Class-Weighted Convolutional Features for Image Retrieval

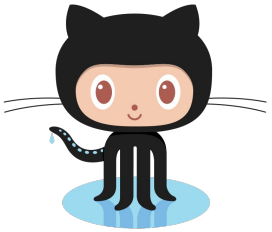
Albert Jiménez

Xavier Giró-i-Nieto

Jose Alvarez

BMVC 2017 LONDON

[arXiv]



[Code]

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

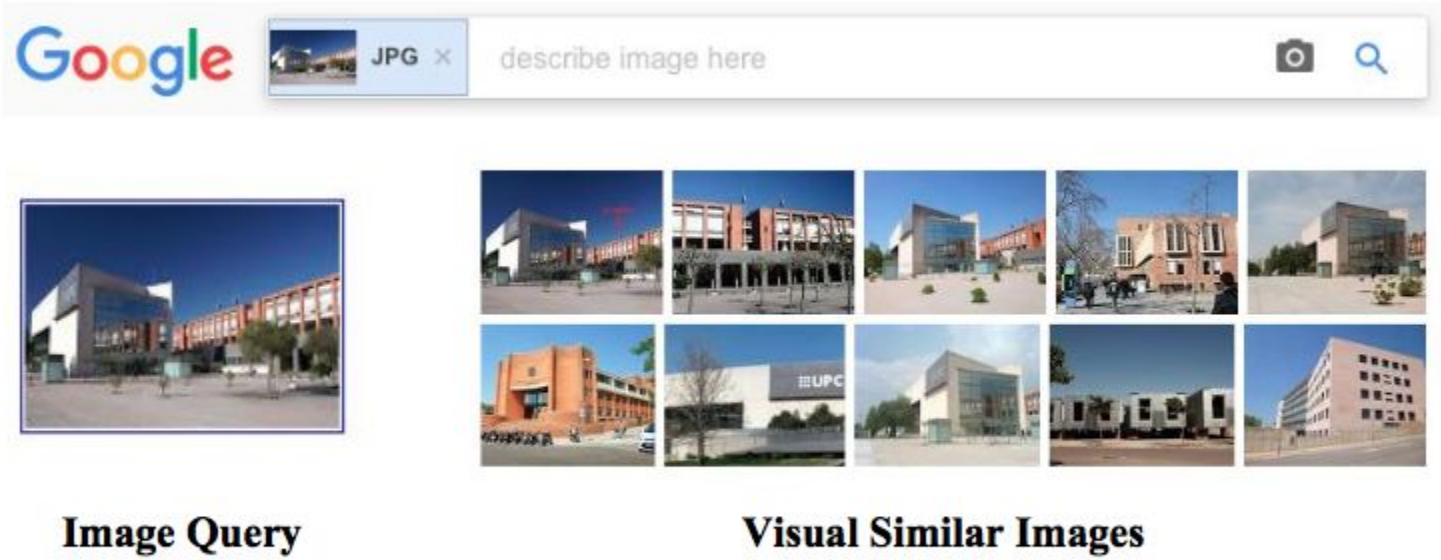
DATA 61 CSIRO

Image Retrieval

Google Search By Image

Image Retrieval

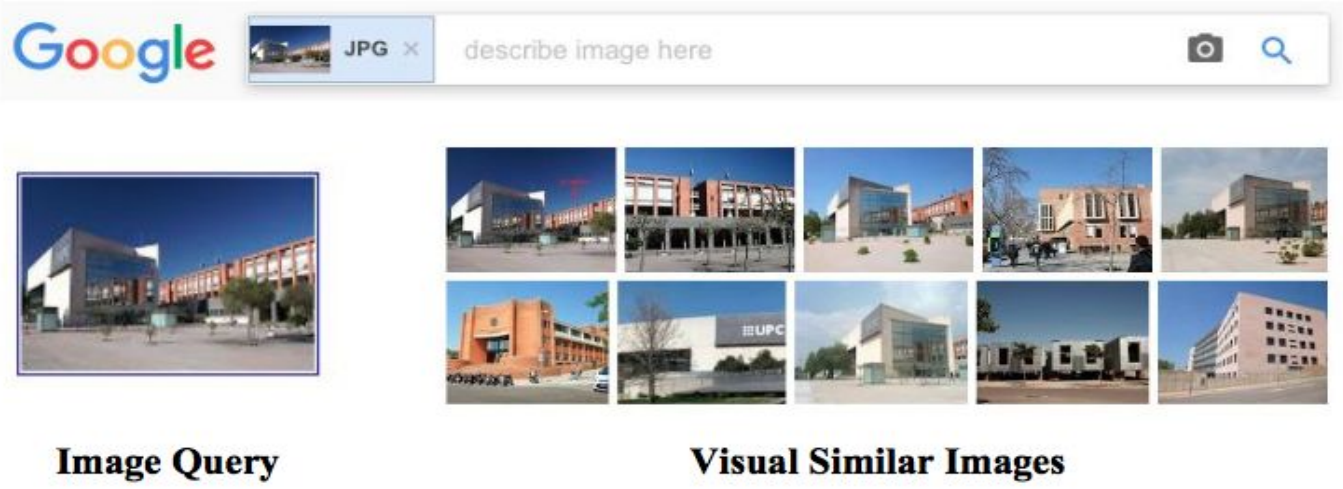
Task Definition



Given an image query, generate a ranked list of similar images

Image Retrieval

Task Definition



Given an image query, generate a ranked list of similar images

What are the properties that we want for these systems?

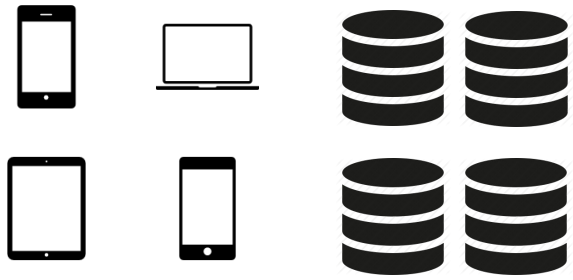
Image Retrieval

Desirable System Properties

Invariant to **scale**, **illumination**, **translation**



Memory Efficient



Be able to provide a **fast search**

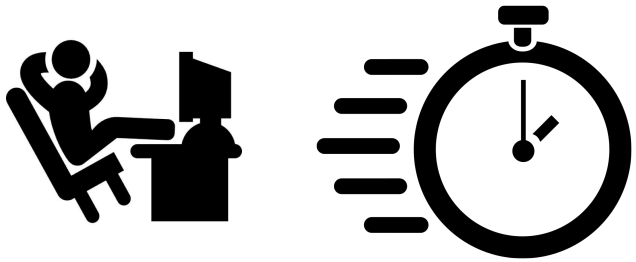


Image Retrieval

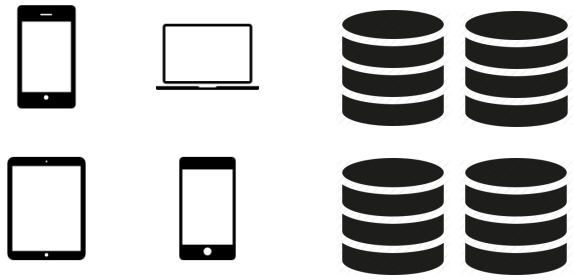
Desirable System Properties

Invariant to **scale**, **illumination**, **translation**

How to achieve them?



Memory Efficient



Be able to provide a **fast search**

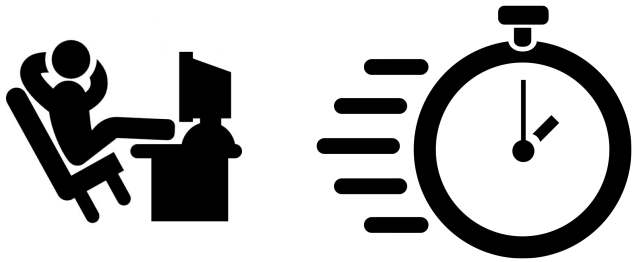
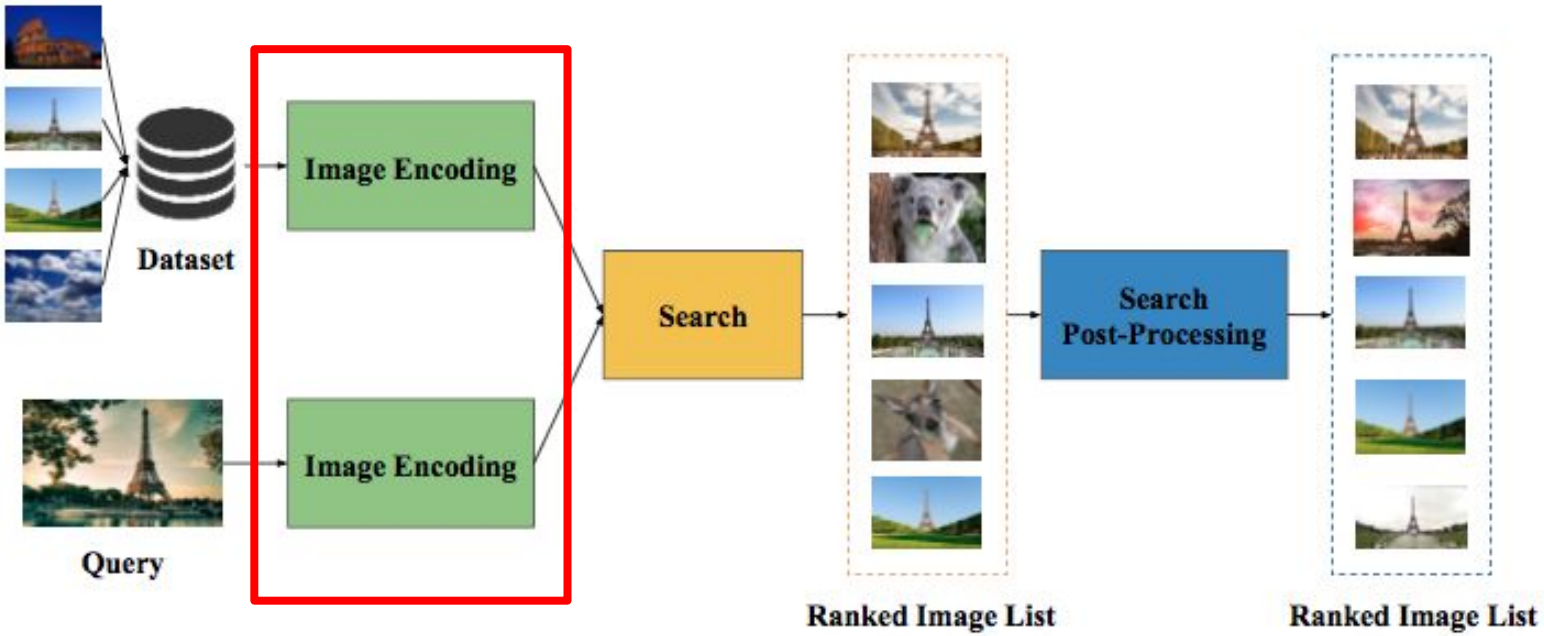


Image Retrieval

General Pipeline

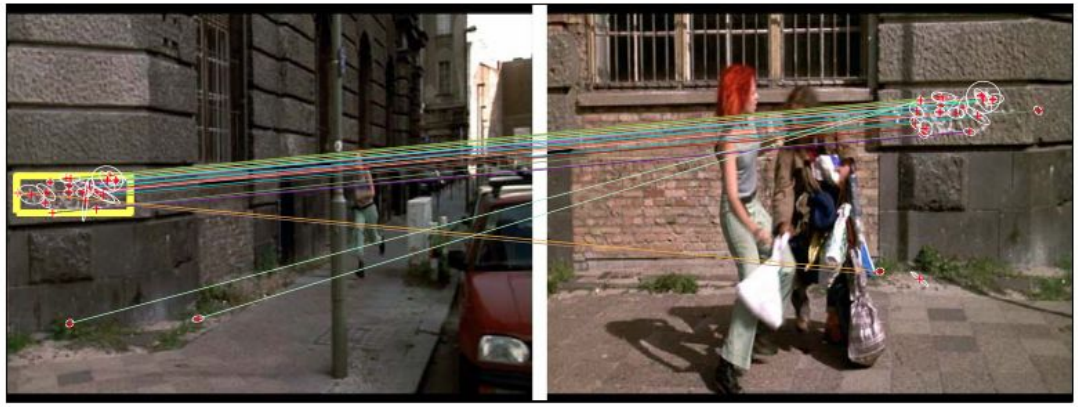


- Encode images into representations that are
  - Invariant
  - Compact

Image Retrieval

Image Representations

- Hand-Crafted Features (Before 2012)
  - SIFT
  - SIFT + BoW
  - VLAD
  - Fisher Vectors

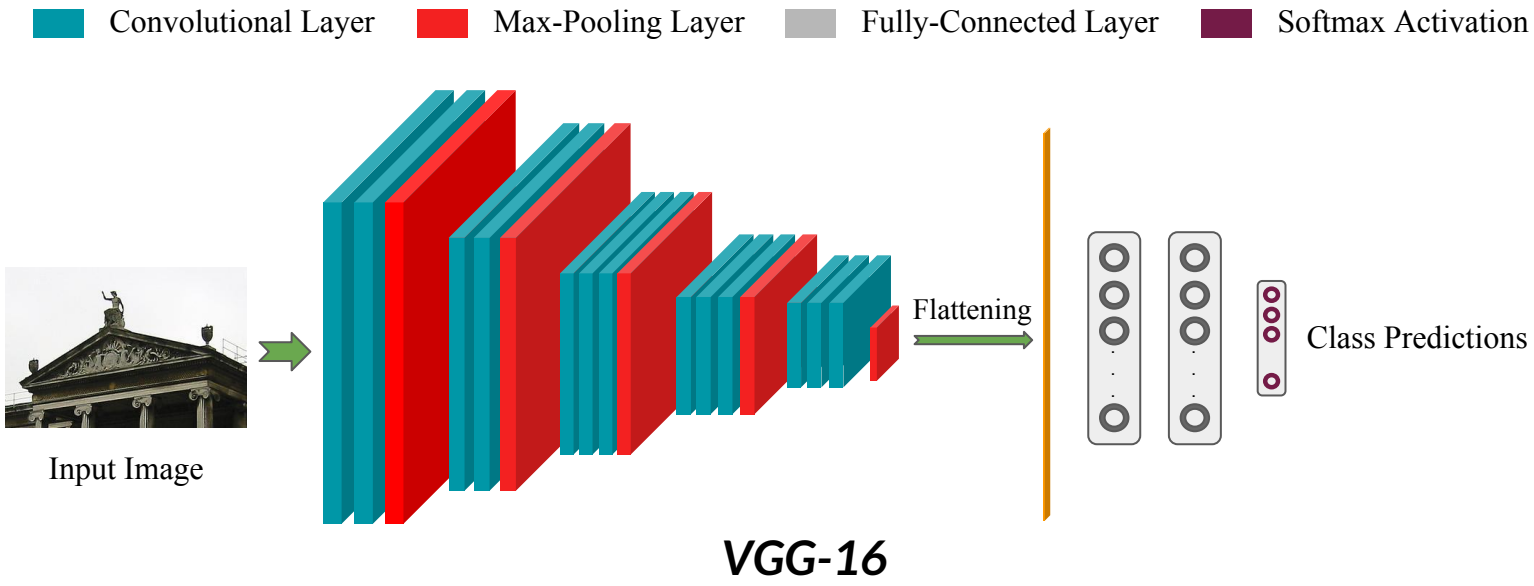


Josef Sivic, Andrew Zisserman, et al. Video google: A text retrieval approach to object matching in videos. In ICCV, 2003.

Image Retrieval

Image Representations

- Learned Features (After 2012)
  - **Convolutional Neural Networks (CNNs)**



Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. (2014)

Image Retrieval

Image Representations

- Learned Features (After 2012)
  - **Convolutional Neural Networks (CNNs)**

Training is expensive!

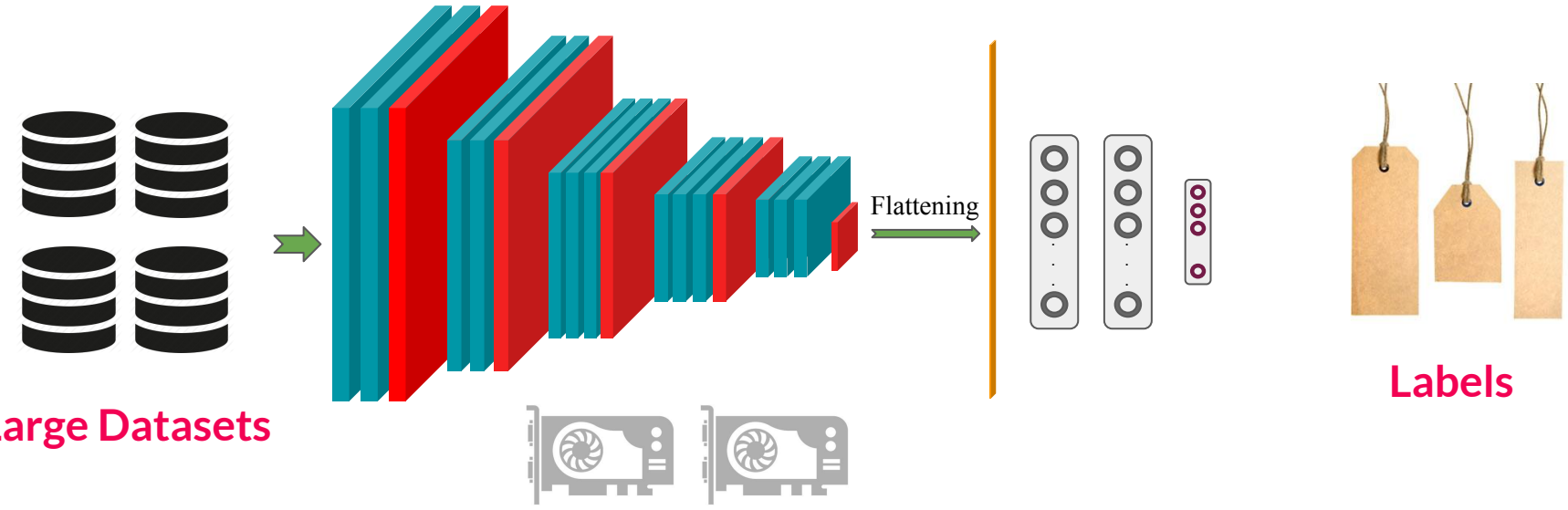
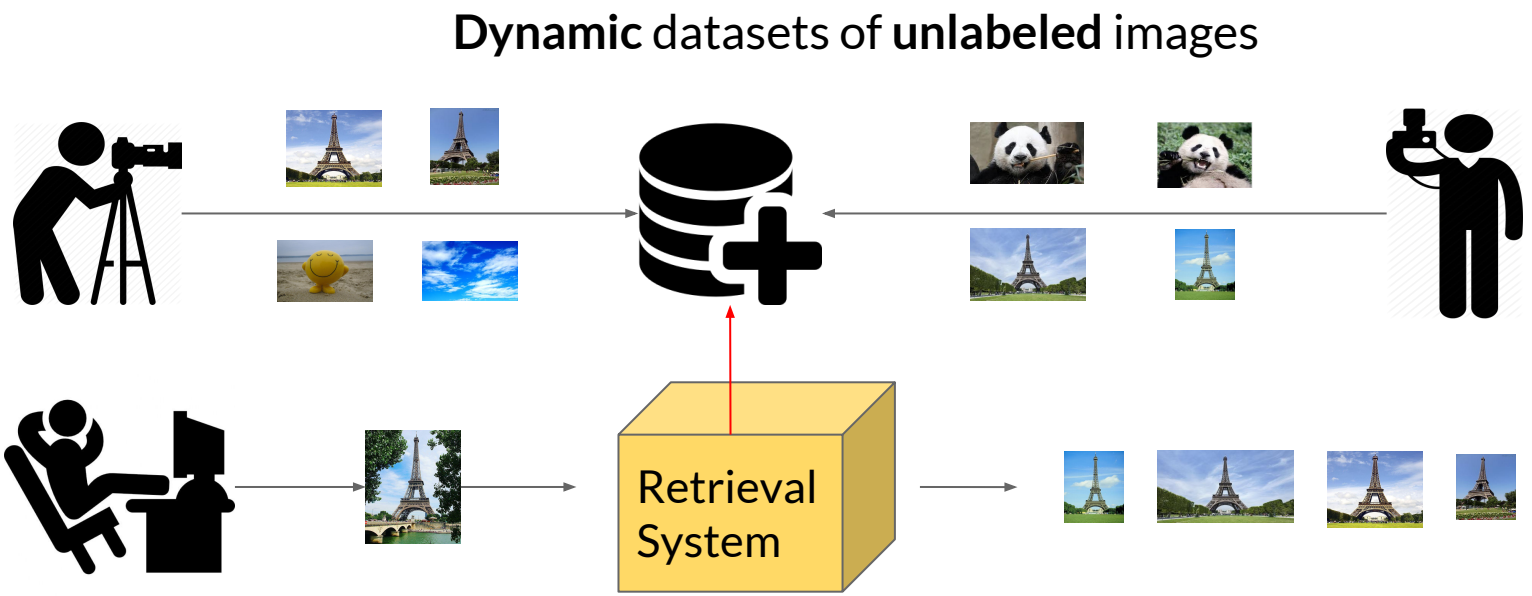


Image Representations

Challenges for Learning Features in Retrieval



➔ Collecting, annotating and cleaning a large dataset to train models requires great effort.

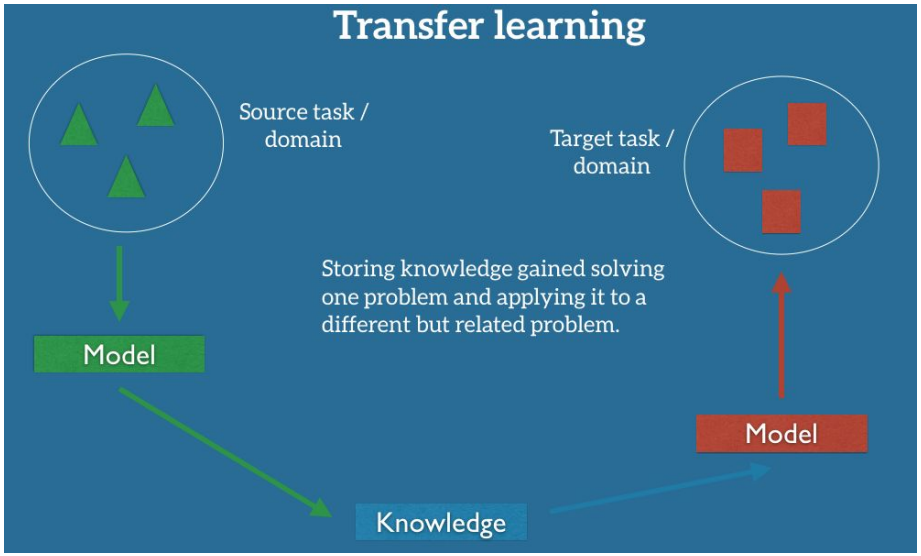
➔ Training or fine-tuning a model every time new data is added is neither efficient nor scalable.

Image Representations

Challenges for Learning Features

One solution that does not involve training is **Transfer Learning**

➔ Transferring the knowledge of a model trained in a large-scale dataset. (e.g. ImageNet)



Classification → Retrieval

How?

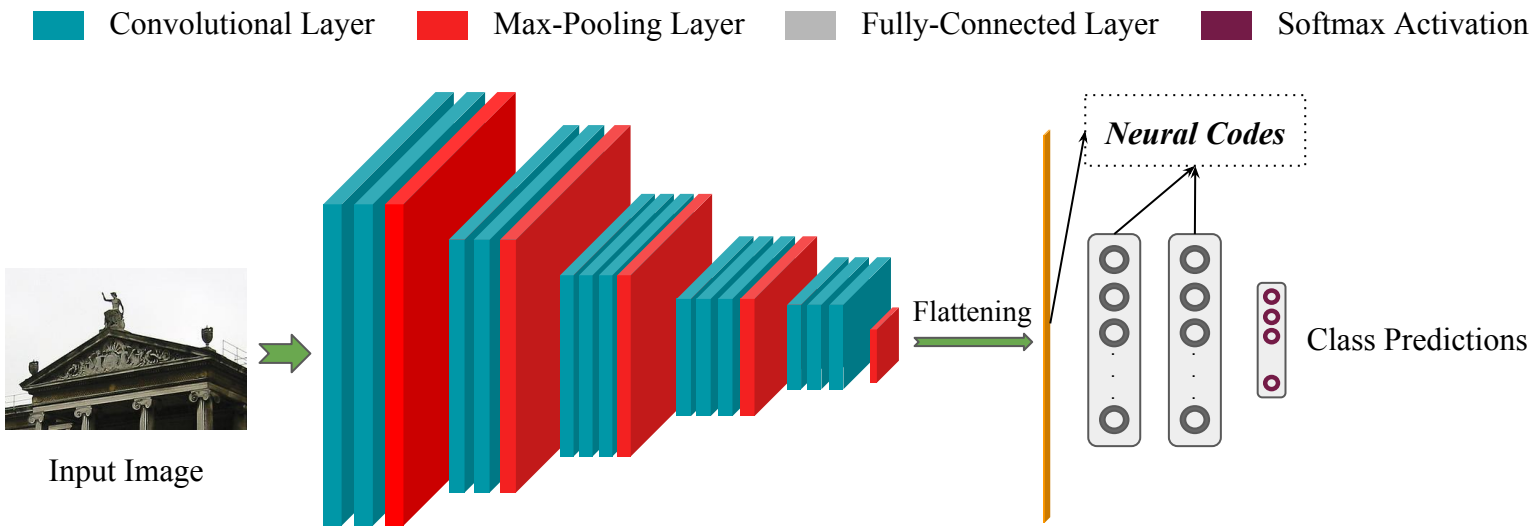
Outline

- Related Work
- Proposal
- Experiments
- Conclusions

Related Work

Image Representations

Fully-Connected Features

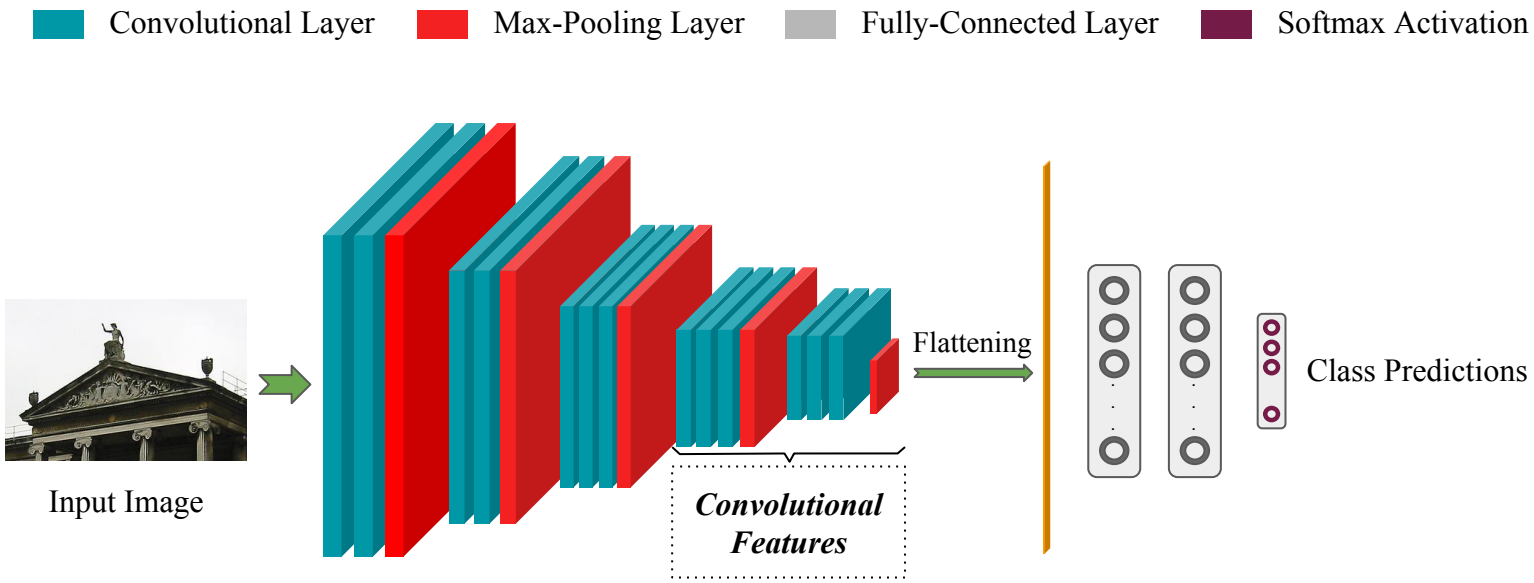


Fully-connected features as global representation

Babenko, A., Slesarev, A., Chigorin, A., & Lempitsky, V. (2014). Neural codes for image retrieval. In *ECCV 2014*
Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In DeepVision *CVPRW 2014*

Image Representations

Convolutional Features



Convolutional features as global representation

➔ They convey the **spatial information** of the image

Babenko, A., & Lempitsky, V. (2015). Aggregating local deep features for image retrieval. *ICCV 2015*

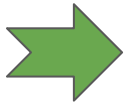
Tolias, G., Sicre, R., & Jégou, H. (2016). Particular object retrieval with integral max-pooling of CNN activations. *ICLR 2016*

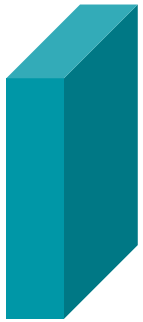
Kalantidis, Y., Mellina, C., & Osindero, S. (2015). Cross-dimensional Weighting for Aggregated Deep Convolutional Features. *arXiv preprint arXiv:1512.04065*.

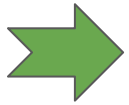
Image Representations

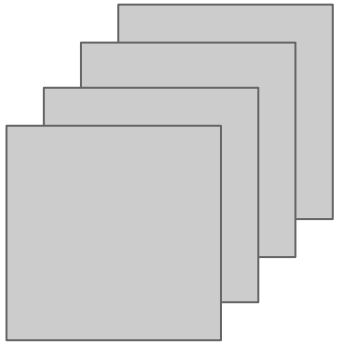
From Feature Maps to Compact Representations

Image | **Convolutional layer** | **Feature Maps**









$I_h \times I_w \times 3$

$(F_h \times F_w \times 3) \times \mathbf{K}$ Filters

Stride: S, Padding: P

$(H \times W) \times \mathbf{K}$ Feature Maps

$H = (I_h - F_h + 2P) / S + 1$

$W = (I_w - F_w + 2P) / S + 1$

Image Representations

From Feature Maps to Compact Representations

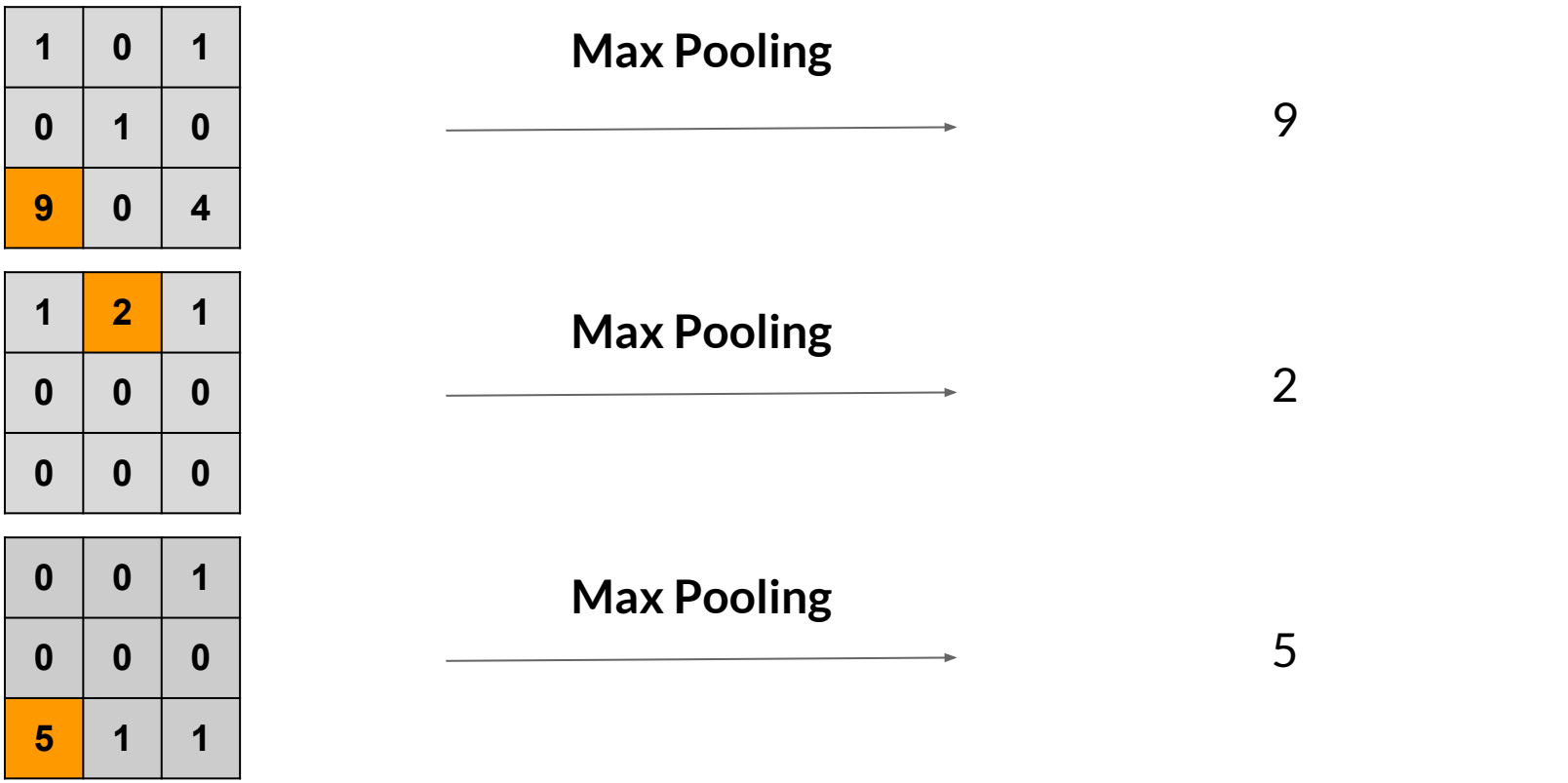


Image Representations

From Feature Maps to Compact Representations

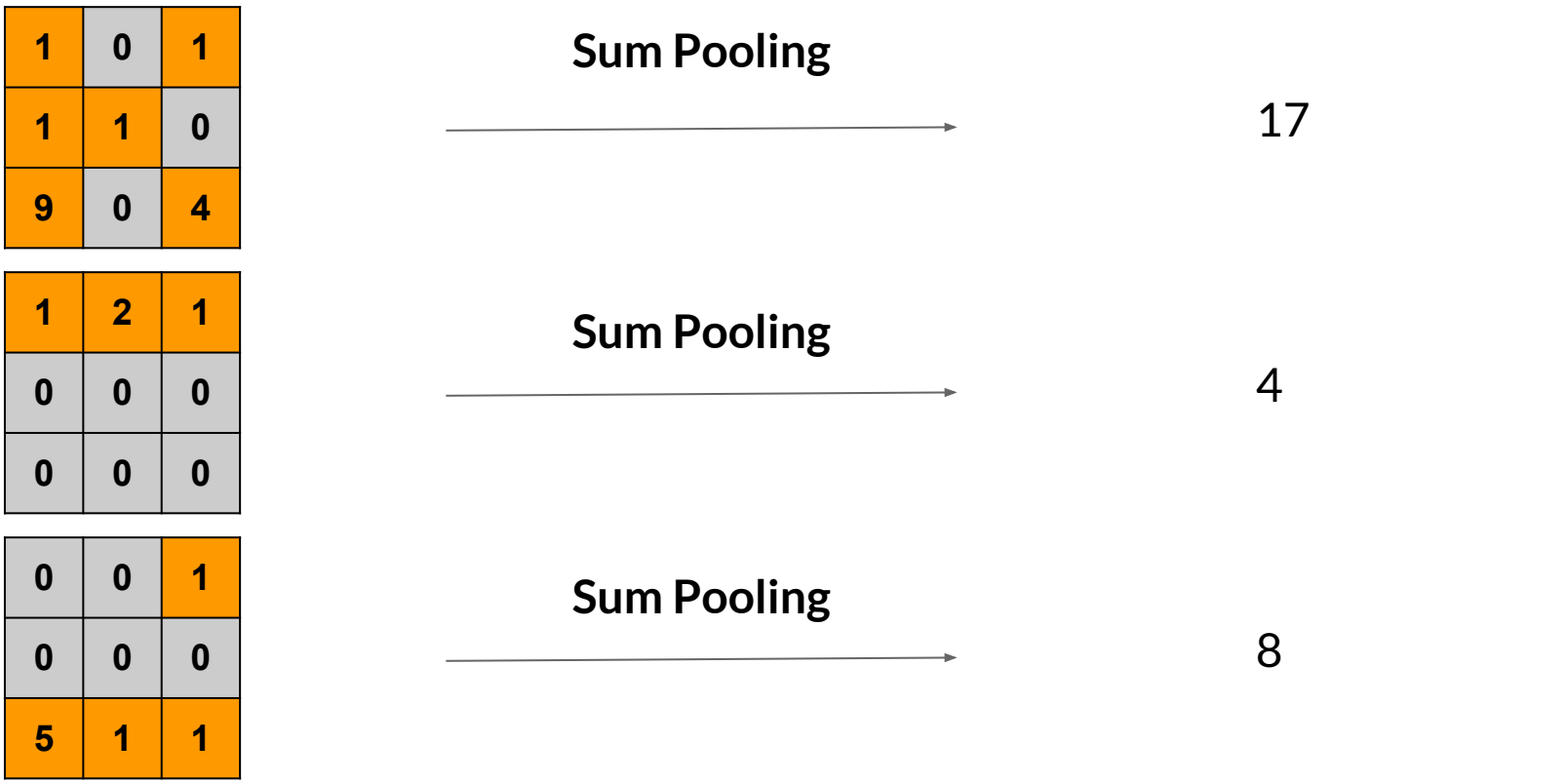
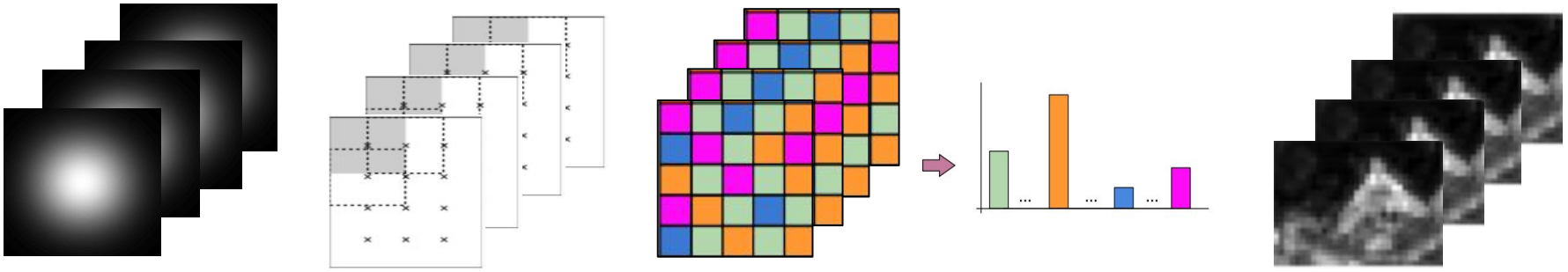


Image Representations

Related Works

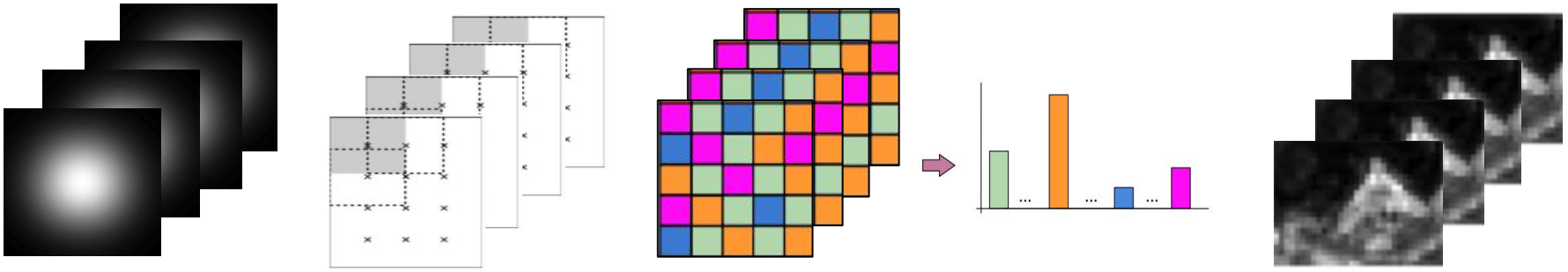


Combine and aggregate convolutional features:

- **SPoC** → Gaussian prior + Sum Pooling
- **R-MAC** → Rigid grid of regions + Max Pooling
- **BoW** → Bag of Visual Words
- **CroW** → Spatial weighting based on normalized sum of feature maps (boost salient content) + channel weighting based on sparsity + Sum Pooling

Image Representations

Related Works



All except CroW:

- Apply fixed weights
- Based on fixed regions

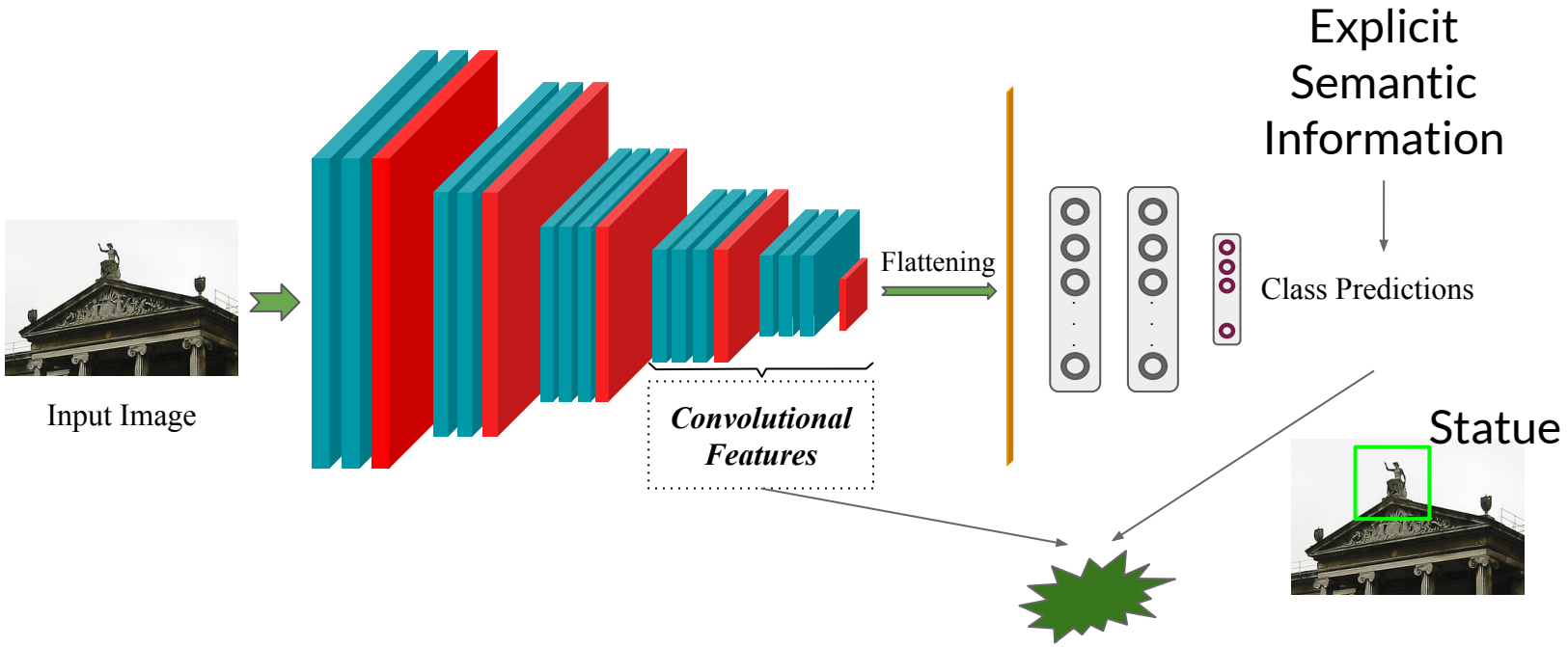
Not based on image contents

Proposal

Motivation

Image Semantics

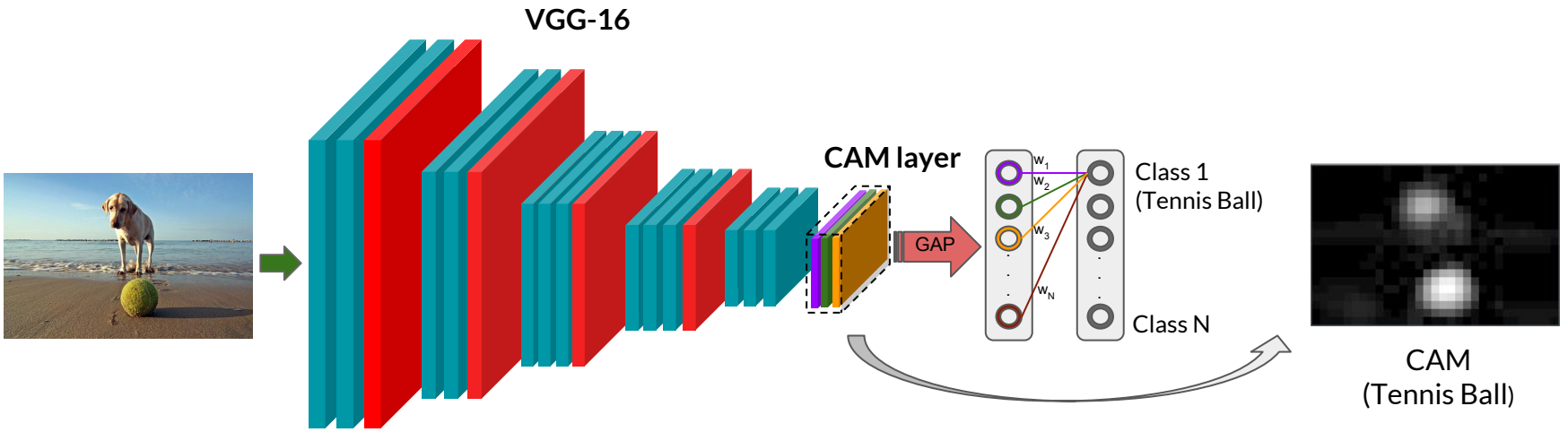
Objective



Combine semantic content of the images with convolutional features

Class Activation Maps

Obtaining Discriminative Regions

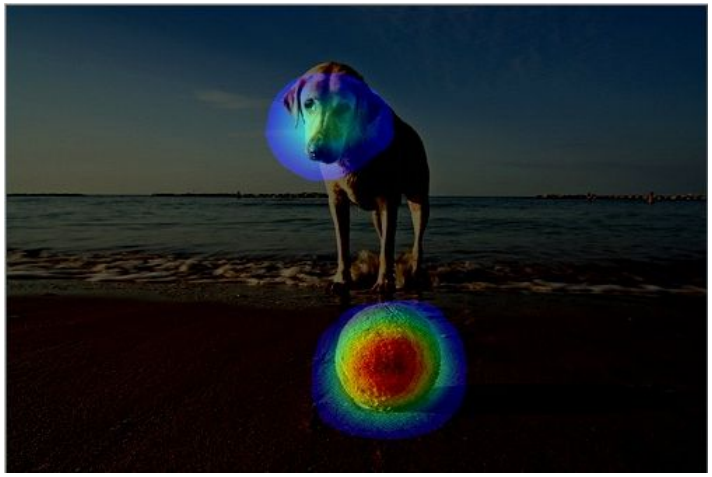


$$CAM_i = w_{1,i} * conv_1 + w_{2,i} * conv_2 + \ldots + w_{N,i} * conv_N$$

B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. 2016. Learning Deep Features for Discriminative Localization. CVPR (2016).

Class Activation Maps

Examples



Tennis Ball

Weimaraner

Simple Classes

Class Activation Maps

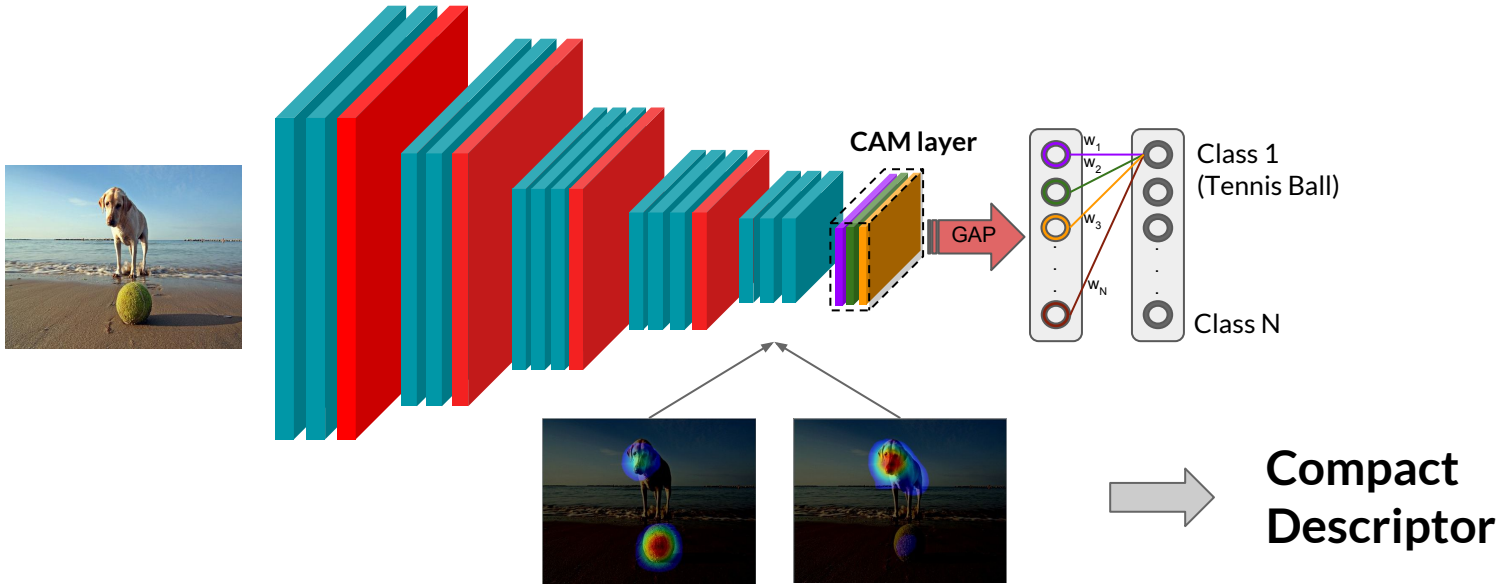
Examples

Sand Bar

Seashore

Complex Classes

Proposal



Class-Weighted Convolutional Features

Encode images dynamically combining **their semantics** with **convolutional features** using only the **knowledge contained** inside a **trained network**

Image Encoding Pipeline

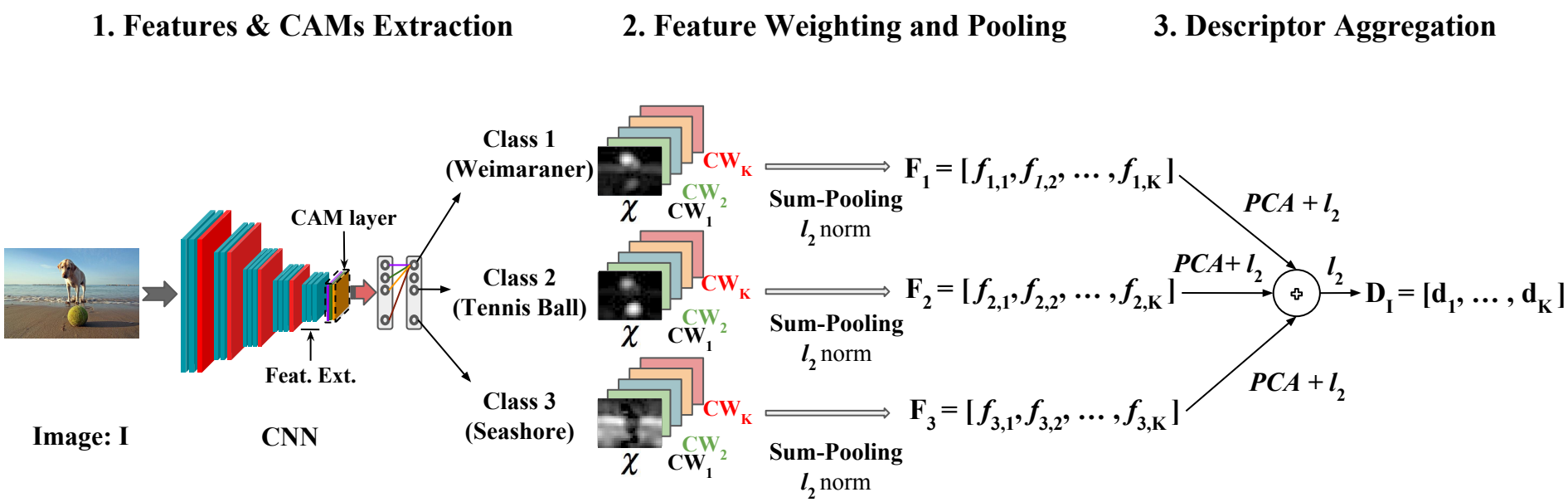
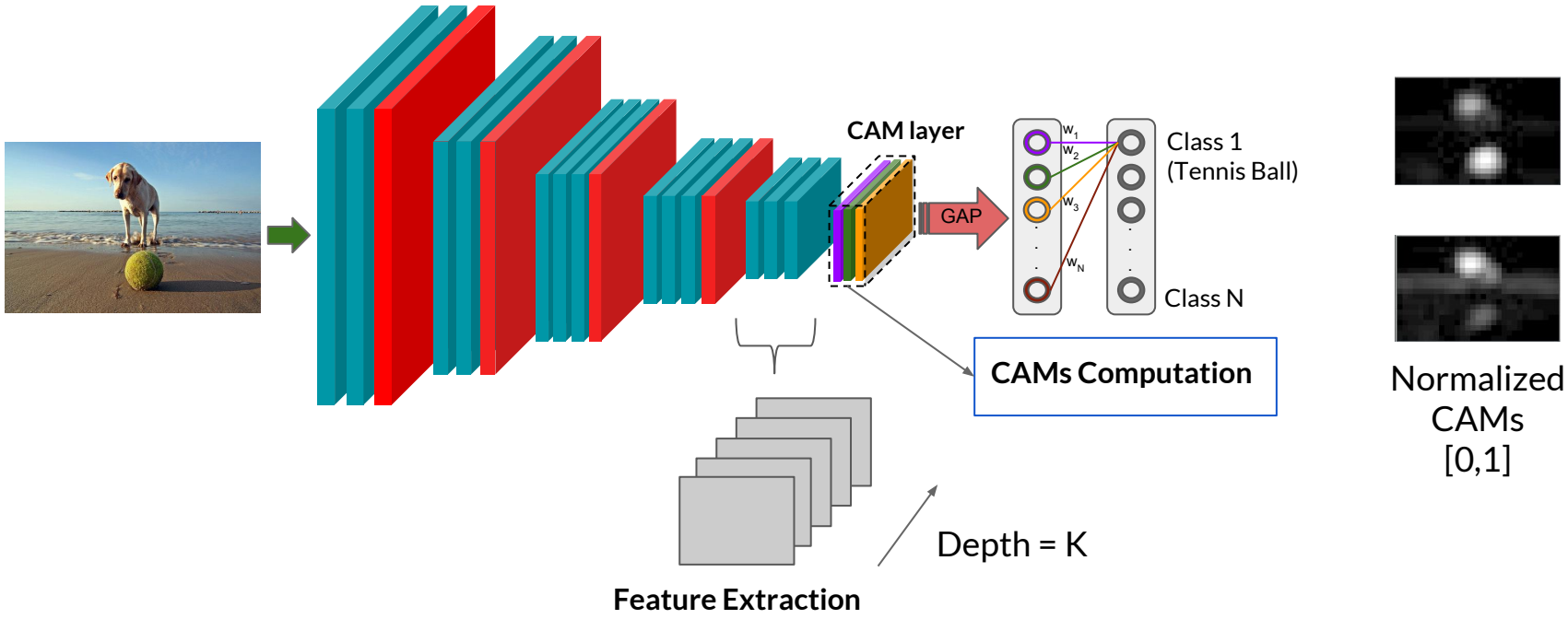


Image Encoding Pipeline

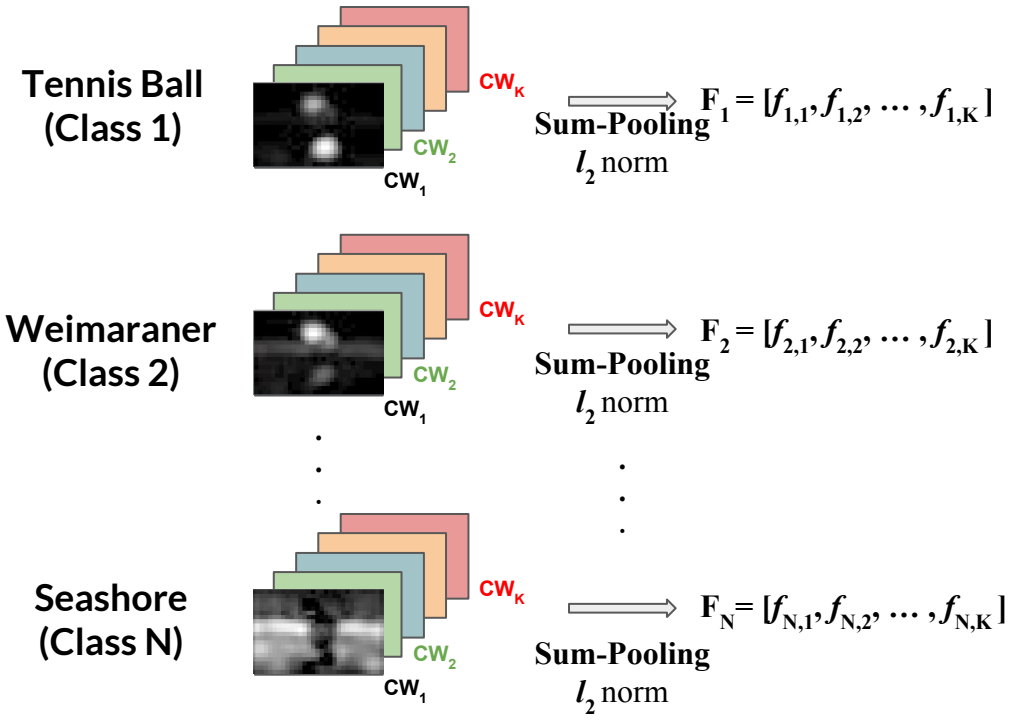
1. Features & CAMs Extraction



In a **single forward pass** we extract convolutional features and image CAMs

Image Encoding Pipeline

2. Feature Weighting and Pooling



One vector per class:

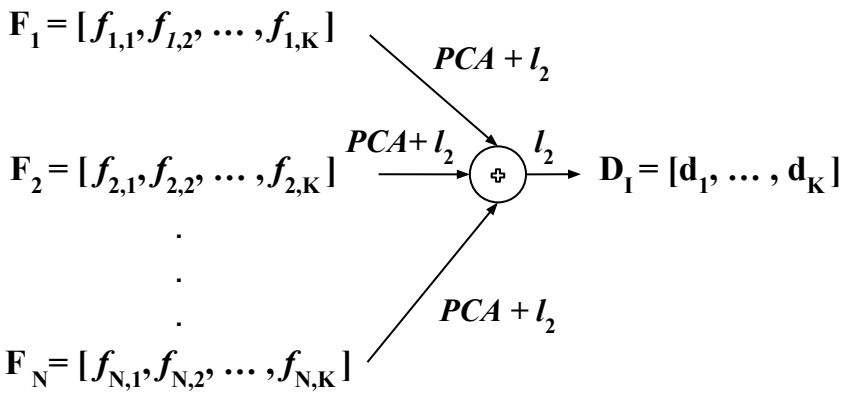
$$F^c = [f_1^c, f_2^c, \ldots, f_K^c]$$

$$f_k^{(c)} = CW_k \sum_{i=1}^{W} \sum_{j=1}^{H} \chi_{i,j}^{(k)} \, CAM_{i,j}^{(c)}$$

Spatial Weighting using Class Activation Maps
Channel Weighting based on feature maps sparsity (as in CroW)

Image Encoding Pipeline

3. Descriptor Aggregation



➔ Which classes do we aggregate?

➔ What is the optimal number of classes (N_C) ?

➔ What is the optimal number of classes (N_{PCA}) to compute the PCA ?

Descriptor Aggregation Strategies

Online Aggregation (OnA)

Image Dataset

Image Encoding Pipeline

Store all the class vectors (F_c)

Query

Image Encoding Pipeline

Query Descriptor

Dataset Descriptor Aggregation

Scores

CNN Most Probable Classes Prediction from query (Retriever, Seashore...)

Descriptor Aggregation Strategies

Offline Aggregation (OfA)

Image Dataset

Image Encoding Pipeline

Dataset Descriptor Aggregation

Store final descriptor

CNN Most Probable Classes Prediction

Query

Image Encoding Pipeline

Query Descriptor

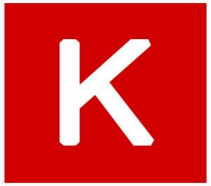
Pre-Computed Descriptors

Scores

Experiments

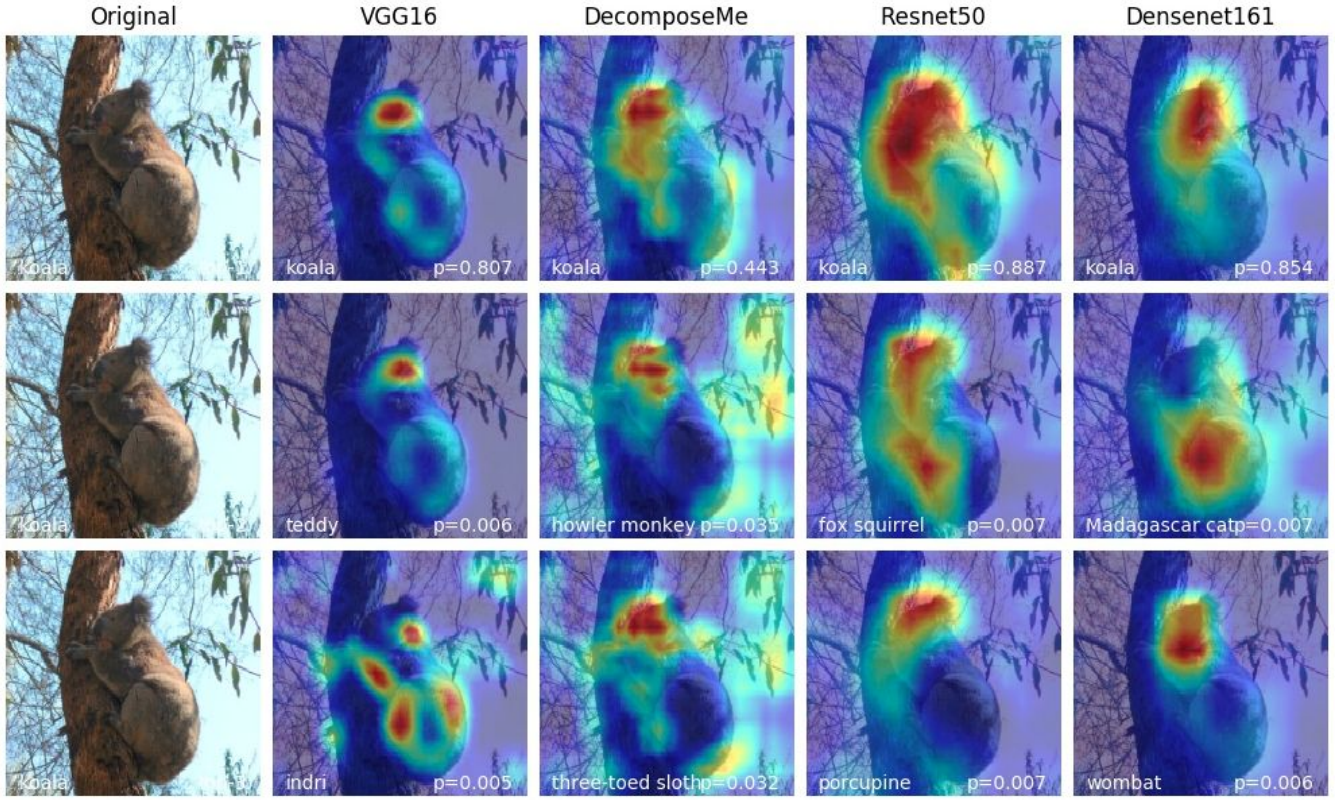
Experimental Setup

- Framework: Keras with Theano as backend / PyTorch
- Images resized to 1024x720 (keeping aspect ratio)
- We explored using different CNNs as feature extractors



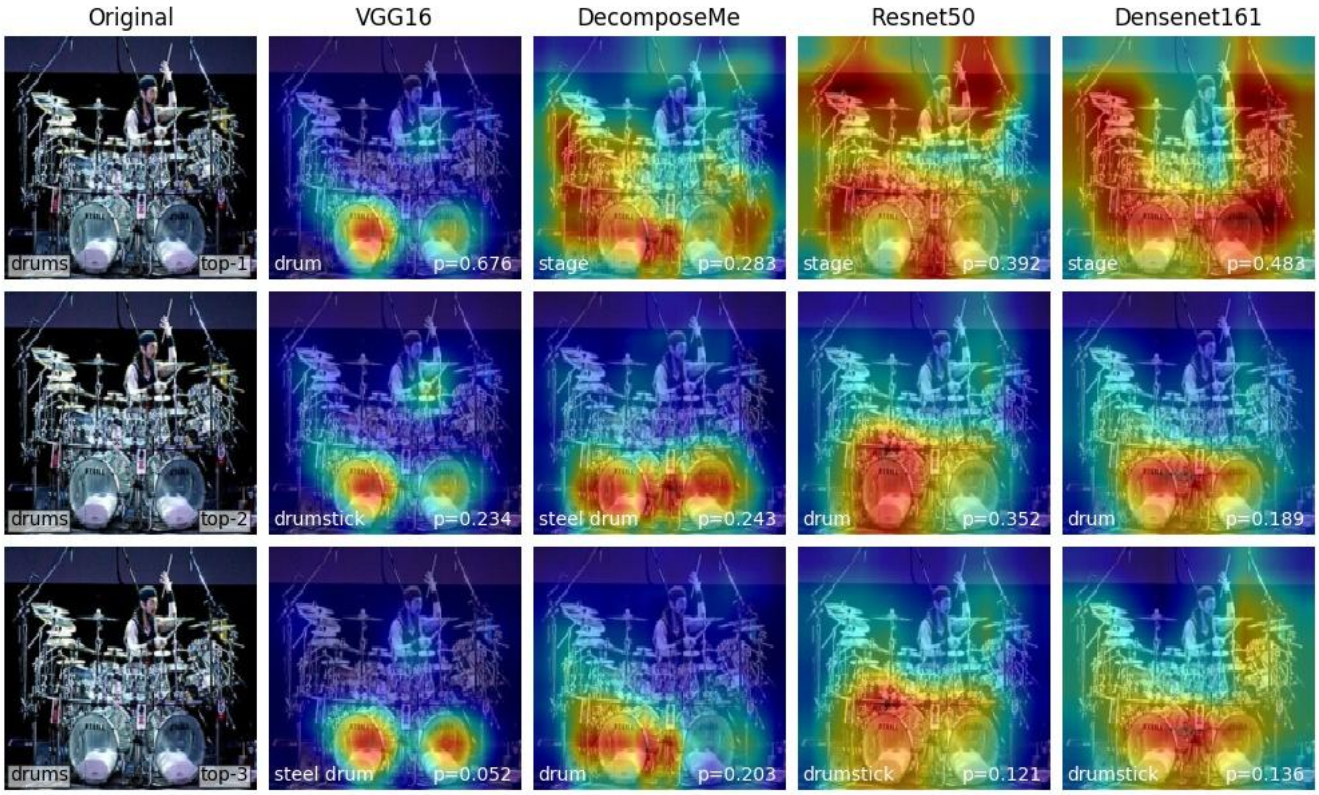
Experimental Setup

Examples of CAMs from different networks



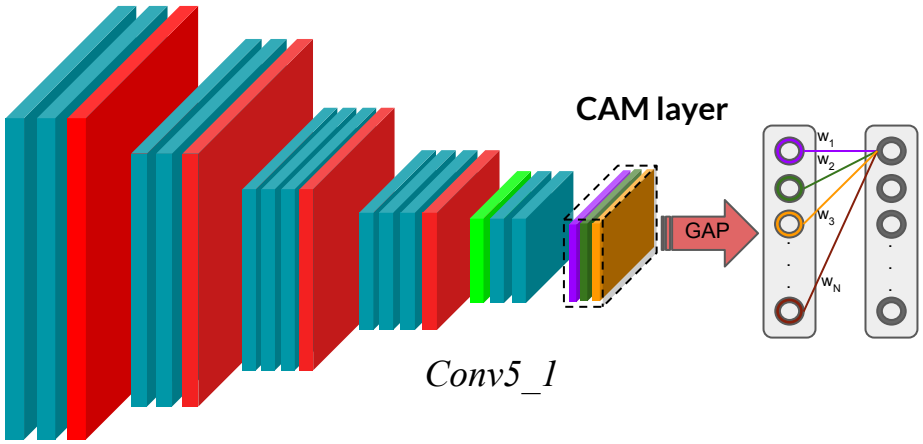
Experimental Setup

Examples of CAMs from different networks



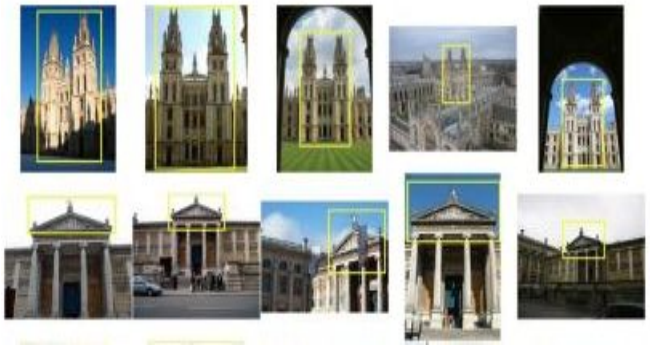
Experimental Setup

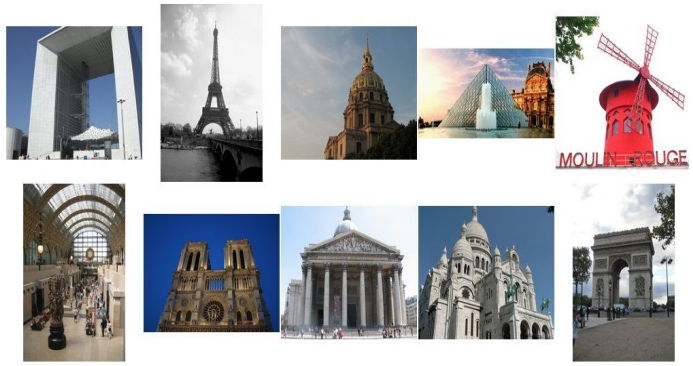
- VGG-16 CAM model as feature extractor (pre-trained on ImageNet)
- Features from *Conv5_1* Layer



Experimental Setup

- Datasets
  - Oxford 5k
  - Paris 6k
  - 100k Distractors (Flickr) → Oxford105k, Paris106k
- Using the features that fall inside the cropped query
- PCA computed with Oxf5k when testing in Par6k and vice versa





Philbin, J. , Chum, O. , Isard, M. , Sivic, J. and Zisserman, A. Object retrieval with large vocabularies and fast spatial matching, *CVPR 2007*
Philbin, J. , Chum, O. , Isard, M. , Sivic, J. and Zisserman, A. Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases. *CVPR 2008*

Experimental Setup

- Scores computed with cosine similarity

$$(||X||_2 =, \; ||Y||_2 = 1)$$

$$Cosine\ similarity\ (X, Y) = \frac{X \cdot Y}{||X||_2 ||Y||_2} = X \cdot Y$$

- Evaluation metric: mean Average Precision (mAP)

Ablation Studies

Finding the optimal parameters

- We carried out ablation studies to validate our method
- Computational Burden
- Number of top-classes to add: N_C
- Number of top-classes to compute the PCA matrix: N_{PCA}

Ablation Studies

Baseline Results & Computational Burden

Descriptor Aggregation	Oxford5k	Paris6k
Raw Features	0.396	0.526
Raw + Crow (channel)	0.420	0.549
Raw Features + PCA	0.589	0.662
Raw + Crow(channel) + PCA	0.607	0.685

Descriptor Aggregation	Time (s)	mAP
Raw + PCA	0.49	0.589
1 CAM	0.5	0.667
8 CAMs	0.6	0.709
32 CAMs	0.9	0.711
64 CAMs	1.5	0.712

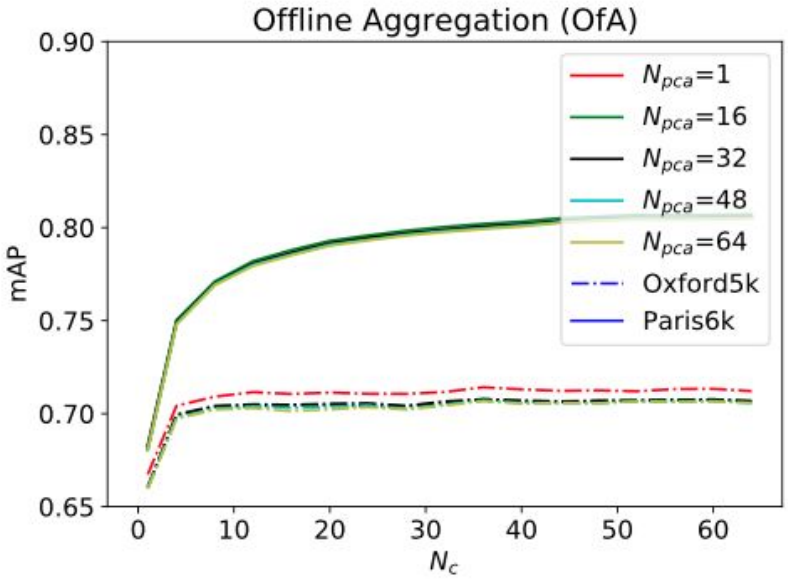
Ablation Studies

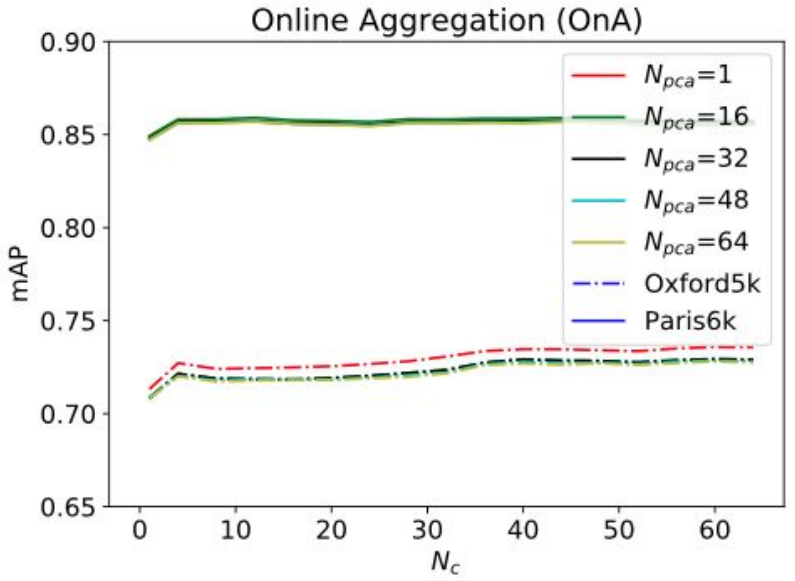
Networks Comparison

Network	Oxf5k	Paris6k
VGG-16 (Raw)	0.396	0.526
VGG-16 (64CAMs)	0.712	0.805
Resnet-50 (Raw)	0.389	0.508
Resnet-50 (64CAMs)	0.699	0.804
Densenet-161 (Raw)	0.339	0.495
Densenet-161 (64CAMs)	0.695	0.799

Ablation Studies

Sensitivity with respect to N_C and N_{pca}



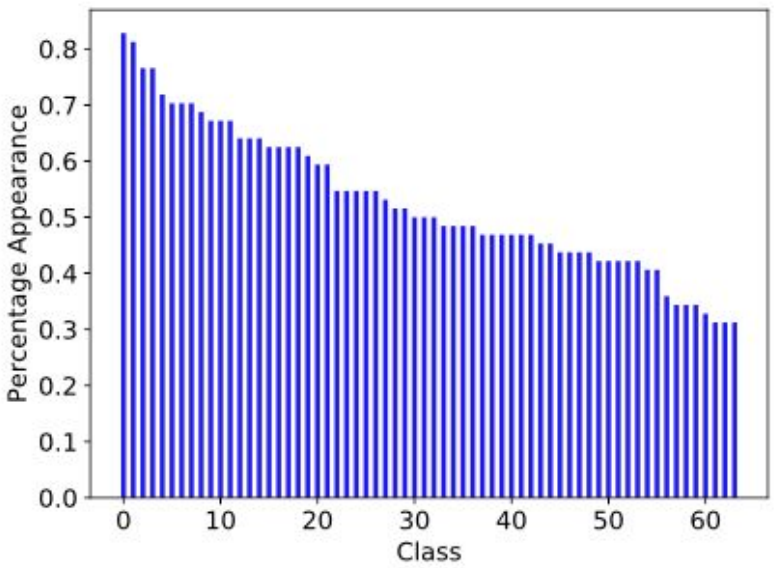


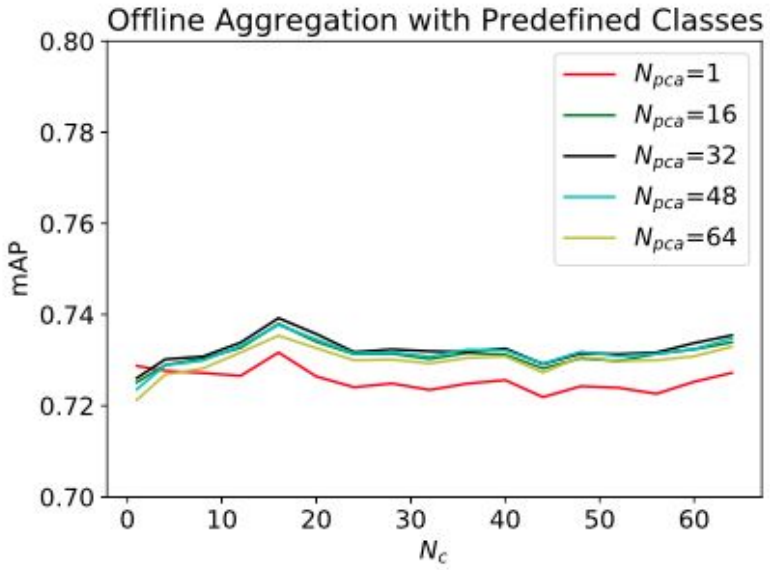
N_c corresponds to the number of classes aggregated.

N_{pca} corresponds to the classes used to compute the PCA (computed with data of Oxford when testing in Paris and vice versa).

Ablation Studies

Using predefined classes





The first 16 classes correspond to: *vault*, *bell cote*, *bannister*, *analog clock*, *movie theater*, *coil*, *pier*, *dome*, *pedestal*, *flagpole*, *church*, *chime*, *suspension bridge*, *birdhouse*, *sundial*, *triumphal arch*. All of them are related to **buildings**.

Comparison with State-of-the-Art

Method	Dim	Oxf5k	Par6k	Oxf105k	Par106k
SPoC[3]	256	0.531	-	0.501	-
uCroW[14]	256	0.666	0.767	0.629	0,695
CroW[14]	512	0.682	0.796	0.632	0.710
R-MAC[31]	512	0.669	0.830	0.616	**0.757**
BoW[16]	25k	0.738	0.820	0.593	0.648
Razavian [22]	32k	**0.843**	0.853	-	-
Ours(OnA)	512	0.736	**0.855**	-	-
Ours(OfA)	512	0.712	0.805	**0.672**	0.733

$N_c = 64$, $N_{pca} = 1$

Search Post-Processing

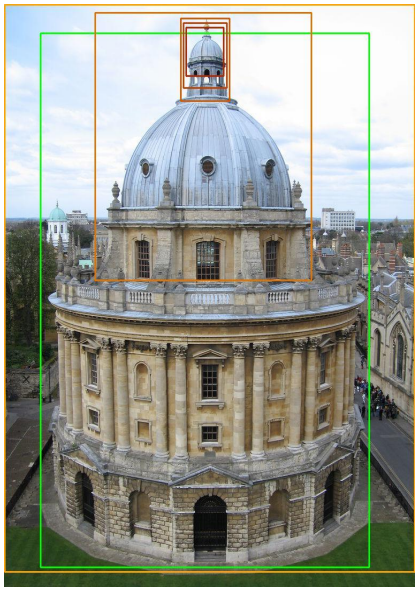
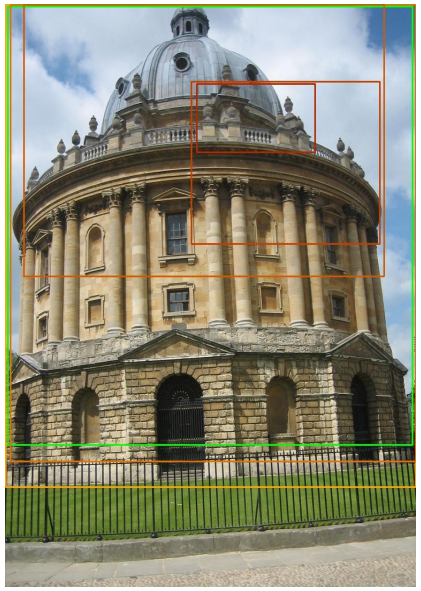
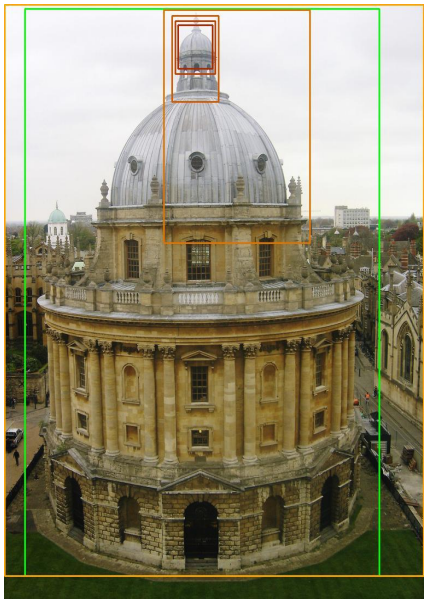
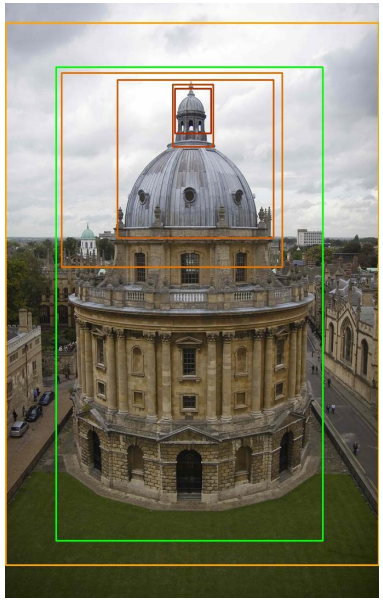
Region Proposals for Re-Ranking

- We establish a set of thresholds based on the max value of the normalized CAM
  - 1%, 10%, 20%, 30%, 40%
- Generate a bounding box covering the largest connected element
- Compare the descriptor of each new region of the target image with the query descriptor and generate new scores.
- Results a bit better if we average over 2 most probable CAMs
- For the top-R target images

Query Expansion

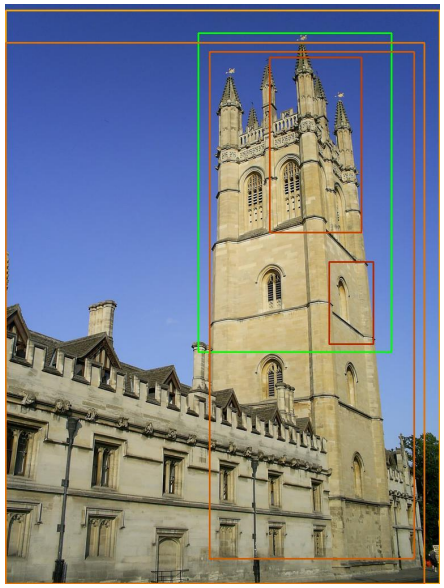
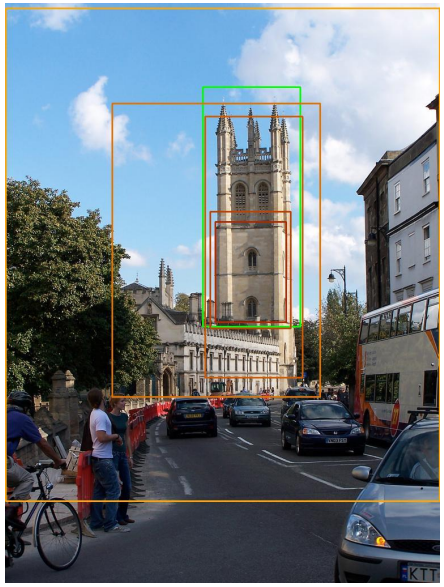
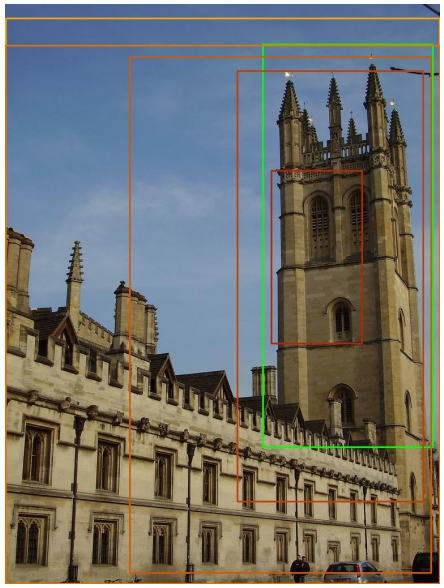
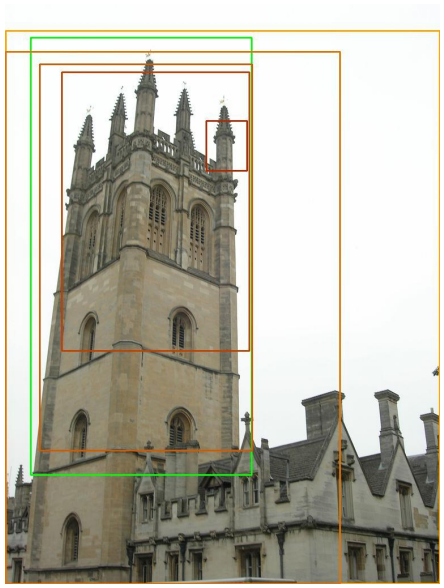
- Generate a new query by l_2 normalizing the sum of the top-QE target descriptors

Region Proposals using CAMs

Green - Ground Truth
Orange → Red - Different Thresholds

Region Proposals using CAMs

Green - Ground Truth
Orange → Red - Different Thresholds

Comparison with State-of-the-Art

After Re-Ranking and Query Expansion

Method	Dim	R	QE	Oxf5k	Par6k	Oxf105k	Par106k
CroW	512	-	10	0.722	0.855	0.678	0.797
Ours(OnA)	512	-	10	0.760	0.873	-	-
Ours(OfA)	512	-	10	0.730	0.836	0.712	0.791
BoW	25k	100	10	0.788	0.848	0.651	0.641
Ours(OnA)	512	100	10	0.780	0.874	-	-
Ours(OfA)	512	100	10	0.773	0.838	0.750	0.780
RMAC	512	1000	5	0.770	**0.877**	0.726	**0.817**
Ours(OnA)	512	1000	5	**0.811**	0.874	-	-
Ours(OfA)	512	1000	5	0.801	0.855	**0.769**	0.800

$N_c = 64$, $N_{pca} = 1$, $N_{re\text{-}ranking} = 6$

Conclusions

Conclusions

- In this work we proposed a technique to build compact image representations focusing on their semantic content.
- We employed an image encoding pipeline that makes use of a pre-trained CNN and Class Activation Maps to extract discriminative regions from the image and weight its convolutional features accordingly
- Our experiments demonstrated that selecting the relevant content of an image to build the image descriptor is beneficial, and contributes to increase the retrieval performance. The proposed approach establishes a new state-of-the-art compared to methods that build image representations combining off-the-shelf features using random or fixed grid regions.


Thank
you