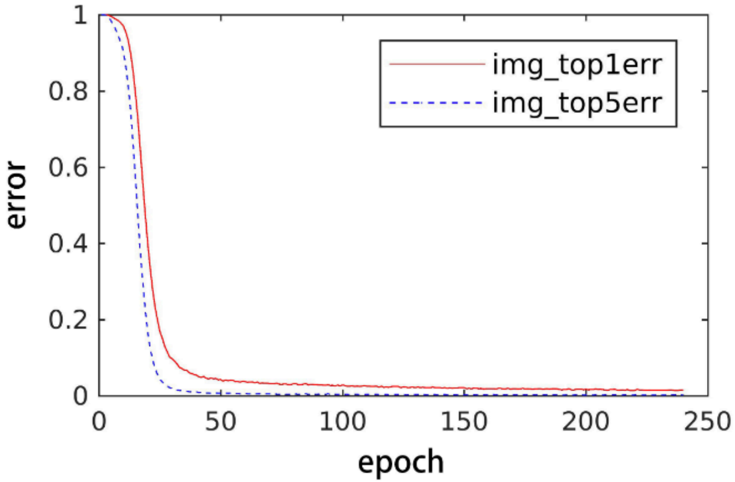
31,783 images with 158,915 captions. The average sentence length is 19.6 words after we remove rare words.

- **MSCOCO:**

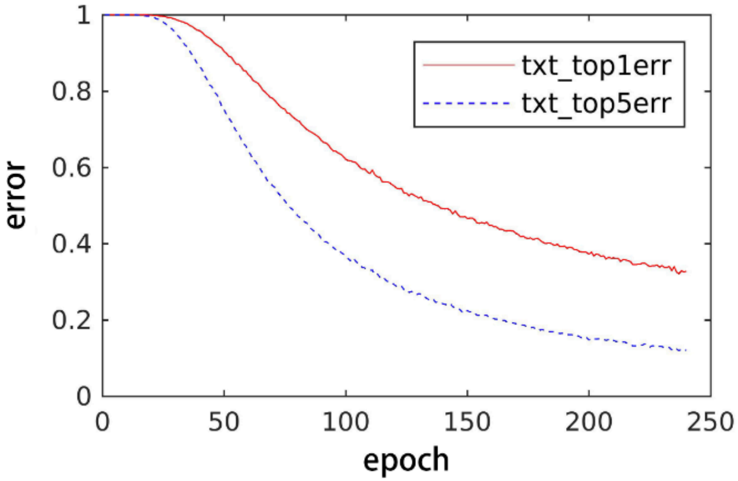
123,287 images with 616,767 captions. The average length of captions is 8.7 after rare word removal.

# Convergence

Although we may face large class number, every class has limited samples.




(a) Image CNN




(b) Text CNN

Fig. 8. Classification error curves when training on Flickr30k. The image CNN (a) and text CNN (b) converge well with 29,783 training classes (image / text groups).

# Flickr30k

| Method | Visual | Textual | Image Query R@1 | Image Query R@5 | Image Query R@10 | Image Query Med | Text Query R@1 | Text Query R@5 | Text Query R@10 | Text Query Med $r$ |
|---|---|---|---|---|---|---|---|---|---|---|
| DeVise [5] | ft AlexNet | ft skip-gram | 4.5 | 18.1 | 29.2 | 26 | 6.7 | 21.9 | 32.7 | 25 |
| Deep Fragment [6] | ft RCNN | fixed word vector from [58] | 16.4 | 40.2 | 54.7 | 8 | 10.3 | 31.4 | 44.5 | 13 |
| DCCA [59] | ft AlexNet | TF-IDF | 16.7 | 39.3 | 52.9 | 8 | 12.6 | 31.0 | 43.0 | 15 |
| DVSA [32] | ft RCNN (init. on Detection) | w2v + ft RNN | 22.2 | 48.2 | 61.4 | 4.8 | 15.2 | 37.7 | 50.5 | 9.2 |
| LRCN [60] | ft VGG-16 | ft RNN | 23.6 | 46.6 | 58.3 | 7 | 17.5 | 40.3 | 50.8 | 9 |
| m-CNN [7] | ft VGG-19 | 4 × ft CNN | 33.6 | 64.1 | 74.9 | 3 | 26.2 | 56.3 | 69.6 | 4 |
| VQA-A [18] | fixed VGG-19 | ft RNN | 33.9 | 62.5 | 74.5 | - | 24.9 | 52.6 | 64.8 | - |
| GMM-FV [17] | fixed VGG-16 | w2v + GMM + HGLMM | 35.0 | 62.0 | 73.8 | 3 | 25.0 | 52.7 | 66.0 | 5 |
| m-RNN [16] | fixed VGG-16 | ft RNN | 35.4 | 63.8 | 73.7 | 3 | 22.8 | 50.7 | 63.1 | 5 |
| RNN-FV [19] | fixed VGG-19 | feature from [17] | 35.6 | 62.5 | 74.2 | 3 | 27.4 | 55.9 | 70.0 | 4 |
| HM-LSTM [21] | fixed RCNN from [32] | w2v + ft RNN | 38.1 | - | 76.5 | 3 | 27.7 | - | 68.8 | 4 |
| SPE [8] | fixed VGG-19 | w2v + HGLMM | 40.3 | 68.9 | 79.9 | - | 29.7 | 60.1 | 72.1 | - |
| sm-LSTM [20] | fixed VGG-19 | ft RNN | 42.5 | 71.9 | 81.5 | 2 | 30.2 | 60.4 | 72.3 | 3 |
| RRF-Net [61] | fixed ResNet-152 | w2v + HGLMM | 47.6 | 77.4 | 87.1 | - | 35.4 | 68.3 | 79.9 | - |
| 2WayNet [49] | fixed VGG-16 | feature from [17] | 49.8 | 67.5 | - | - | 36.0 | 55.6 | - | - |
| DAN (VGG-19) [9] | fixed VGG-19 | ft RNN | 41.4 | 73.5 | 82.5 | 2 | 31.8 | 61.7 | 72.5 | 3 |
| DAN (ResNet-152) [9] | fixed ResNet-152 | ft RNN | 55.0 | 81.8 | 89.0 | 1 | **39.4** | 69.2 | 79.1 | 2 |
| Ours (VGG-19) Stage I | fixed VGG-19 | ft ResNet-50† (w2v init.) | 37.5 | 66.0 | 75.6 | 3 | 27.2 | 55.4 | 67.6 | 4 |
| Ours (VGG-19) Stage II | ft VGG-19 | ft ResNet-50† (w2v init.) | 47.6 | 77.3 | 87.1 | 2 | 35.3 | 66.6 | 78.2 | 3 |
| Ours (ResNet-50) Stage I | fixed ResNet-50 | ft ResNet-50† (w2v init.) | 41.2 | 69.7 | 78.9 | 2 | 28.6 | 56.2 | 67.8 | 4 |
| Ours (ResNet-50) Stage II | ft ResNet-50 | ft ResNet-50† (w2v init.) | 53.9 | 80.9 | **89.9** | **1** | 39.2 | **69.8** | 80.8 | **2** |
| Ours (ResNet-152) Stage I | fixed ResNet-152 | ft ResNet-152† (w2v init.) | 44.2 | 70.2 | 79.7 | 2 | 30.7 | 59.2 | 70.8 | 4 |
| Ours (ResNet-152) Stage II | ft ResNet-152 | ft ResNet-152† (w2v init.) | **55.6** | **81.9** | 89.5 | **1** | 39.1 | 69.2 | **80.9** | **2** |

# MSCOCO

| Method | Visual | Textual | Image Query | | | | Text Query | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | R@1 | R@5 | R@10 | Med | R@1 | R@5 | R@10 | Med $r$ |
| 1K test images | | | | | | | | | | |
| DVSA [32] | ft RCNN | w2v + ft RNN | 38.4 | 69.9 | 80.5 | 1 | 27.4 | 60.2 | 74.8 | 3 |
| GMM-FV [17] | fixed VGG-16 | w2v + GMM + HGLMM | 39.4 | 67.9 | 80.9 | 2 | 25.1 | 59.8 | 76.6 | 4 |
| m-RNN [16] | fixed VGG-16 | ft RNN | 41.0 | 73.0 | 83.5 | 2 | 29.0 | 42.2 | 77.0 | 3 |
| RNN-FV [19] | fixed VGG-19 | feature from [17] | 41.5 | 72.0 | 82.9 | 2 | 29.2 | 64.7 | 80.4 | 3 |
| m-CNN [7] | ft VGG-19 | $4 \times$ ft CNN | 42.8 | 73.1 | 84.1 | 2 | 32.6 | 68.6 | 82.8 | 3 |
| HM-LSTM [21] | fixed CNN from [32] | ft RNN | 43.9 | - | 87.8 | 2 | 36.1 | - | 86.7 | 3 |
| SPE [8] | fixed VGG-19 | w2v + HGLMM | 50.1 | 79.7 | 89.2 | - | 39.6 | 75.2 | 86.9 | - |
| VQA-A [18] | fixed VGG-19 | ft RNN | 50.5 | 80.1 | 89.7 | - | 37.0 | 70.9 | 82.9 | - |
| sm-LSTM [20] | fixed VGG-19 | ft RNN | 53.2 | 83.1 | 91.5 | 1 | 40.7 | 75.8 | 87.4 | 2 |
| 2WayNet [49] | fixed VGG-16 | feature from [17] | 55.8 | 75.2 | - | - | 39.7 | 63.3 | - | - |
| RRF-Net [61] | fixed ResNet-152 | w2v + HGLMM | 56.4 | 85.3 | 91.5 | - | 43.9 | 78.1 | 88.6 | - |
| Ours (VGG-19) Stage I | fixed VGG-19 | ft ResNet-50† (w2v init.) | 46.0 | 75.6 | 85.3 | 2 | 34.4 | 66.6 | 78.7 | 3 |
| Ours (VGG-19) Stage II | ft VGG-19 | ft ResNet-50† (w2v init.) | 59.4 | 86.2 | 92.9 | 1 | 41.6 | 76.3 | 87.5 | 2 |
| Ours (ResNet-50) Stage I | fixed ResNet-50 | ft ResNet-50† (w2v init.) | 52.2 | 80.4 | 88.7 | 1 | 37.2 | 69.5 | 80.6 | 2 |
| Ours (ResNet-50) Stage II | ft ResNet-50 | ft ResNet-50† (w2v init.) | **65.6** | **89.8** | **95.5** | **1** | **47.1** | **79.9** | **90.0** | **2** |
| 5K test images | | | | | | | | | | |
| GMM-FV [17] | fixed VGG-16 | w2v + GMM + HGLMM | 17.3 | 39.0 | 50.2 | 10 | 10.8 | 28.3 | 40.1 | 17 |
| DVSA [32] | ft RCNN | w2v + ft RNN | 16.5 | 39.2 | 52.0 | 9 | 10.7 | 29.6 | 42.2 | 14 |
| VQA-A [18] | fixed VGG-19 | ft RNN | 23.5 | 50.7 | 63.6 | - | 16.7 | 40.5 | 53.8 | - |
| Ours (VGG-19) Stage I | fixed VGG-19 | ft ResNet-50† (w2v init.) | 24.5 | 50.1 | 62.1 | 5 | 16.5 | 39.1 | 51.8 | 10 |
| Ours (VGG-19) Stage II | ft VGG-19 | ft ResNet-50† (w2v init.) | 35.5 | 63.2 | 75.6 | 3 | 21.0 | 47.5 | 60.9 | 6 |
| Ours (ResNet-50) Stage I | fixed ResNet-50 | ft ResNet-50† (w2v init.) | 28.6 | 56.2 | 68.0 | 4 | 18.7 | 42.4 | 55.1 | 8 |
| Ours (ResNet-50) Stage II | ft ResNet-50 | ft ResNet-50† (w2v init.) | **41.2** | **70.5** | **81.1** | **2** | **25.3** | **53.4** | **66.4** | **5** |

# Further Analysis and Discussion

# Ablation Study: Ranking Loss + Instance Loss

| Method | Stage | Image Query | | Text Query | |
|---|---|---|---|---|---|
| | | R@1 | R@10 | R@1 | R@10 |
| Only Ranking Loss | I | 6.1 | 27.3 | 4.9 | 27.8 |
| Only Instance Loss | I | 39.9 | 79.1 | 28.2 | 67.9 |
| Instance Loss + Ranking Loss | I | 37.6 | 75.1 | 24.1 | 65.6 |
| Only Instance Loss | II | 50.5 | 86.0 | 34.9 | 75.7 |
| Only Ranking Loss | II | 47.5 | 85.4 | 29.0 | 68.7 |
| Full model | II | 55.4 | 89.3 | 39.7 | 80.8 |

Table 4. Ranking loss and instance loss retrieval results on Flickr30k validation set. Except for the different losses, we apply the entirely same network (ResNet-50). For a clear comparison, we also fixed the image CNN in Stage I and tune the entire network in Stage II to observe the overfitting.

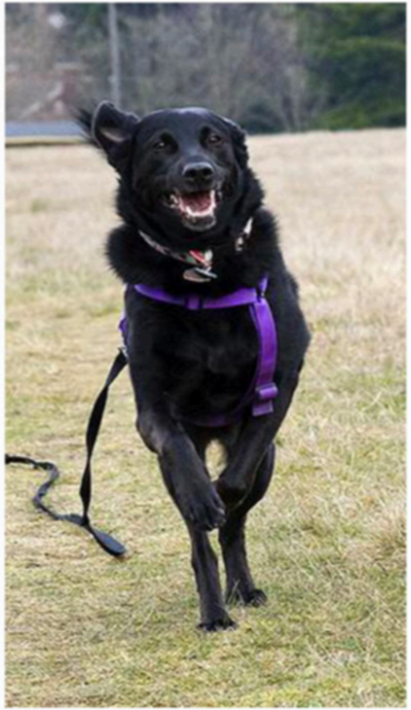
# Ablation Study: K-class Loss vs. Instance Loss

| Methods | Image-Query R@1 | Text-Query R@1 |
|---|---|---|
| 3000 categories (StageI) | 38.0 | 26.1 |
| 10000 categories (StageI) | 44.7 | 31.3 |
| Our (StageI) | 52.2 | 37.2 |

Table 5. K-class Loss vs. Instance Loss on MSCOCO. We use the K-means clustering result as pseudo categories. The experiment is based on Res50 + Res50$^{\dagger}$ as the model structure.

# Explainable

The : -0.0023
man : 0.057
dressed : 0.0085
(liked REMOVED)
an : 0.0025
indian : -0.0420
wearing : 0.0207
feathers : -0.0354
is : 0.0133
standing : -0.0305
in : -0.0127
front : -0.0341
(of REMOVED)
the : -0.0130
microphone : -0.0238



A : 0.0026
black : -0.1042
dog : -0.0219
with: -0.0000
purple : -0.0643
collar : -0.0046
black : -0.0096
leash : 0.0022
runs : 0.0044
in : -0.0021
the : -0.0013
grass : -0.0254

# Future Works

# Possible Approaches

1) Investigate the feasibility of high-fidelity **generated samples** for training. The generated samples could largely enrich the training set.
2) Mixture of **Unsupervised Learning/ Semi-supervised Learning**
3) **Domain Adaptation**

One last comment

# Neural Networks are lazy

The models could easily overfit the datasets. Sometimes **adding constraints and data augmentation** are important to train a robust network.

Training Neural Networks sometime is tricky, and models will find the short way to overfit the objective. If it is difficult to optimise, the two-step learning policy could perform well. (**Curriculum learning**)

# Questions?

The code is available at