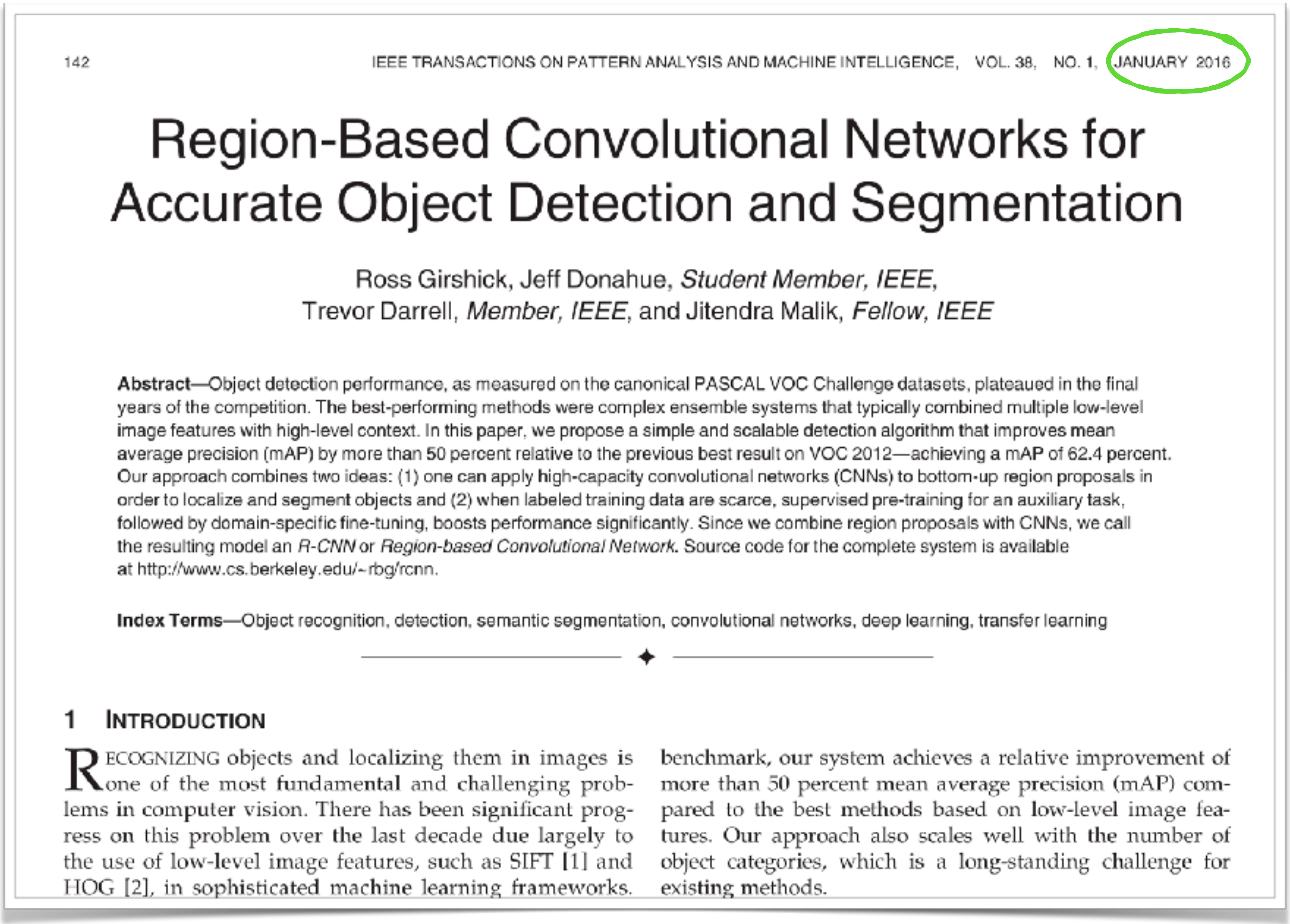
RCNN



Region-Based Convolutional Networks for Accurate Object Detection and Segmentation

Ross Girshick, Jeff Donahue, *Student Member, IEEE*,
Trevor Darrell, *Member, IEEE*, and Jitendra Malik, *Fellow, IEEE*

Abstract—Object detection performance, as measured on the canonical PASCAL VOC Challenge datasets, plateaued in the final years of the competition. The best-performing methods were complex ensemble systems that typically combined multiple low-level image features with high-level context. In this paper, we propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 50 percent relative to the previous best result on VOC 2012—achieving a mAP of 62.4 percent. Our approach combines two ideas: (1) one can apply high-capacity convolutional networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data are scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, boosts performance significantly. Since we combine region proposals with CNNs, we call the resulting model an *R-CNN* or *Region-based Convolutional Network*. Source code for the complete system is available at http://www.cs.berkeley.edu/~rbg/rcnn.

Index Terms—Object recognition, detection, semantic segmentation, convolutional networks, deep learning, transfer learning

1 Introduction

Recognizing objects and localizing them in images is one of the most fundamental and challenging problems in computer vision. There has been significant progress on this problem over the last decade due largely to the use of low-level image features, such as SIFT [1] and HOG [2], in sophisticated machine learning frameworks. benchmark, our system achieves a relative improvement of more than 50 percent mean average precision (mAP) compared to the best methods based on low-level image features. Our approach also scales well with the number of object categories, which is a long-standing challenge for existing methods.

Nov. 2013

Visual Object Classes Challenge 2010 (VOC2010)

PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

(click on an image to see the annotation)

For news and updates, see the PASCAL Visual Object Classes Homepage

News

- **09-Mar-11**: Test data can now be downloaded from the PASCAL VOC Evaluation Server. You can also use the evaluation server to evaluate your method on the test data.
- 25-Oct-10: Detailed results of all submitted methods are now online. For summarized results and information about selected methods, please see the workshop presentations.
- 03-Sep-10: Provisional programme for the workshop is now online.
- 31-Aug-10: Submission of results is now closed. We are aiming to release results to participants only on 1st September.
- 09-Aug-10: The evaluation server is now online and results may be submitted.
- 09-Aug-10: The deadline for submission of results is extended to 30th August, 23:00 GMT. There will be no further extensions.
- 24-May-10: The test data is now available.
- 08-May-10: The development kit including training/validation data is now available.
- 03-May-10: A new taster competition on (still image) action classification has been introduced.
- 08-Apr-10: A new taster competition on Large Scale Visual Recognition has been introduced in cooperation with ImageNet.
- 01-Mar-10: We are preparing to run the VOC2010 challenge. The provisional timetable is below.
- 01-Mar-10: The challenge workshop will be held in conjunction with ECCV 2010, 11th September 2010, Crete.

Classification/Detection Competitions

1. **Classification**: For each of the twenty classes, predicting presence/absence of an example of that class in the test image.
2. **Detection**: Predicting the bounding box and label of each object from the twenty target classes in the test image.

20 classes

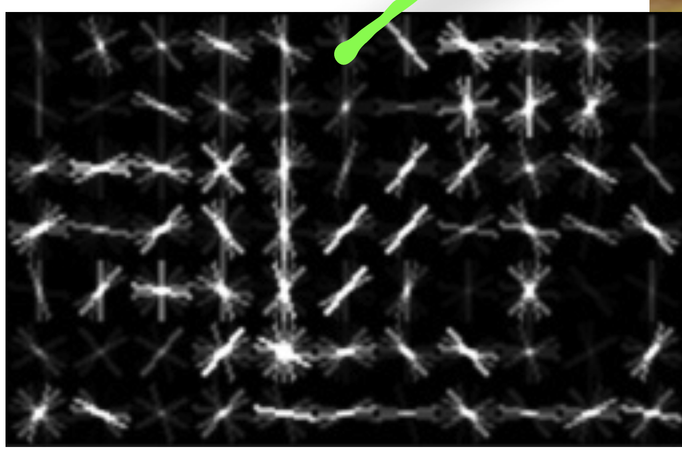
Participants may enter either (or both) of these competitions, and can choose to tackle any (or all) of the twenty object classes. The challenge allows for two approaches to each of the competitions:

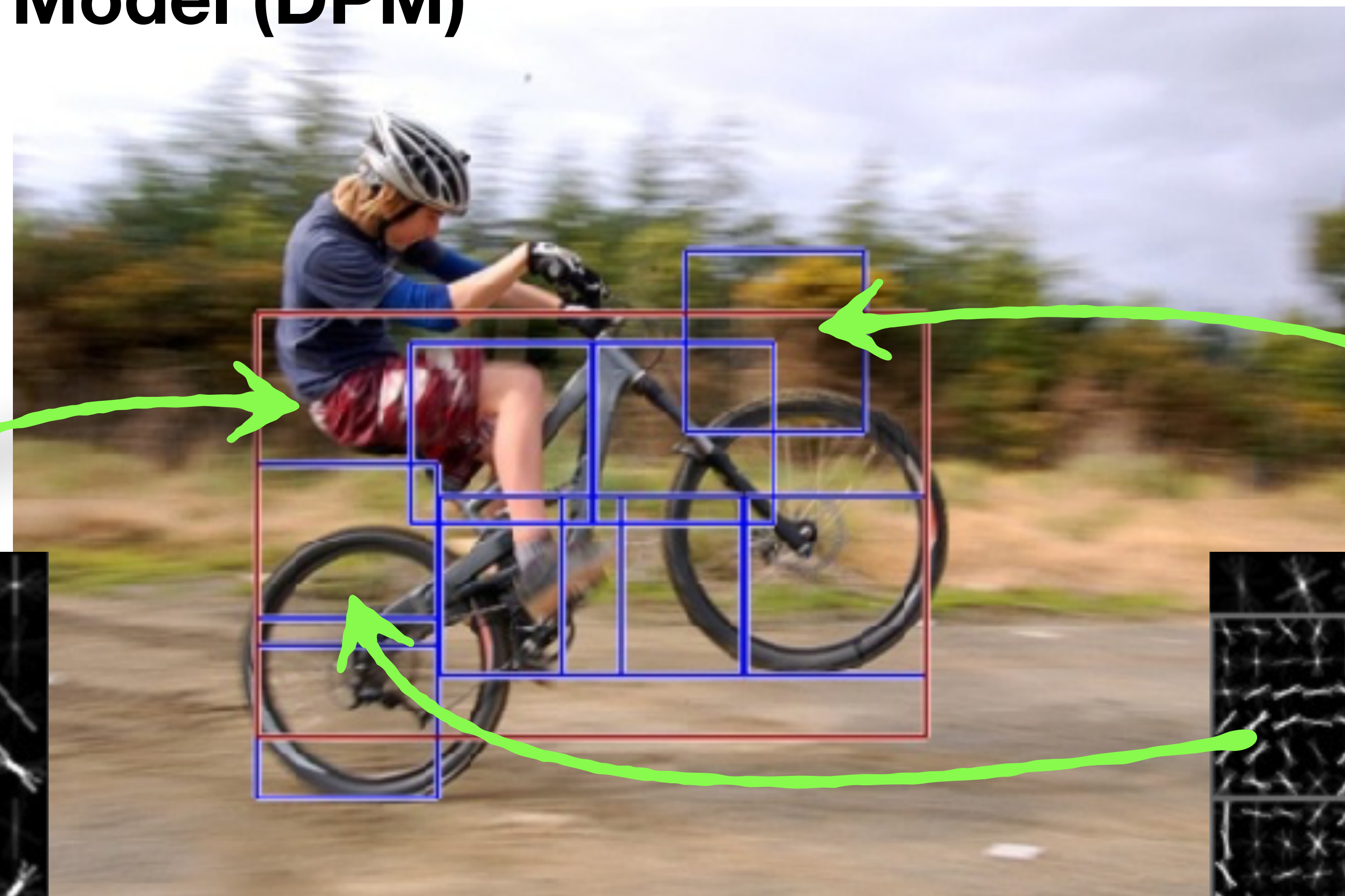
1. Participants may use systems built or trained using any methods or data excluding the provided test sets.
2. Systems are to be built or trained using only the provided training/validation data.

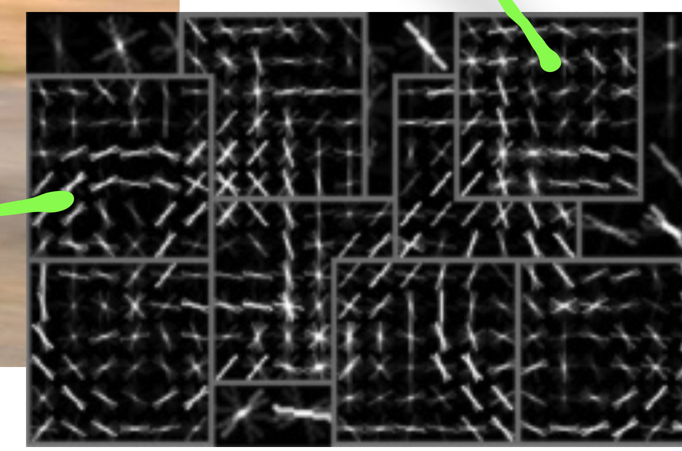
The intention in the first case is to establish just what level of success can currently be achieved on these problems and by what method; in the second case the intention is to establish which method is most successful given a specified training set.

Deformable Parts Model (DPM)

‘Root’ HOG







P. Felzenszwalb, D. McAllester, D. Ramanan
A Discriminatively Trained, Multiscale, Deformable Part Model
IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008

Parts HOG

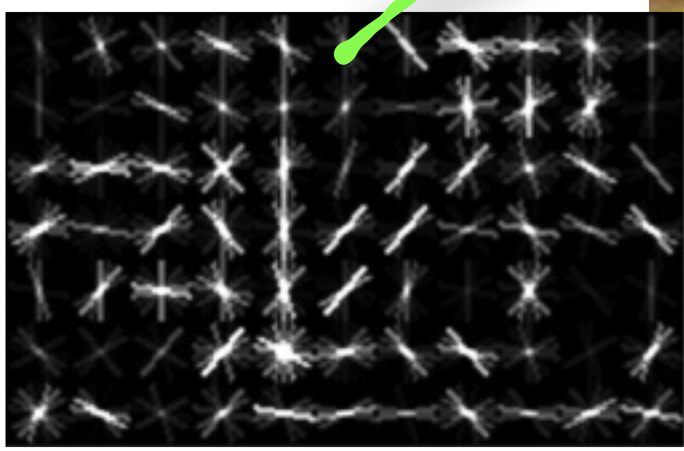
TABLE 1
Detection Average Precision (Percent) on VOC 2010 Test

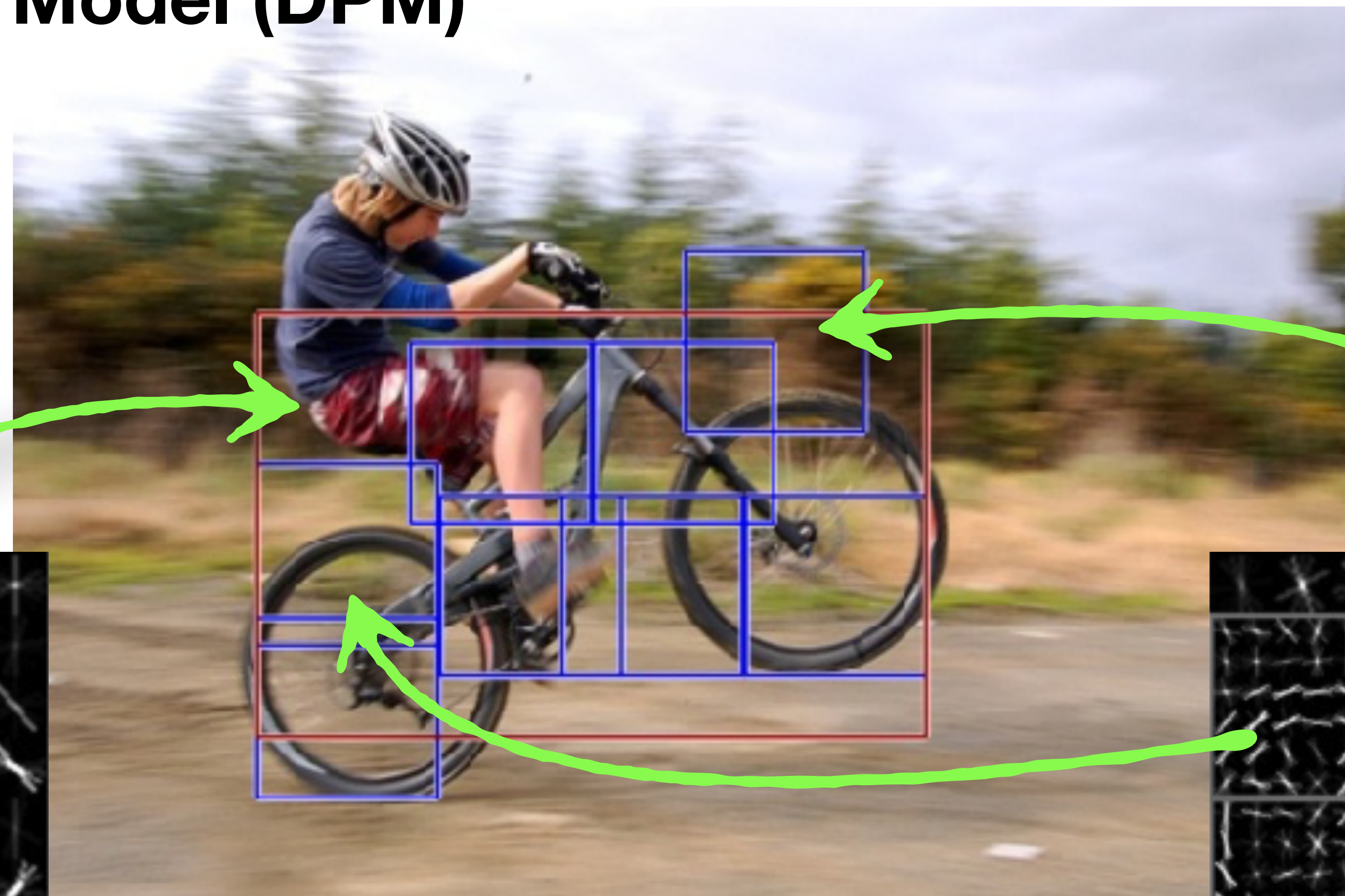
VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [23]	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4

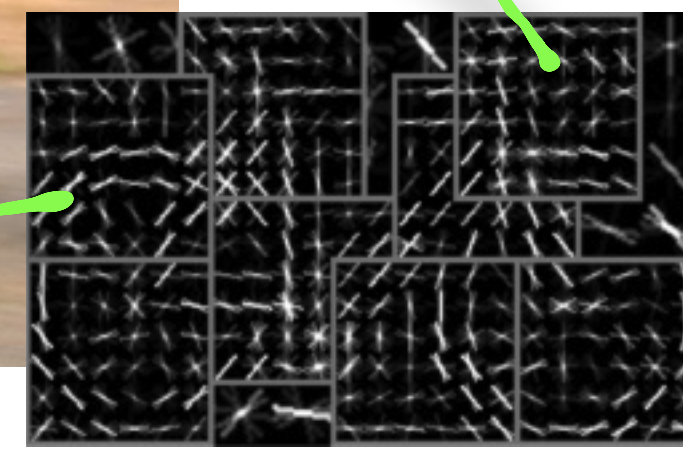
Best Method Pre-Deep Networks!

Deformable Parts Model (DPM)

'Root' HOG







P. Felzenszwalb, D. McAllester, D. Ramanan
A Discriminatively Trained, Multiscale, Deformable Part Model
IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008

Parts HOG

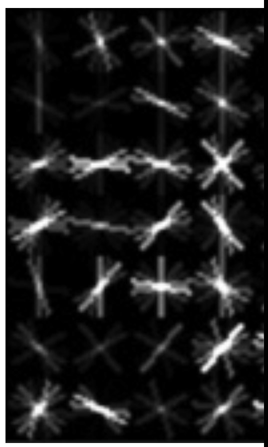
TABLE 1
Detection Average Precision (Percent) on VOC 2010 Test

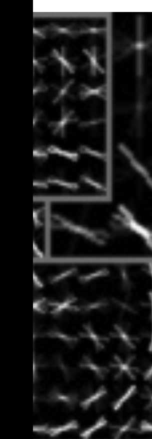
VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [23]	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [21]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [54]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [57]	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN T-Net	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN T-Net BB	71.8	65.8	53.0	36.8	35.9	59.7	60.0	69.9	27.9	50.6	41.4	70.0	62.0	69.0	58.1	29.5	59.4	39.3	61.2	52.4	53.7
R-CNN O-Net	76.5	70.4	58.0	40.2	39.6	61.8	63.7	81.0	36.2	64.5	45.7	80.5	71.9	74.3	60.6	31.5	64.7	52.5	64.6	57.2	59.8
R-CNN O-Net BB	**79.3**	**72.4**	**63.1**	**44.0**	**44.4**	**64.6**	**66.3**	**84.9**	**38.8**	**67.3**	**48.4**	**82.3**	**75.0**	**76.7**	**65.7**	**35.8**	**66.2**	**54.8**	**69.1**	**58.8**	**62.9**

T-Net stands for TorontoNet and O-Net for OxfordNet (Section 3.1.2). R-CNNs are most directly comparable to UVA and Regionlets since all methods use selective search region proposals. Bounding-box regression is described in Section 7.3. At publication time, SegDPM was the top-performer on the PASCAL VOC leaderboard. DPM and SegDPM use context rescoring not used by the other methods. SegDPM and all R-CNNs use additional training data.

Deform

'Root' HOG





HOG

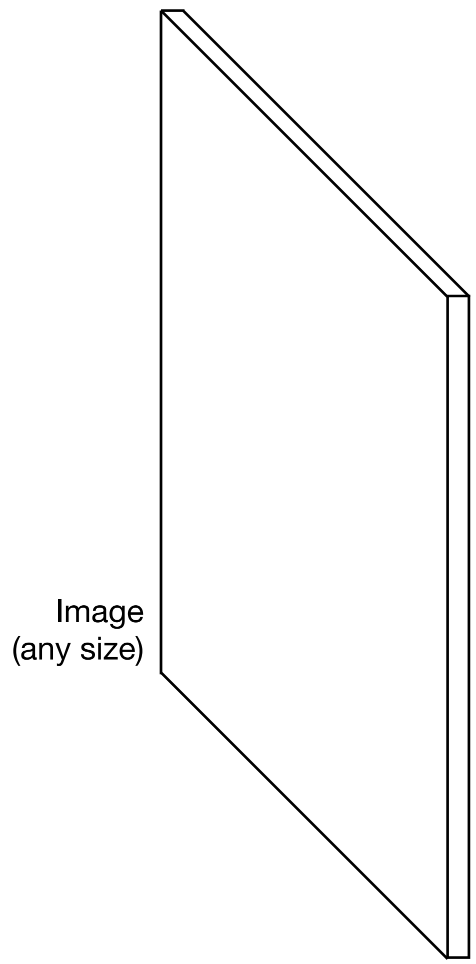
VOC 201																					mAP
DPM v5																					33.4
UVA [21																					35.1
Regionle																					39.7
SegDPM																					40.4
R-CNN T																					50.2
R-CNN T-Net BB	71.8	65.8	53.0	36.8	35.9	59.7	60.0	69.9	27.9	50.6	41.4	70.0	62.0	69.0	58.1	29.5	59.4	39.3	61.2	52.4	53.7
R-CNN O-Net	76.5	70.4	58.0	40.2	39.6	61.8	63.7	81.0	36.2	64.5	45.7	80.5	71.9	74.3	60.6	31.5	64.7	52.5	64.6	57.2	59.8
R-CNN O-Net BB	**79.3**	**72.4**	**63.1**	**44.0**	**44.4**	**64.6**	**66.3**	**84.9**	**38.8**	**67.3**	**48.4**	**82.3**	**75.0**	**76.7**	**65.7**	**35.8**	**66.2**	**54.8**	**69.1**	**58.8**	**62.9**

T-Net stands for TorontoNet and O-Net for OxfordNet (Section 3.1.2). R-CNNs are most directly comparable to UVA and Regionlets since all methods use selective search region proposals. Bounding-box regression is described in Section 7.3. At publication time, SegDPM was the top-performer on the PASCAL VOC leaderboard. DPM and SegDPM use context rescoring not used by the other methods. SegDPM and all R-CNNs use additional training data.

It was a simple idea…

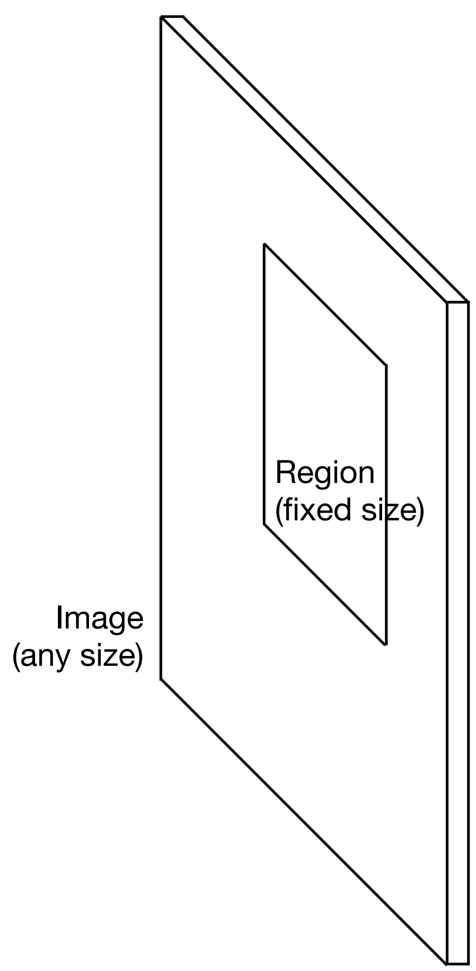
RCNN

(Region-based Convolutional Network)



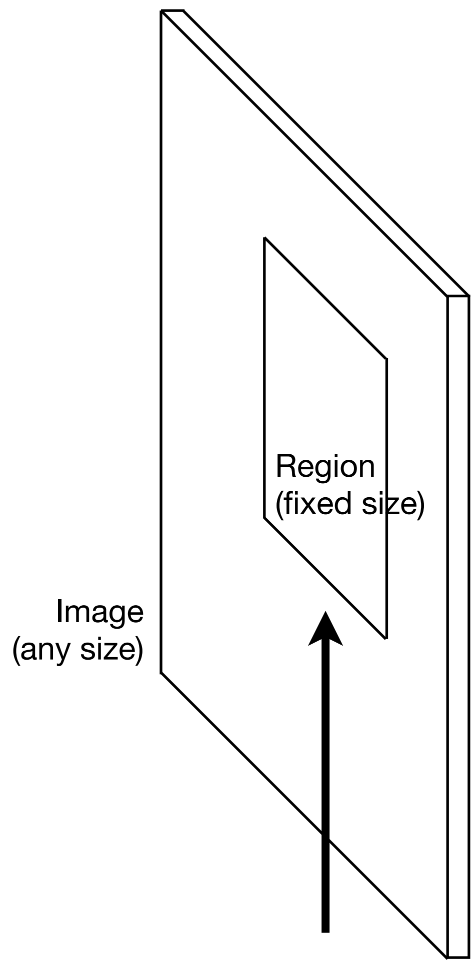
RCNN

(Region-based Convolutional Network)



RCNN

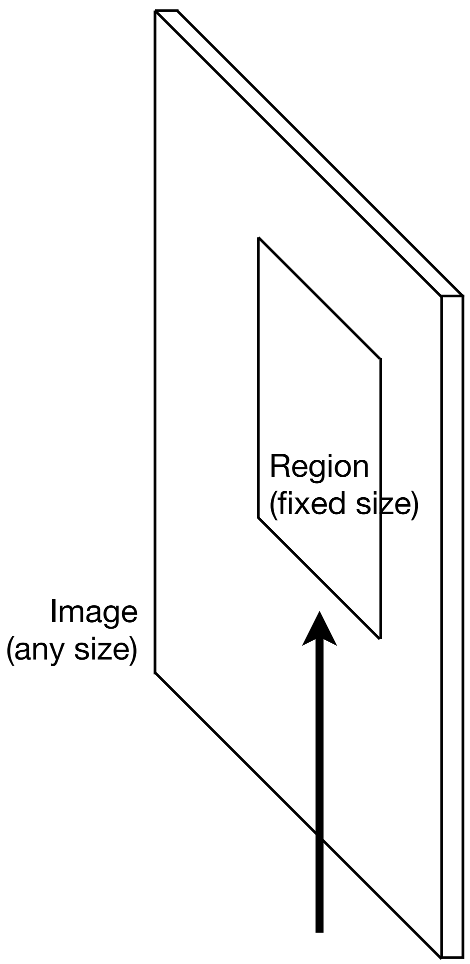
(Region-based Convolutional Network)



Region Generator

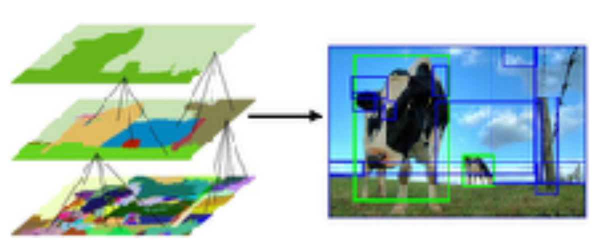
RCNN

(Region-based Convolutional Network)



Region Generator

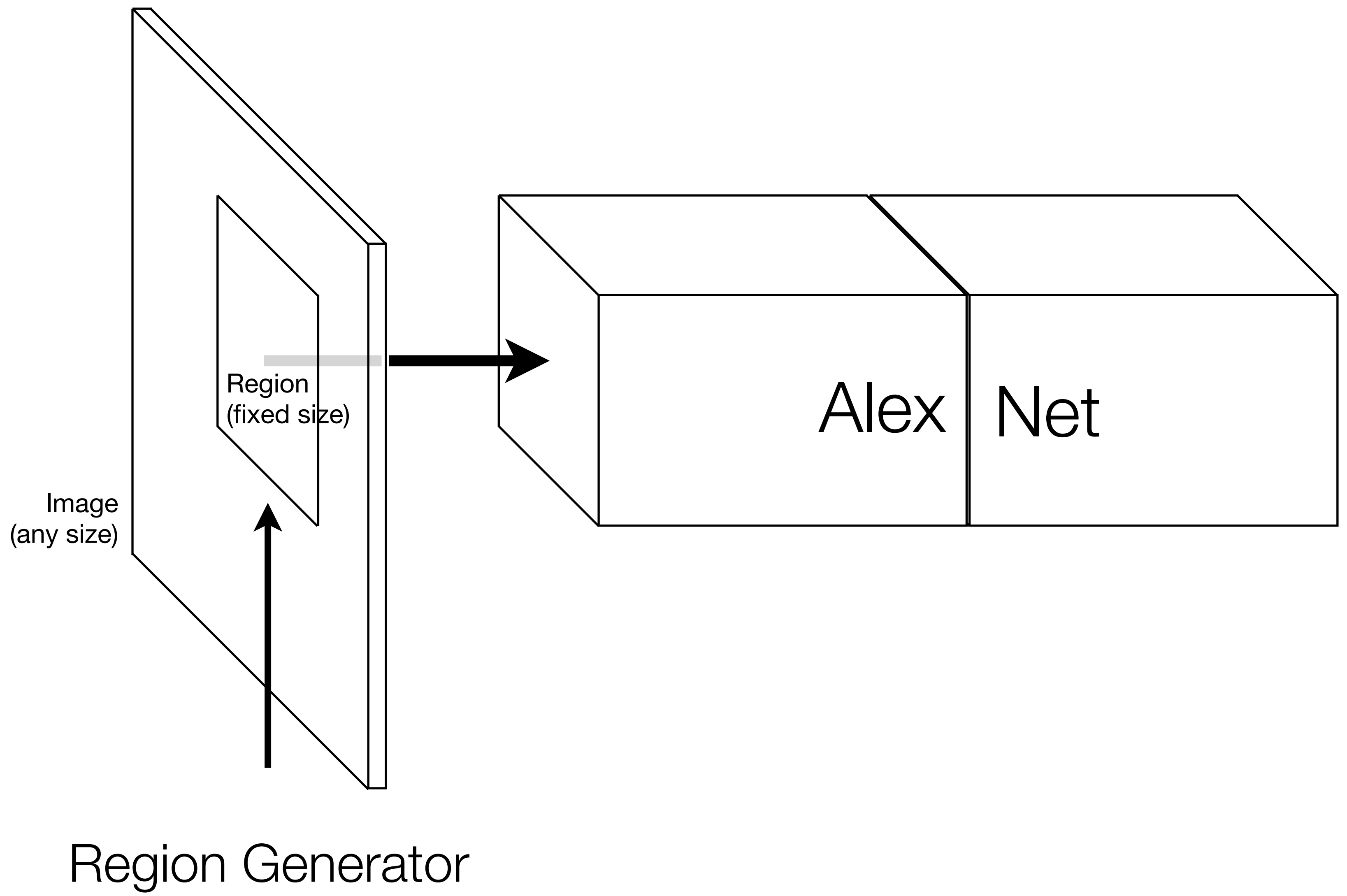
'Selective Search'



Selective Search for Object Recognition
J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders
In International Journal of Computer Vision 2013.

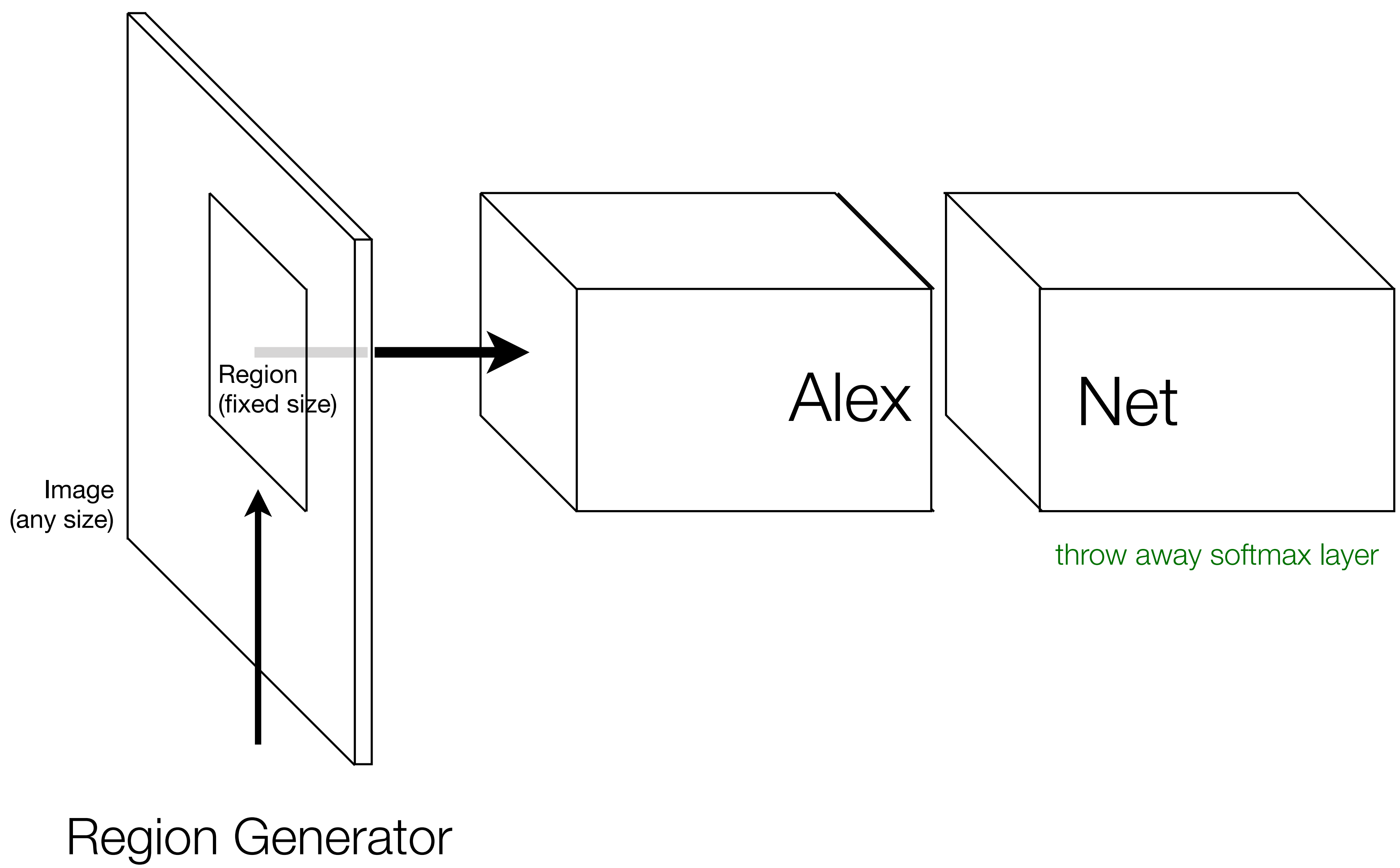
RCNN

(Region-based Convolutional Network)



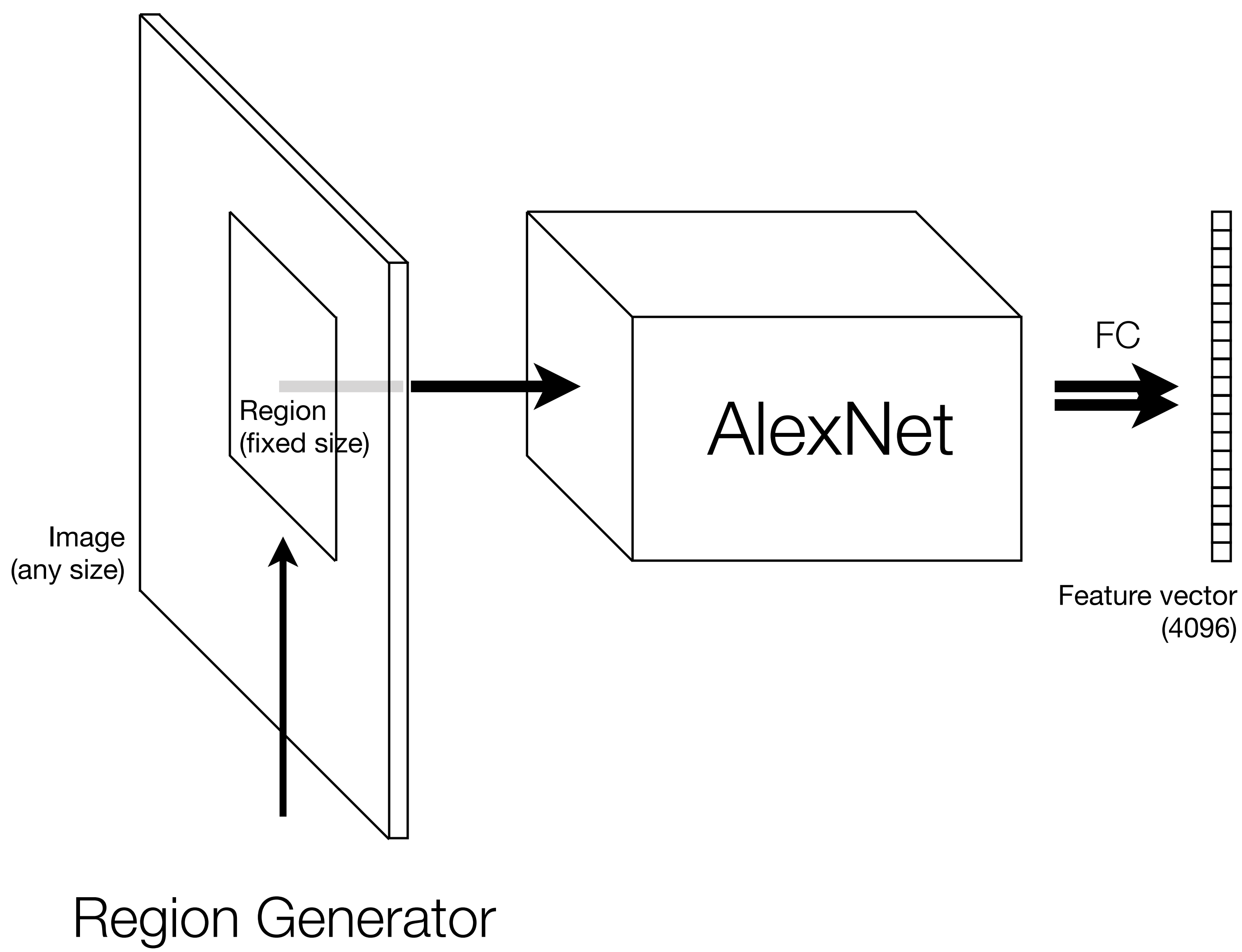
RCNN

(Region-based Convolutional Network)



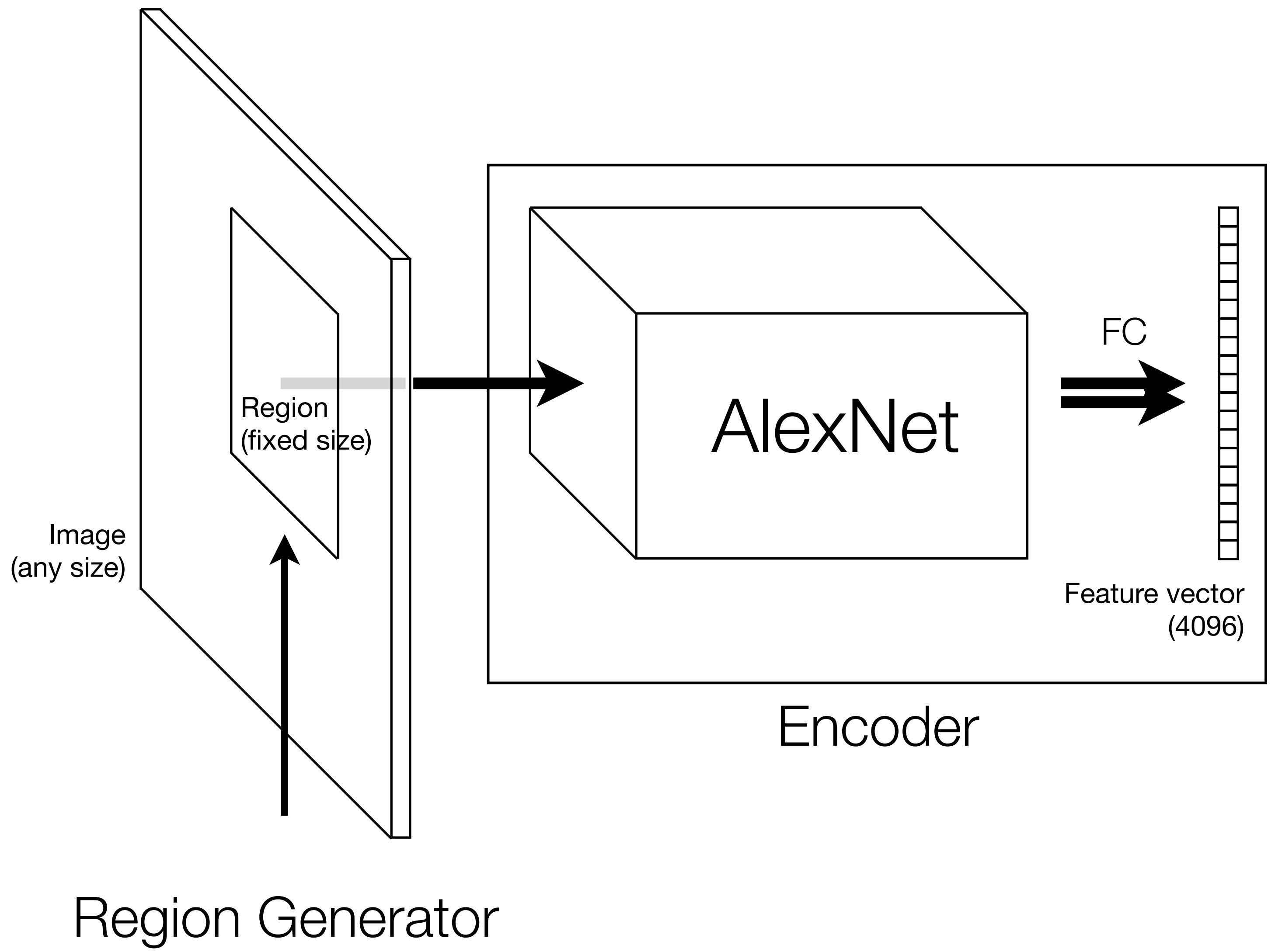
RCNN

(Region-based Convolutional Network)



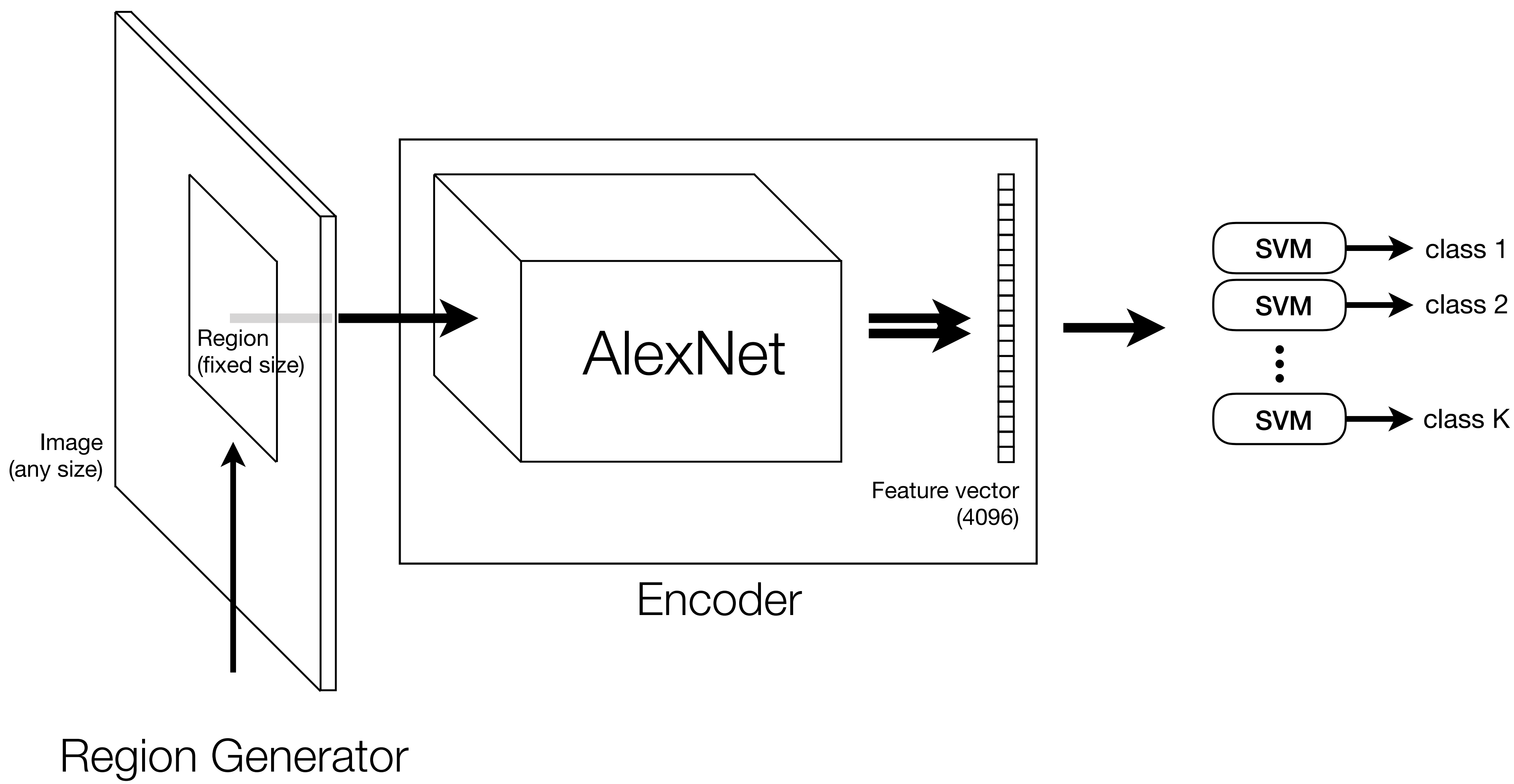
RCNN

(Region-based Convolutional Network)



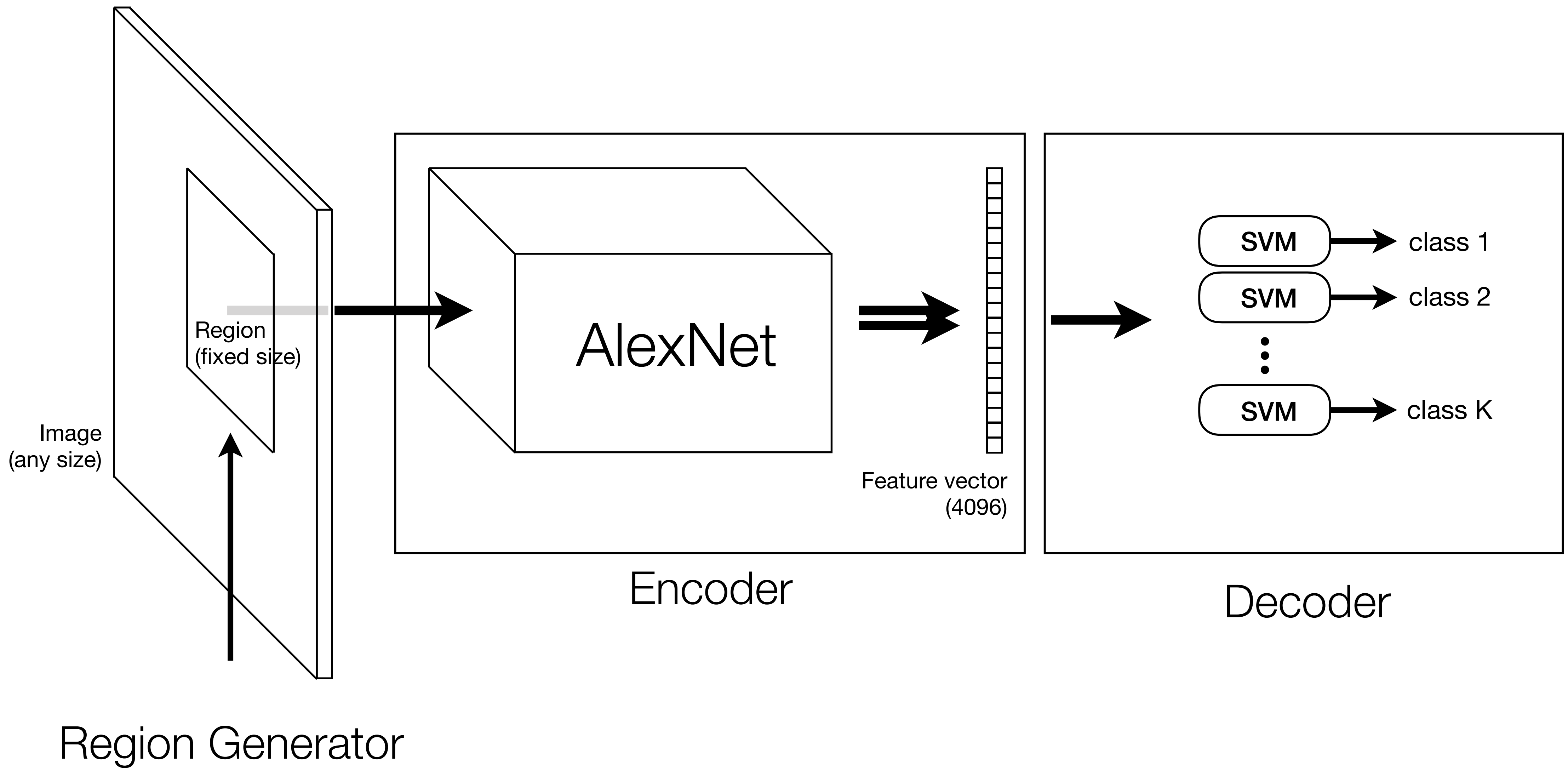
RCNN

(Region-based Convolutional Network)



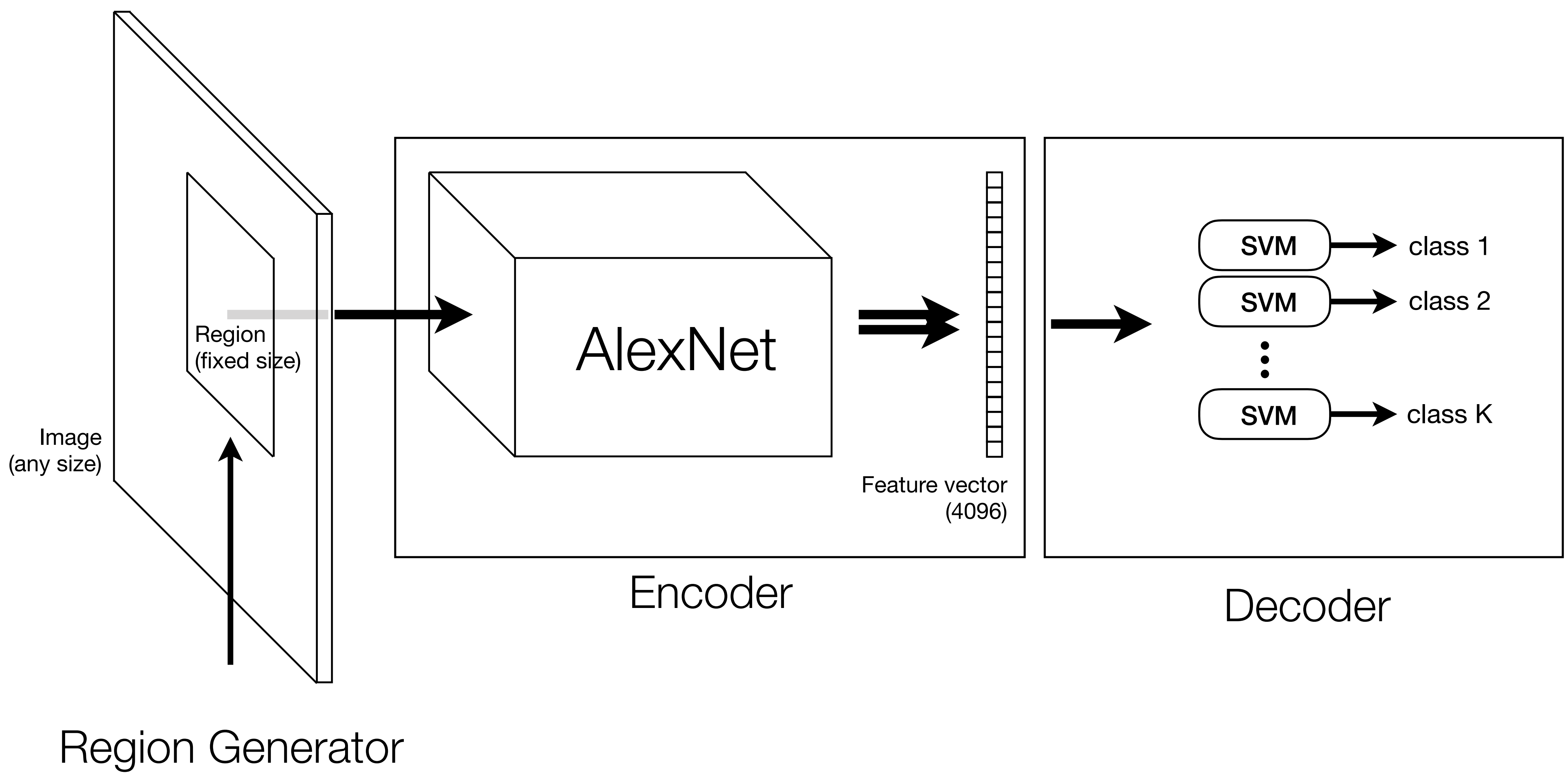
RCNN

(Region-based Convolutional Network)



RCNN

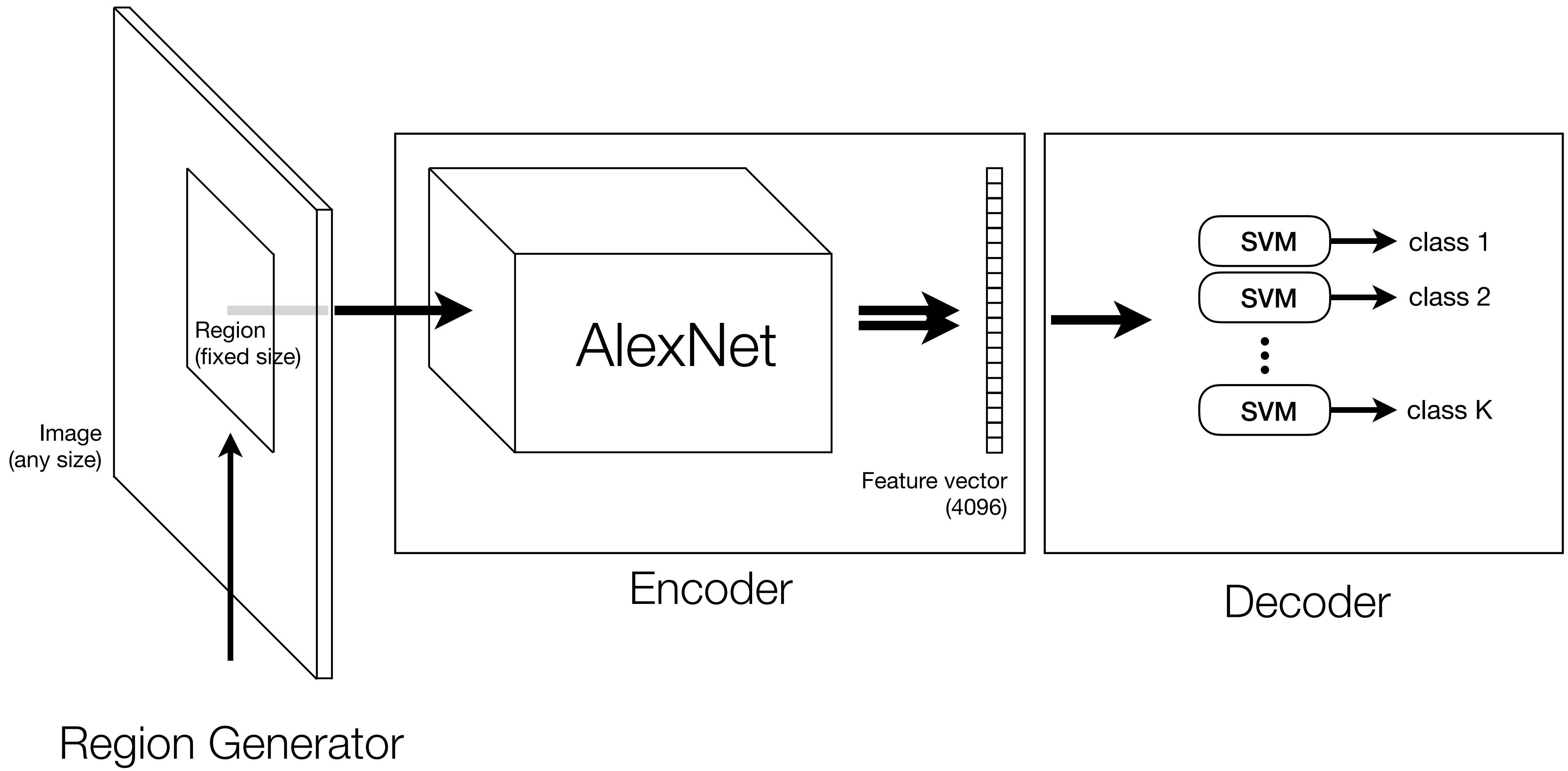
(Region-based Convolutional Network)



… that's it.

RCNN

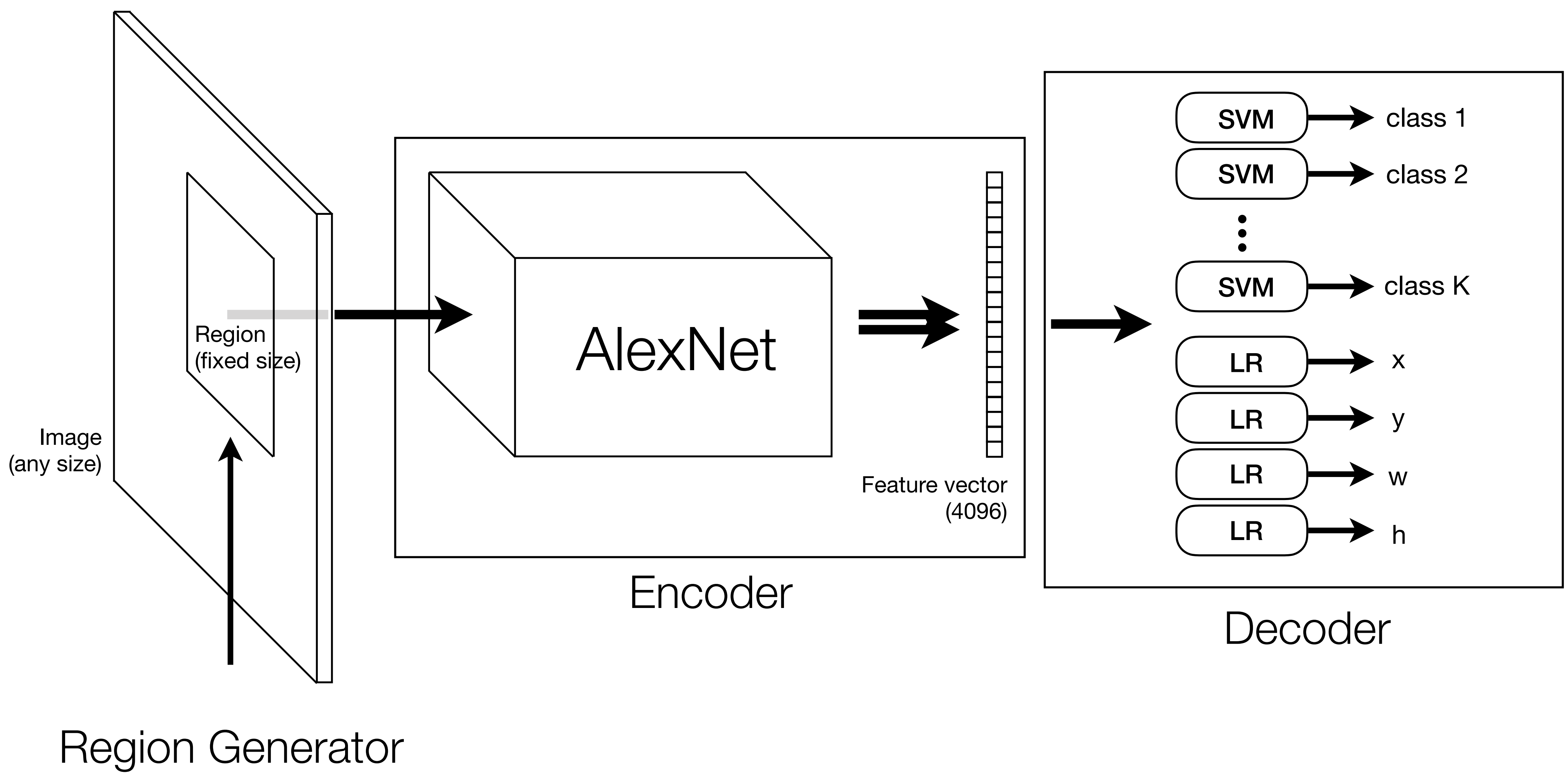
(Region-based Convolutional Network)



Region Generator bounding boxes are not always perfect.
What should we do?

RCNN

(Region-based Convolutional Network)



Bounding Box Regression!

RCNN Training Tricks

- Supervised Pre-Training
  (CNN pre-trained for classification on ILSVRC 2012)
- Domain-specific Fine-Tuning
  (replace last layer, learning rate of 0.001, train for classification on VOC images)
- SVM trained over FC7 features
  (one linear SVM per class)

Important Concepts

- Region Proposals
- Bounding box regression
- Classification CNN as a feature extractor
- Fine-tuning

Visualization of top six 'Pool5' activations

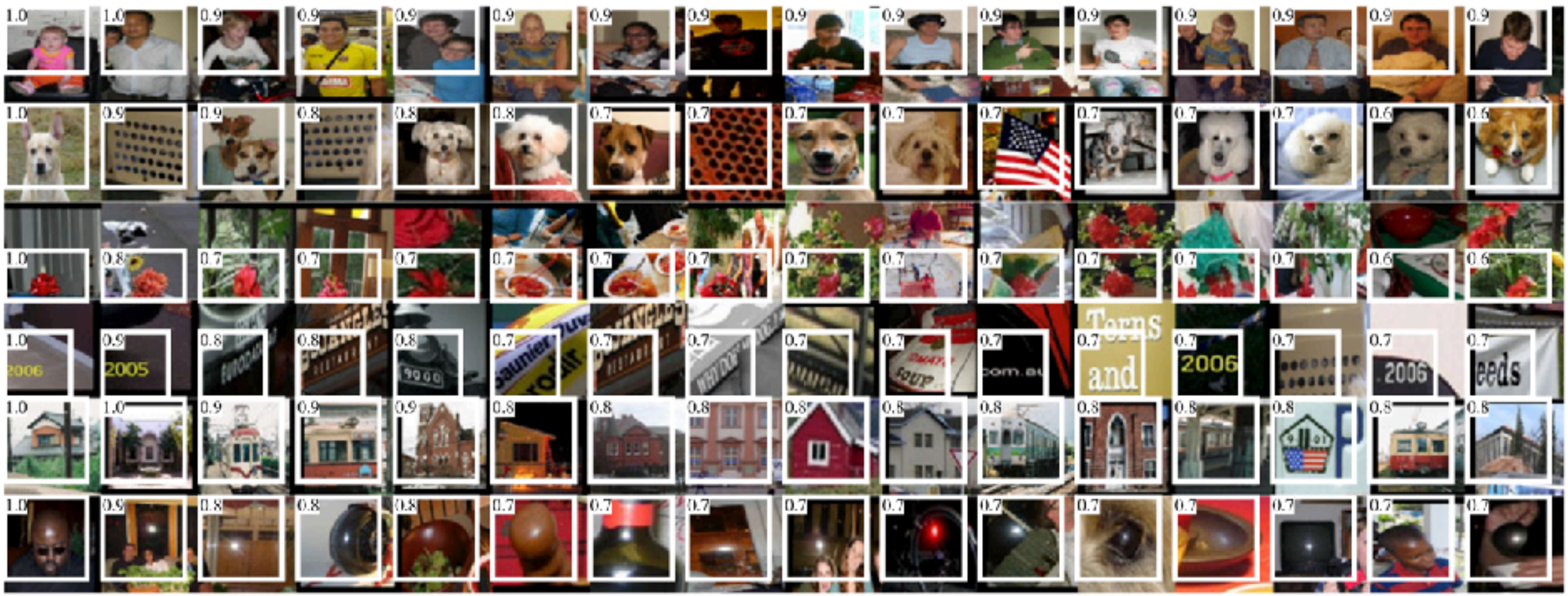
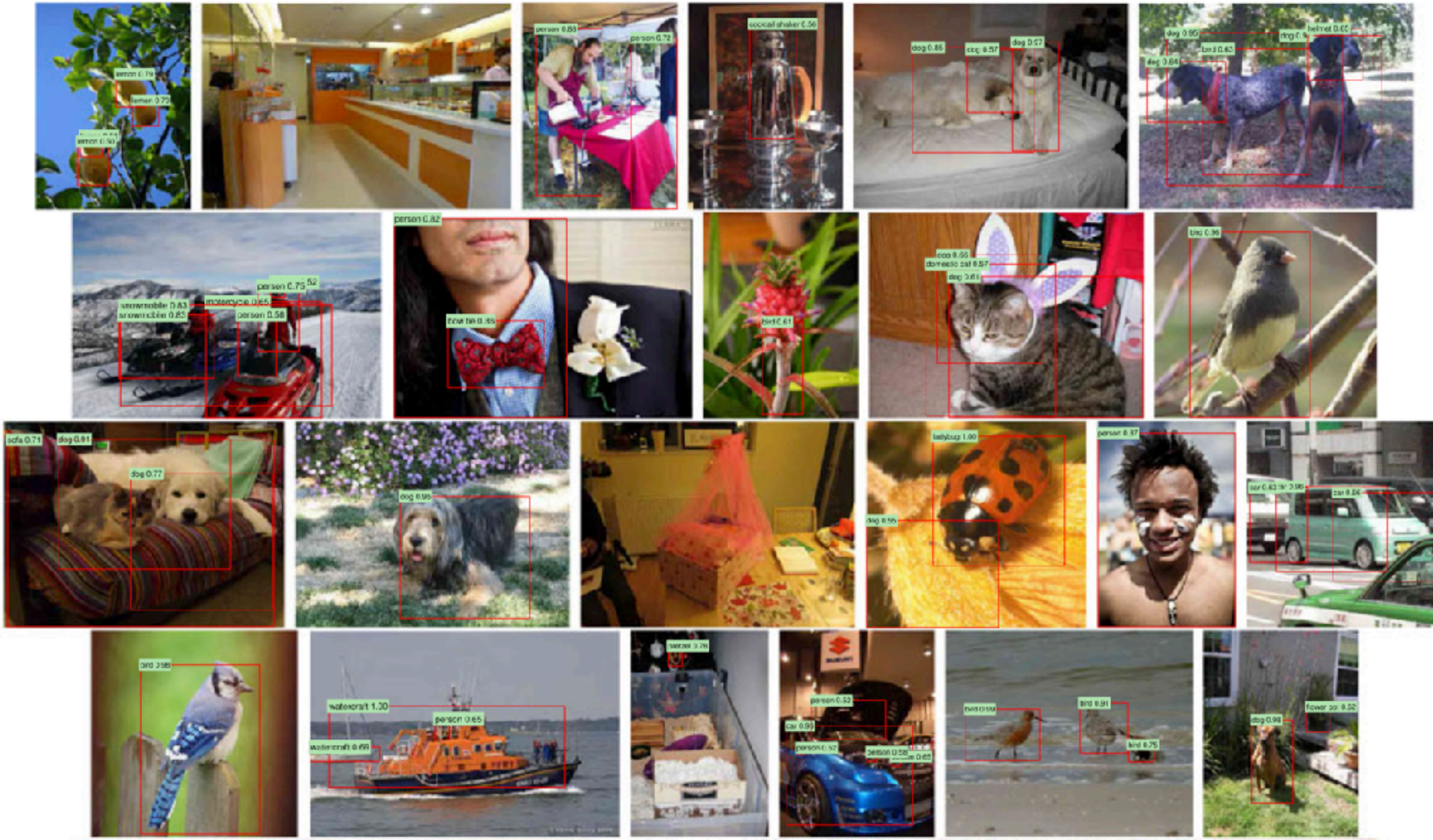


Fig. 4. Top regions for six pool_5 units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

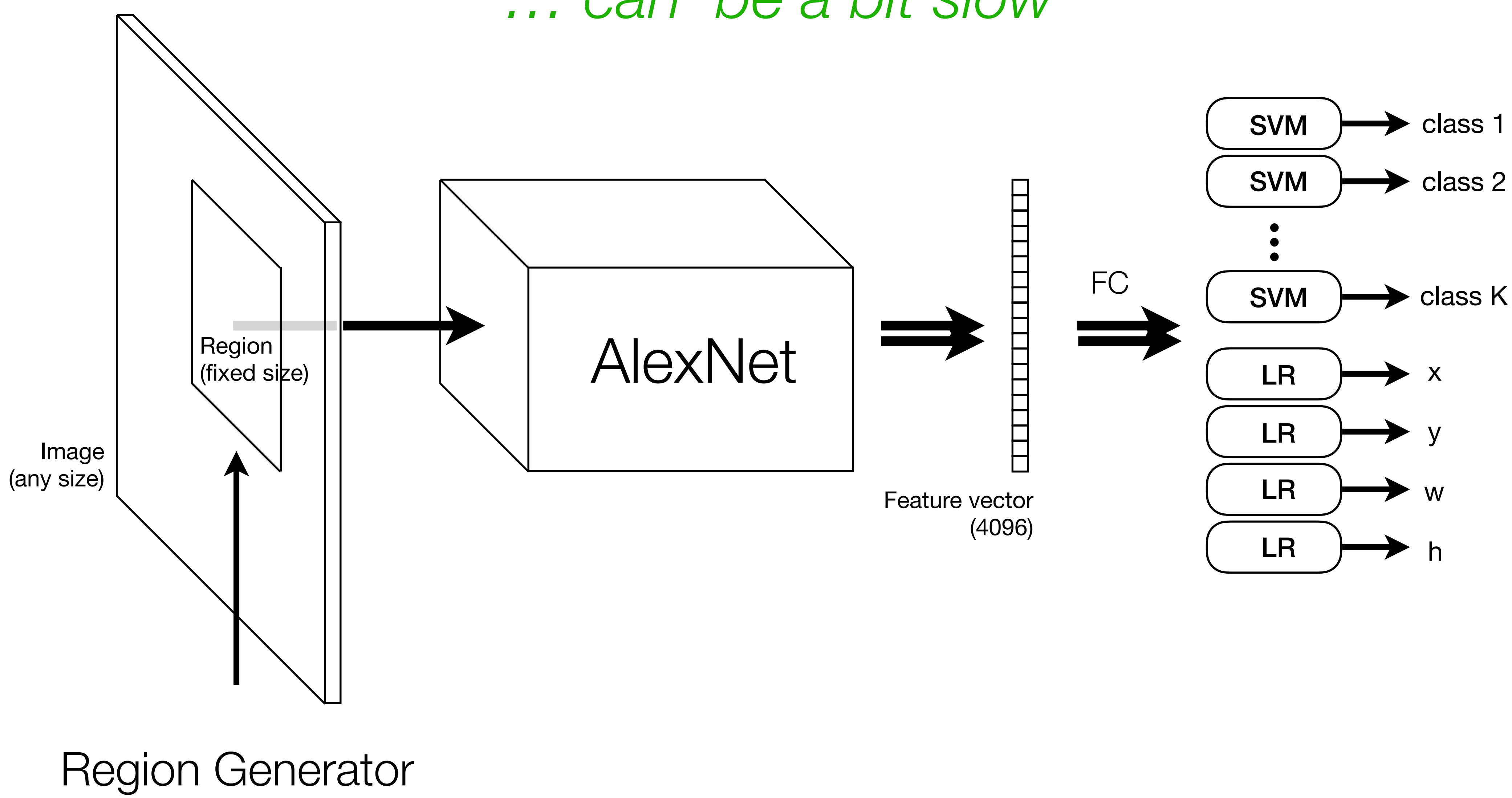


Randomly sampled results

RCNN

(Region-based Convolutional Network)

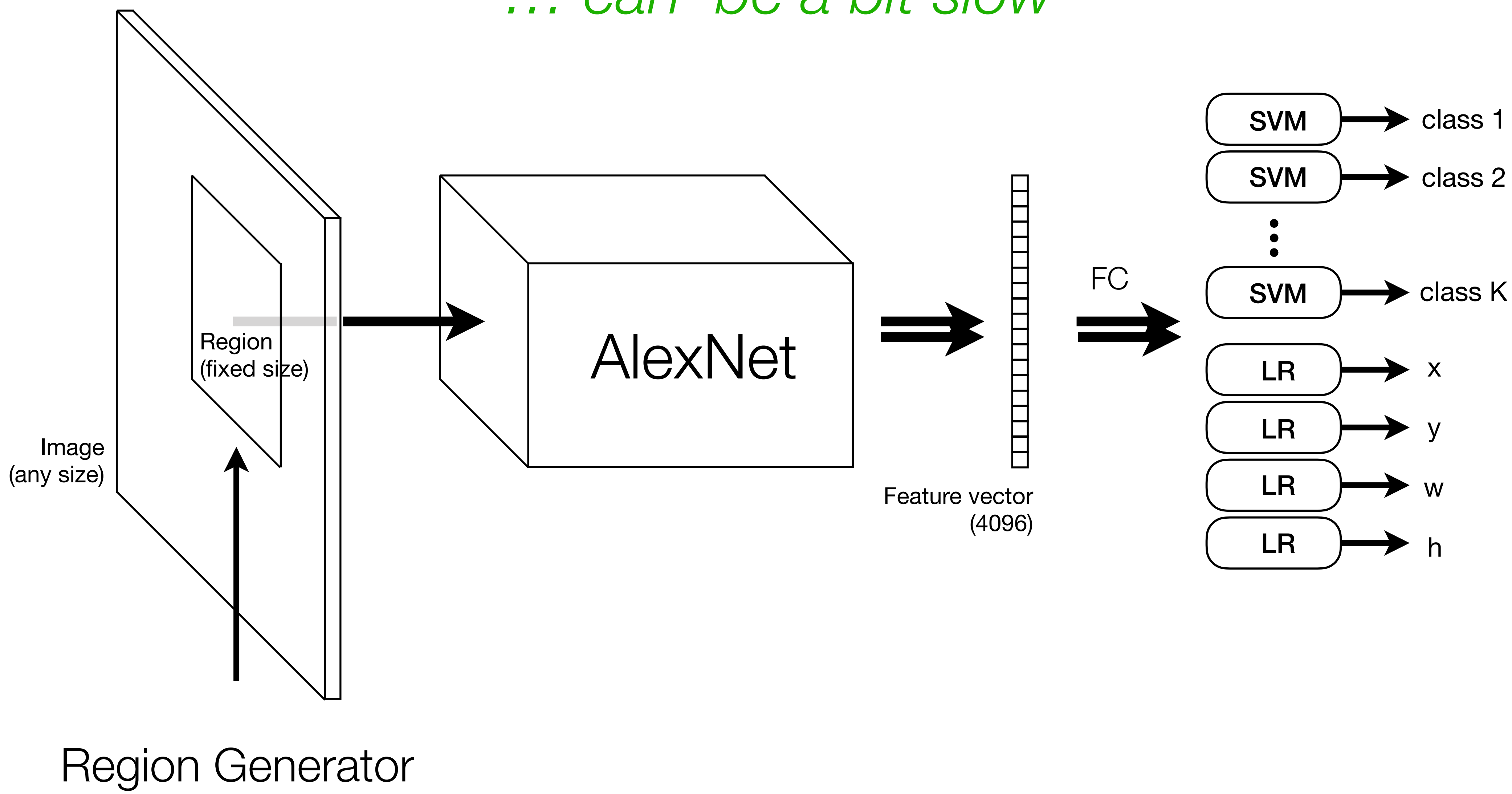
… can be a bit slow



RCNN

(Region-based Convolutional Network)

... can be a bit slow

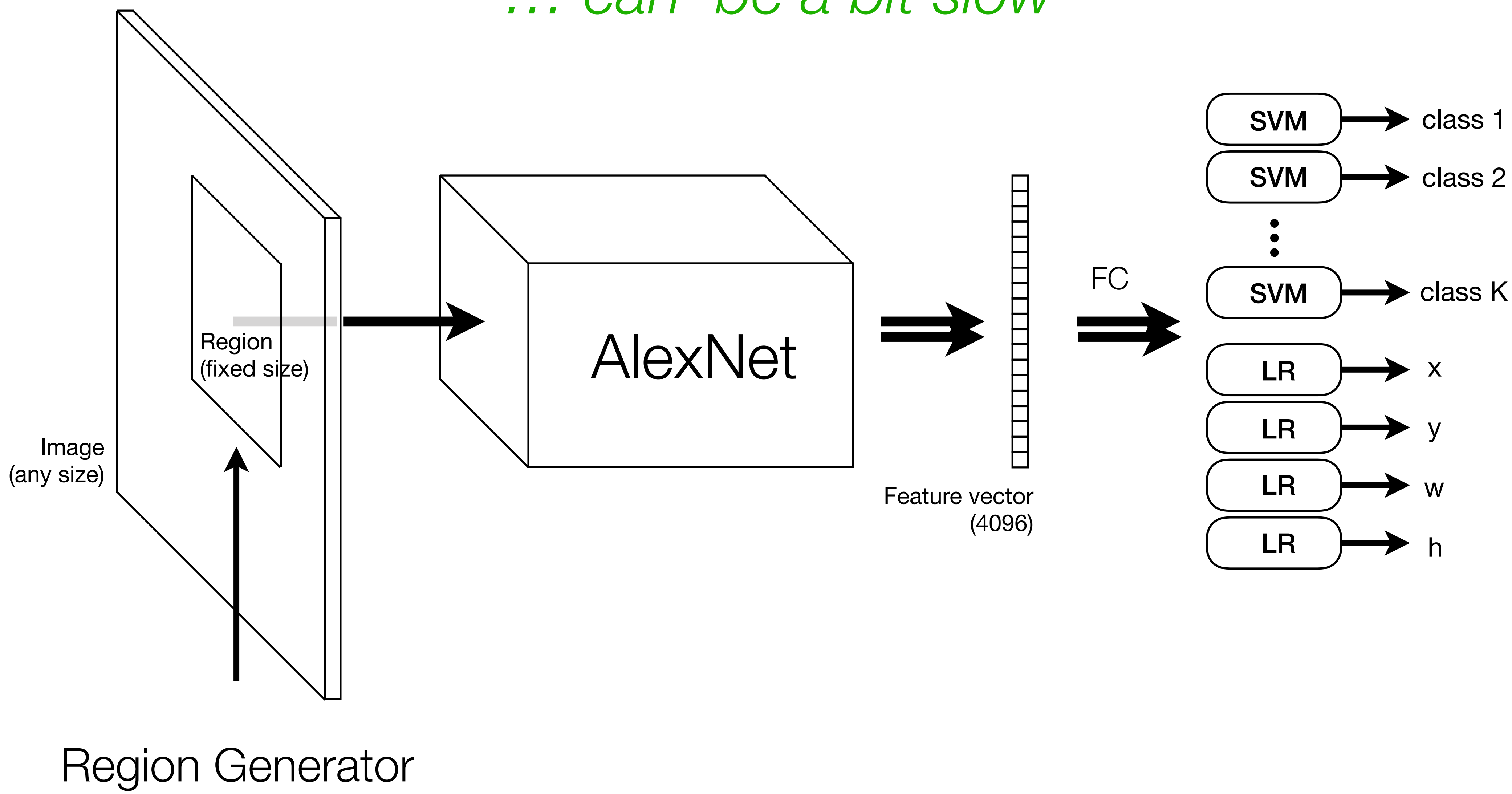


Images can generate around 2000 regions.

RCNN

(Region-based Convolutional Network)

… can be a bit slow



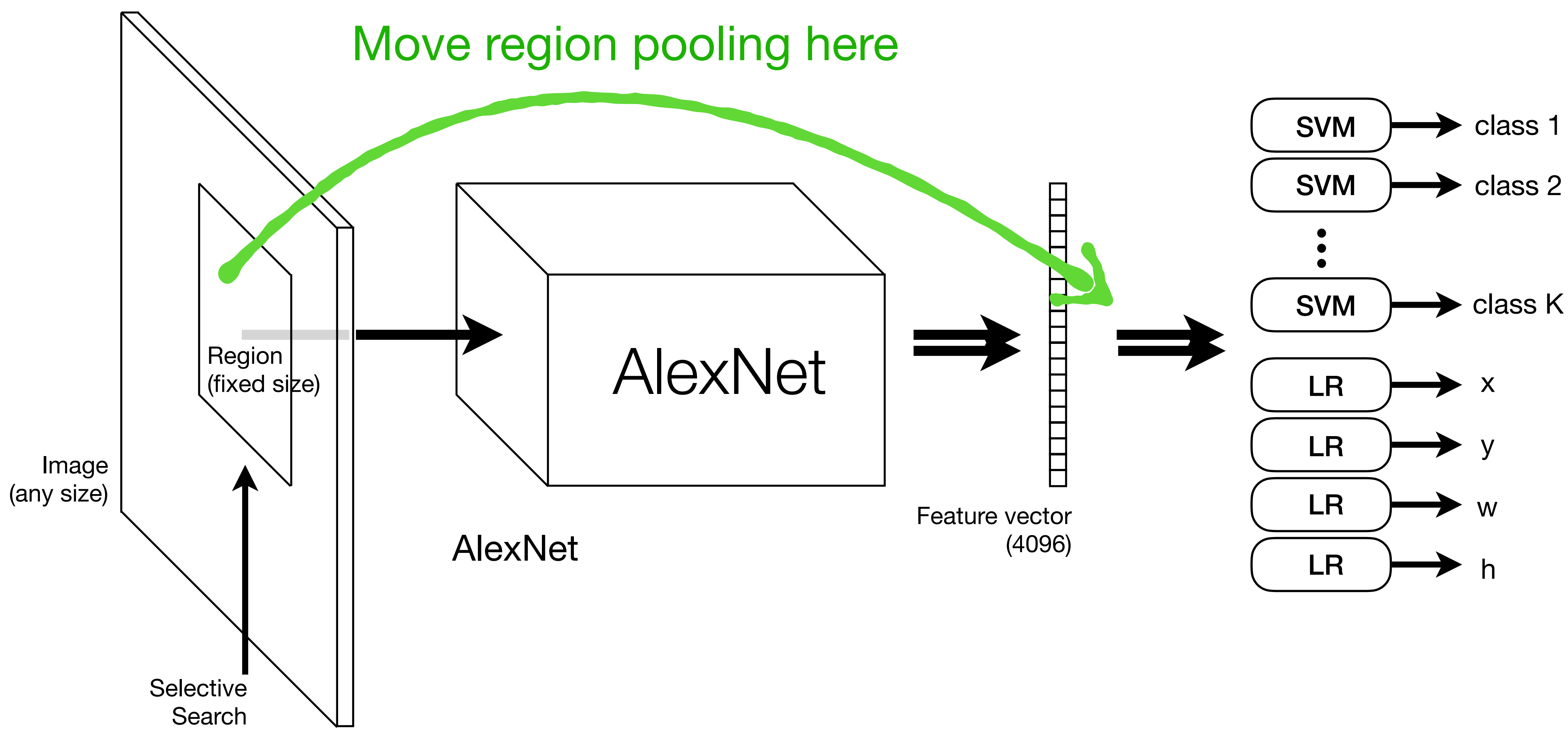
Images can generate around 2000 regions.
How many times do we perform the forward pass?
Can we speed this up?

YES!!!

RCNN

(Region Convolutional Network)

Move region pooling here

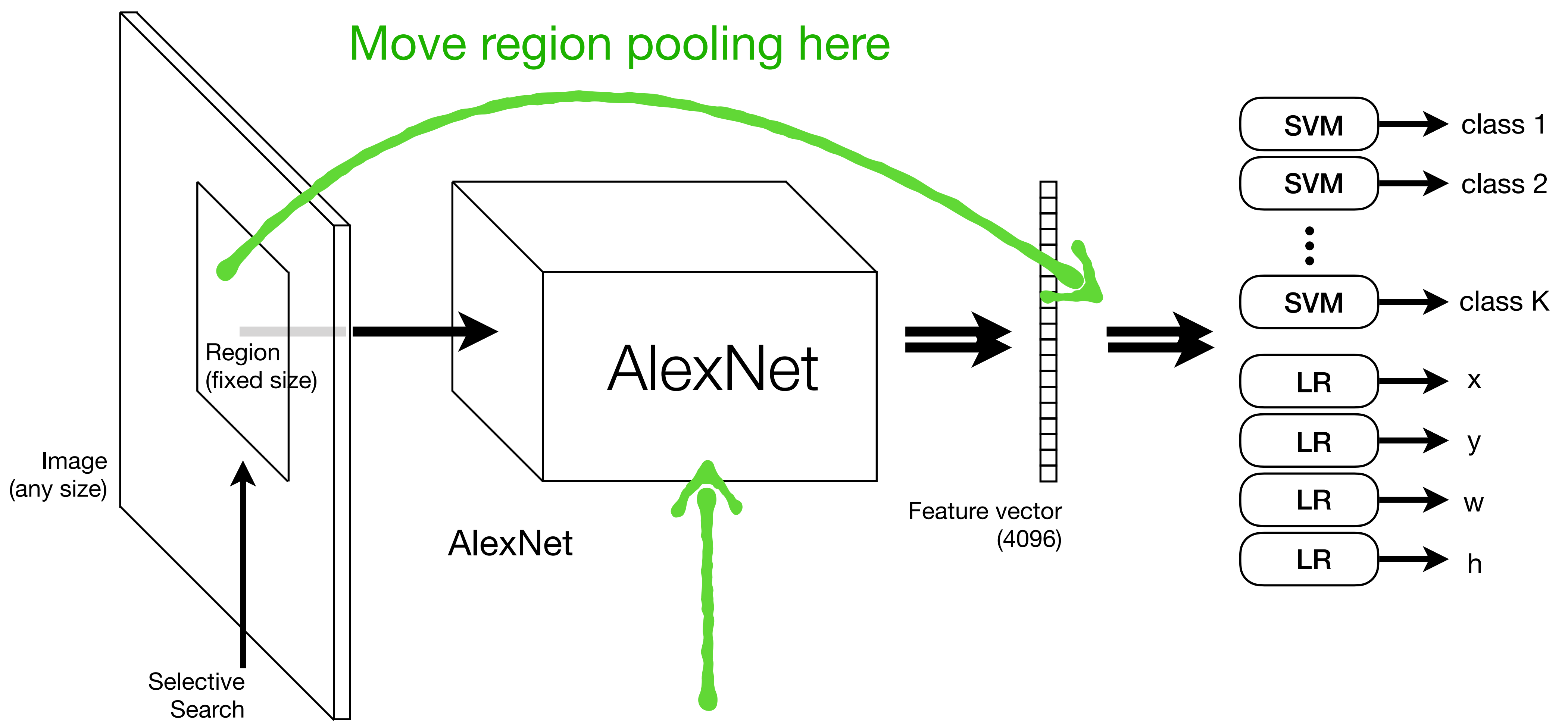


YES!!!

RCNN

(Region Convolutional Network)

Move region pooling here

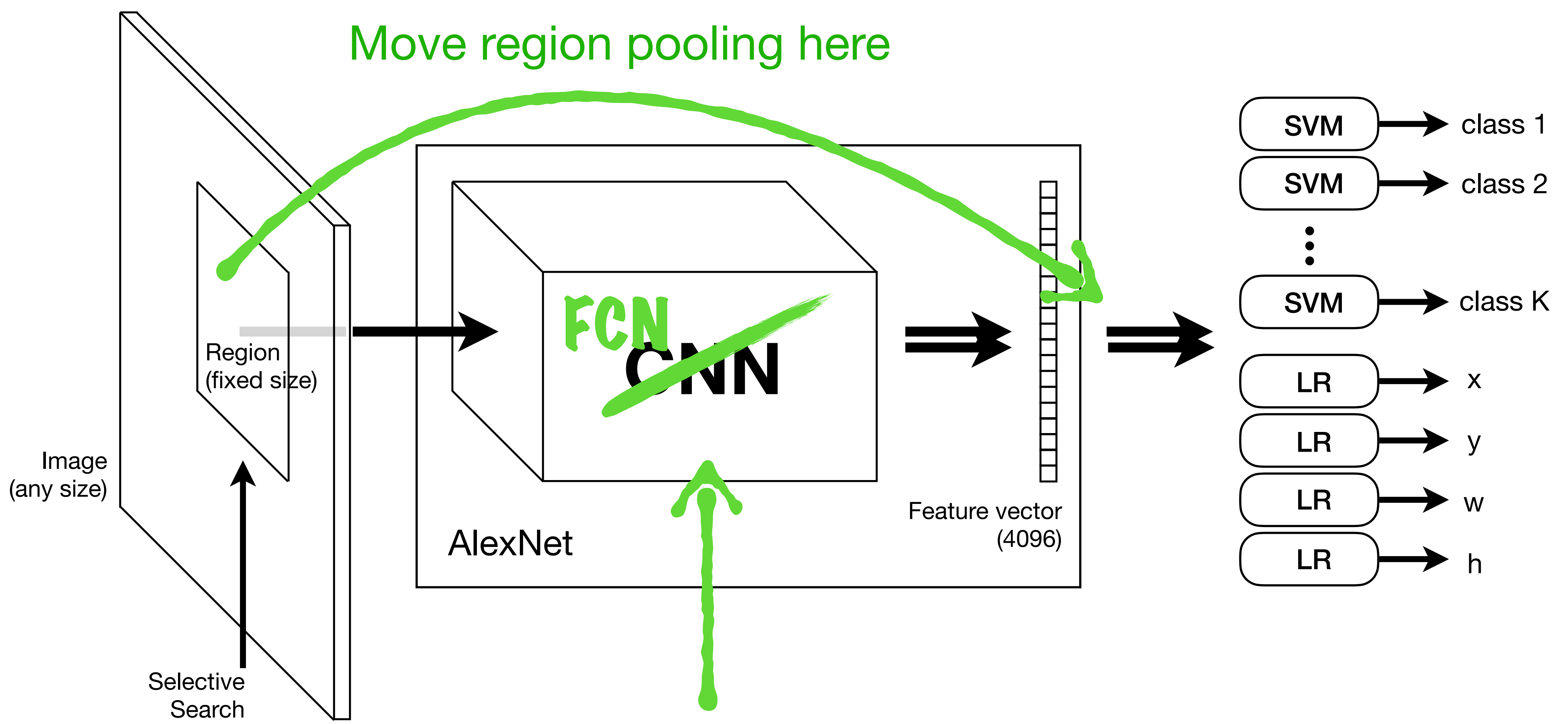


Extend to deal with
different image sizes

YES!!!

Fast RCNN

(Region Convolutional Network)



FCN here to deal with
different image sizes