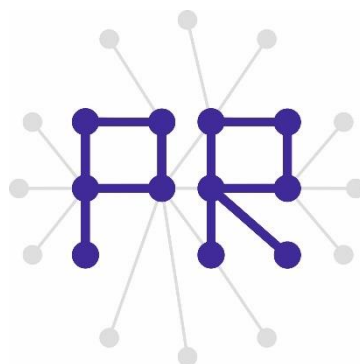


# Multimedia Retrieval Exercise Course

## 6 Query by Example: Evaluating Retrieval Results

Kimiaki Shirahama, D.E.

Research Group for Pattern Recognition  
Institute for Vision and Graphics  
University of Siegen, Germany



# Overview of Today's Lesson

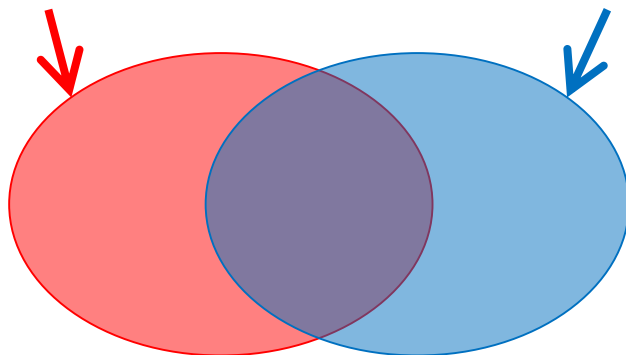
- ❑ Evaluating a Retrieval Result
  - Problem of Precision and Recall
  - Average Precision
- ❑ Visualising a Retrieval Result
  - HTML output
- ❑ Bonus: Deep Learning Features

# Problem of Precision and Recall

- Precision: Fraction of retrieved images that are relevant
- Recall: Fraction of relevant images that are retrieved  
(F-measure: Combination of precision and recall)

Set of retrieved images

Set of relevant images



Precision = (Overlapping region) / (Red region)

Recall = (Overlapping region) / (Blue region)

## Precision and Recall are for **not-ranked** results

If the number of sunflower images is 6 and a system returns images with the five highest scores as a retrieval result, (Result 1) and (Result 2) have the same precision (0.6) and recall (0.5). However, (Result 2) is intuitively better than (Result 1) because relevant images are ranked at higher positions

(Result 1)

(Result 2)

1



2



3



4



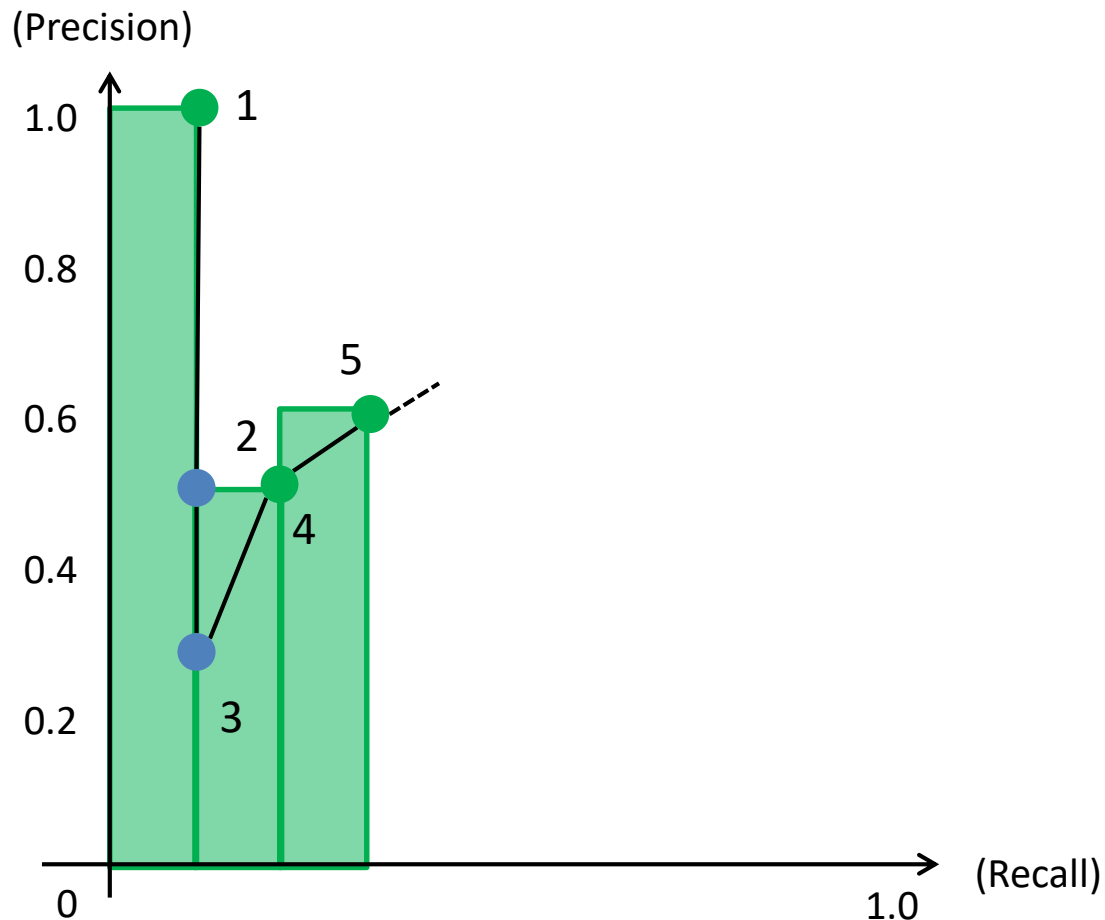
5



# Average Precision

Evaluation measure for **ranked** results

➡ (Interpolated) average value of precision at every recall level



*Approximated area under the precision-recall curve*

(Result 1)

1



2



3



4



5

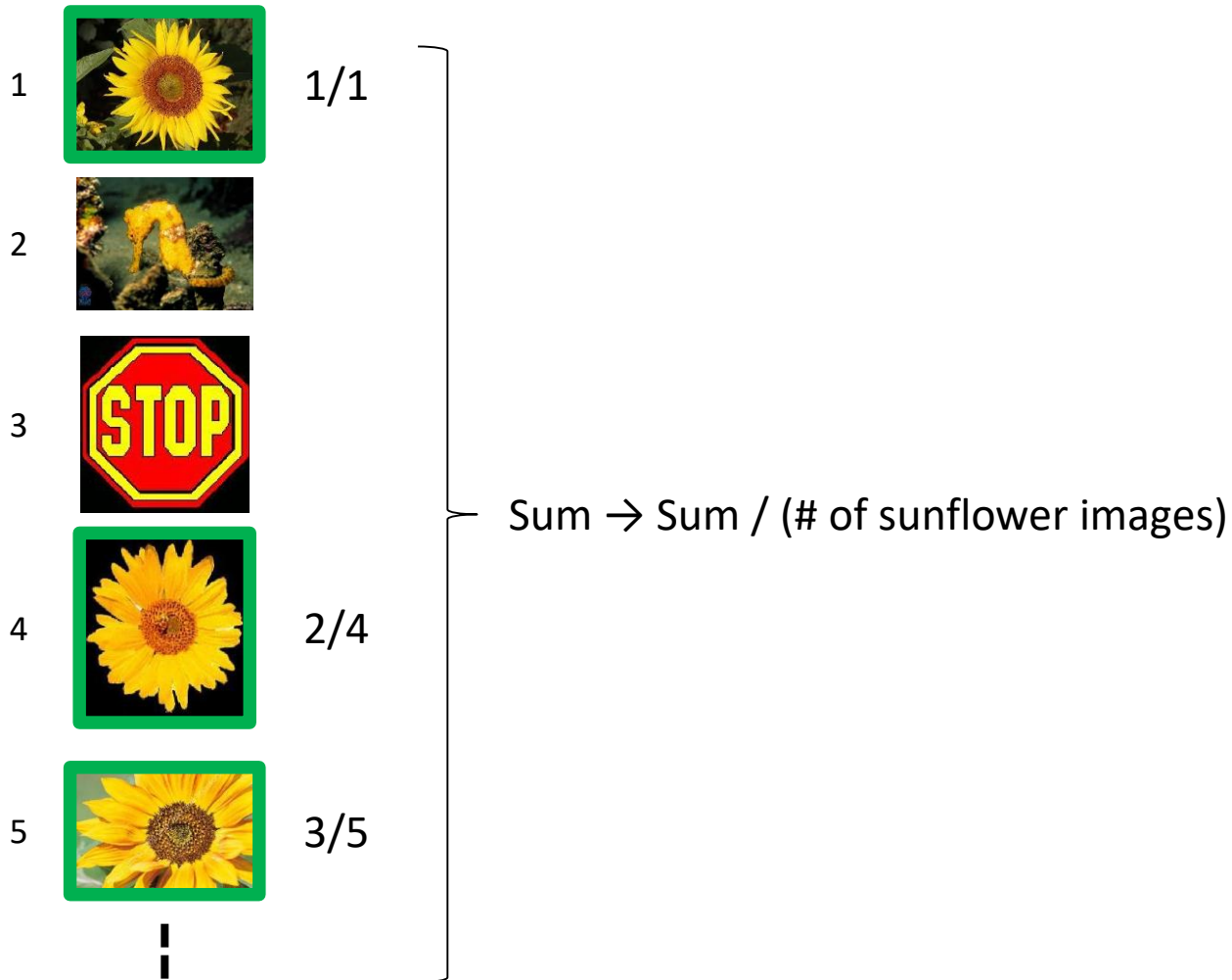


⋮

# How to Compute APs

- Stop at every relevant image, calculate the precision using images ranked above
- Sum up precisions at all relevant images, and divide the sum by the number of relevant images

(Result 1)



# Other Issues about APs

- **Mean Average Precision (MAP):** Average of APs over different query images
- APs are somehow difficult to understand  
(Based on my experience)
  - $0 \leq AP < 0.1$ : Bad retrieval
  - $0.1 \leq AP < 0.2$ : OK
  - $0.2 \leq AP < 0.3$ : Accurate
  - $0.3 \leq AP$ : Very accurate

It is said that APs of text retrieval (like Google) are about 0.6.

# Visualising a Retrieval Result

In my opinion, rather than implementing a method, it is much more important to investigate whether the method works or not (or has no bug)

➡ Output a retrieval result in **HTML format**

(Example of an HTML file showing 100 images with the highest similarities)

```
<html>
<head>
<title>Retrieval result using "query image filename" </title>
</head>
<body>
0-th ranked image: (similarity=)
<br>
1-th ranked image: (similarity=)
<br>
...
99-th ranked image: (similarity=)
<br>
</body>
</html>
```

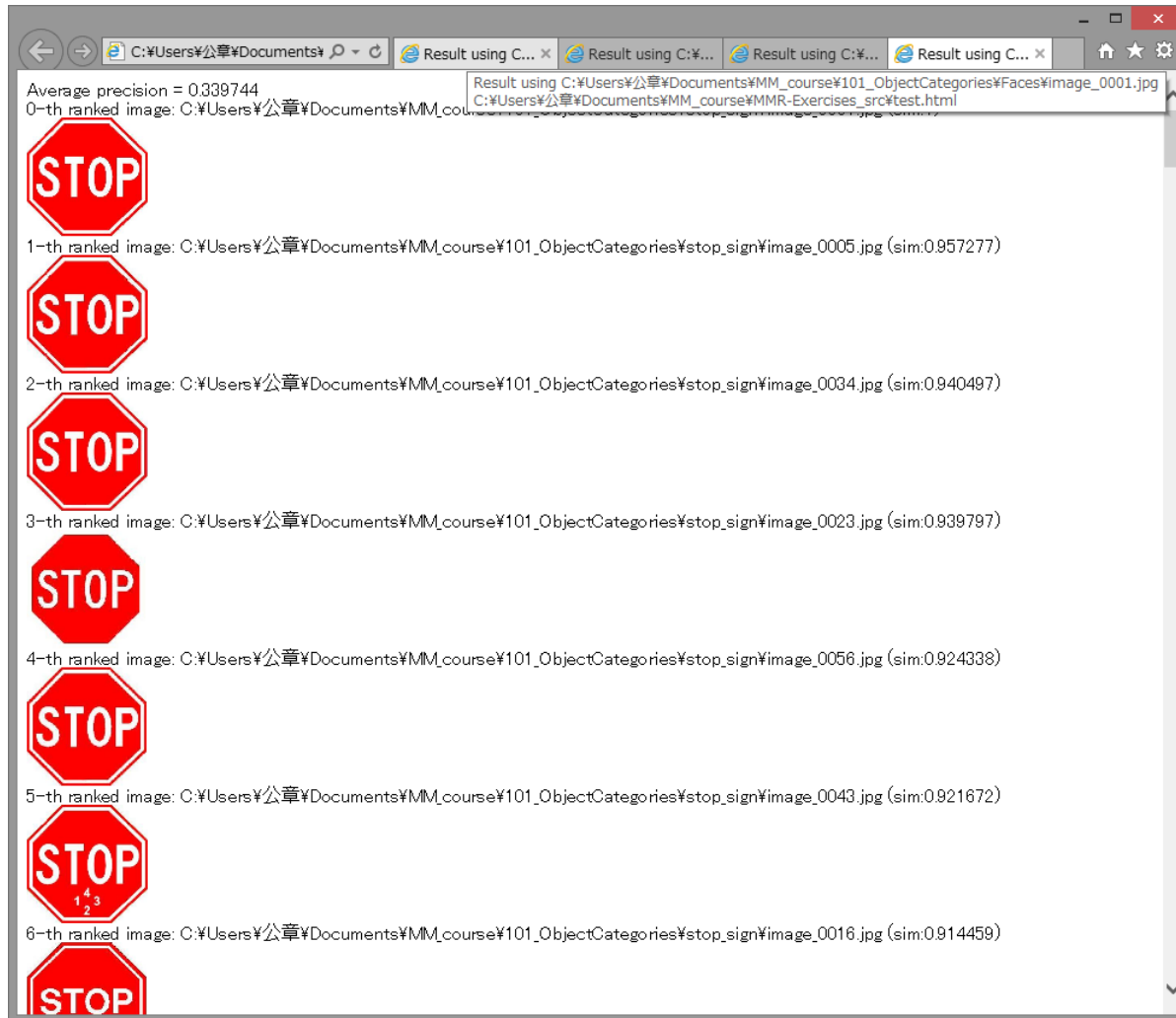
Header part

Body part

Output this kind of **TEXT** file with the file extension ".html"

# Final Task of Query by Example Retrieval System

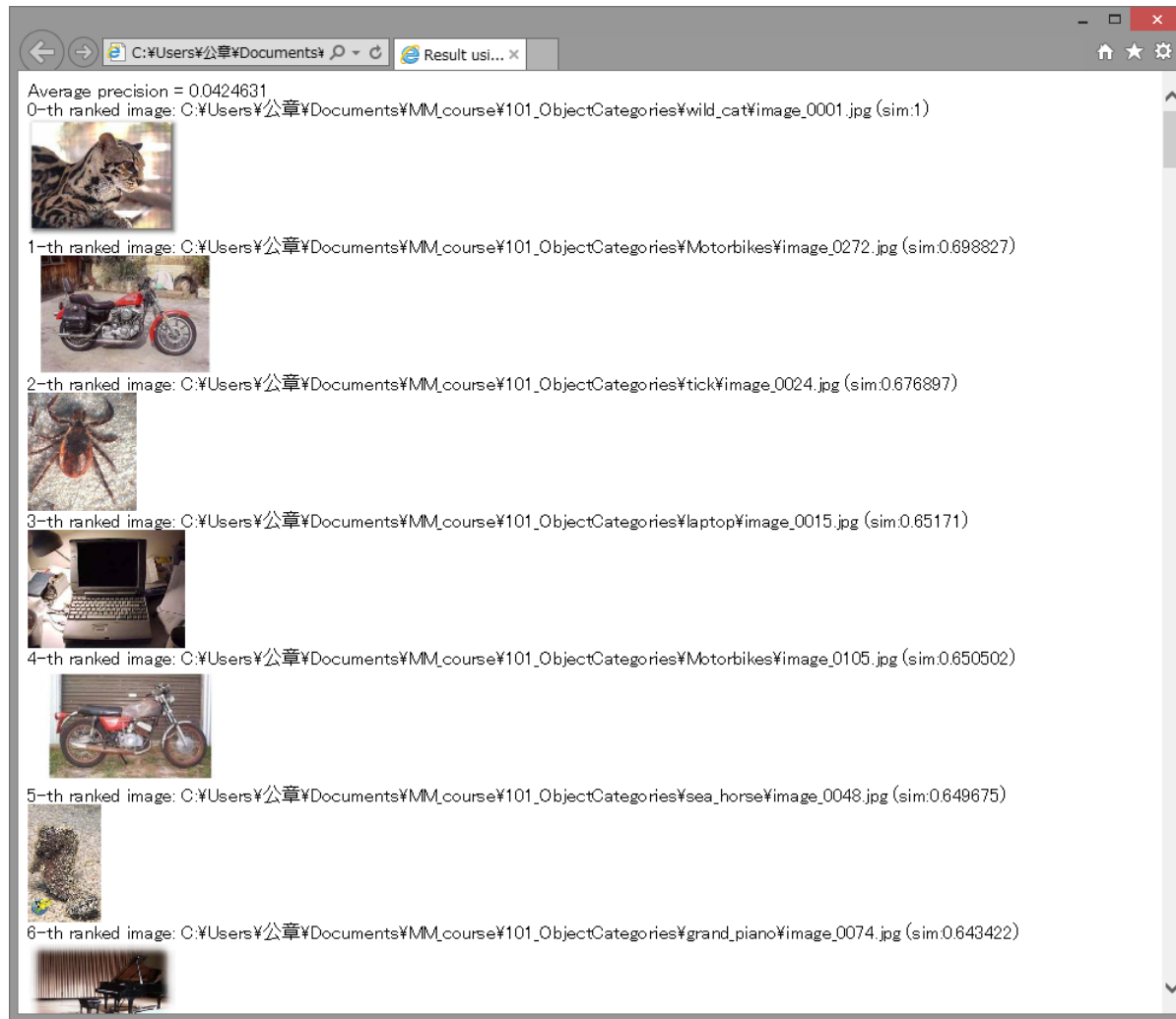
At least 10 query images, evaluate and output retrieval results



You can make a more good-looking HTML file (The design is up to you)



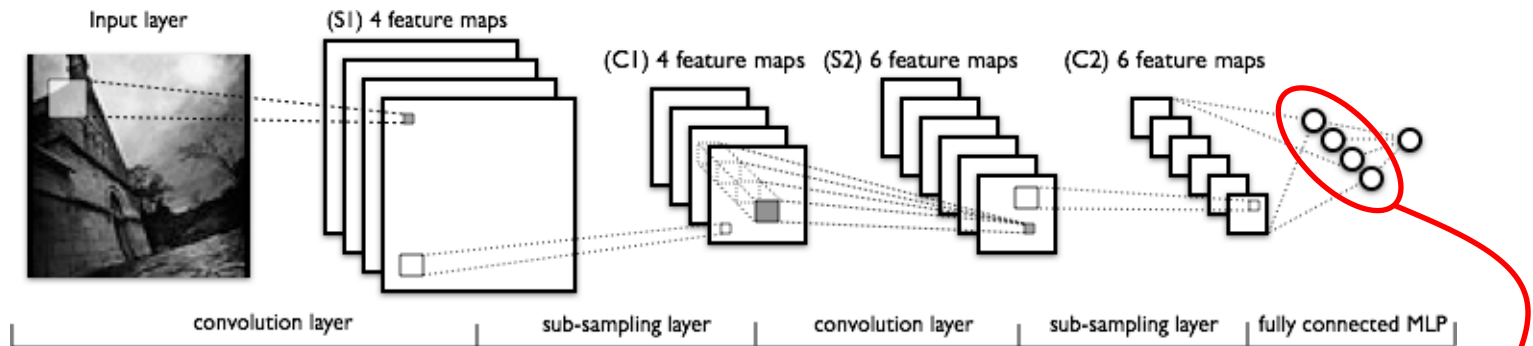
# ... And Experience the Semantic Gap



Color histogram is very simple, and works badly for several queries ☹

# Bonus: Deep Learning Features

Concept of deep learning (Convolutional Neural Network (CNN) for image classification)



(This image is taken from <http://deeplearning.net/tutorial/lenet.html#lenet>)

Outputs of neurons on a fully-connected (fc) layer can be used as a feature

**VGG16:** A popular CNN consisting of 16 convolutional layers (no longer state-of-the-art)

➡ Use neuron outputs on the fc7 layer as a feature of an image

On the web-page of this course, you can download

- File of extracted features(vgg16\_fc7\_sample.txt, about 277MB): Please change the path of each image as we did for histograms\_sample.txt
- Python code used to extract these features (extractVGG16\_fc7.py): To run this code, you need to install Tensorflow and Keras (very easy to install and use (at least on linux))

The image retrieval system that we implemented can work on fc7 features with very very small modifications! (I will explain and demonstrate in the lesson)