Lec12 - Object Detection

CS 783 - Visual Recognition

Vinay P. Namboodiri

IIT Kanpur

14^{th} February 2019

Contents

Outline

Object Detection

We now consider the problem of object detection.

Figure: An example of object detection, fig. from P. Kontschneider et al., NIPS 2012

Object Detection

We now consider the problem of object detection.

Figure: An example of object detection, fig. from P. Kontschneider et al., NIPS 2012

- Formally, in object detection, the problem is to be able to specify the set of bounding boxes B for an image I that contain instances of a class C or not.

Object Detection

We now consider the problem of object detection.

Figure: An example of object detection, fig. from P. Kontschneider et al., NIPS 2012

- Formally, in object detection, the problem is to be able to specify the set of bounding boxes B for an image I that contain instances of a class C or not.
- In contrast to classification where given an image, we just had to specify the class, here, for an image, we have to output the classes and the locations of the various instances of objects.

Challenges of Object Detection

We now consider the problem of object detection.

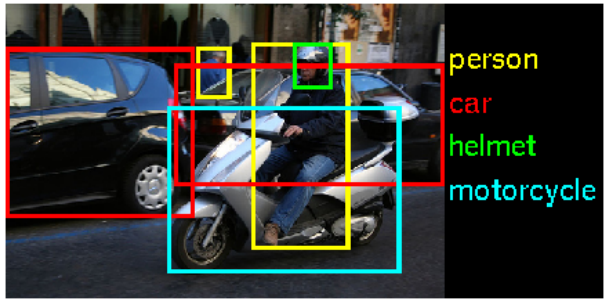


Figure: Example of a typical object detection task in an image

Challenges of Object Detection

We now consider the problem of object detection.

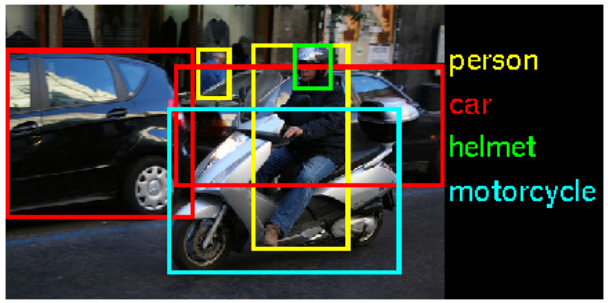


Figure: Example of a typical object detection task in an image

- In object detection, there may be several classes of objects present in an image, many with overlapping bounding boxes. The challenge is to individually identify each object of a class.

Challenges of Object Detection

We now consider the problem of object detection.

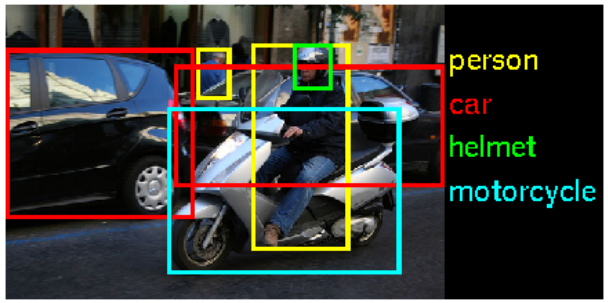


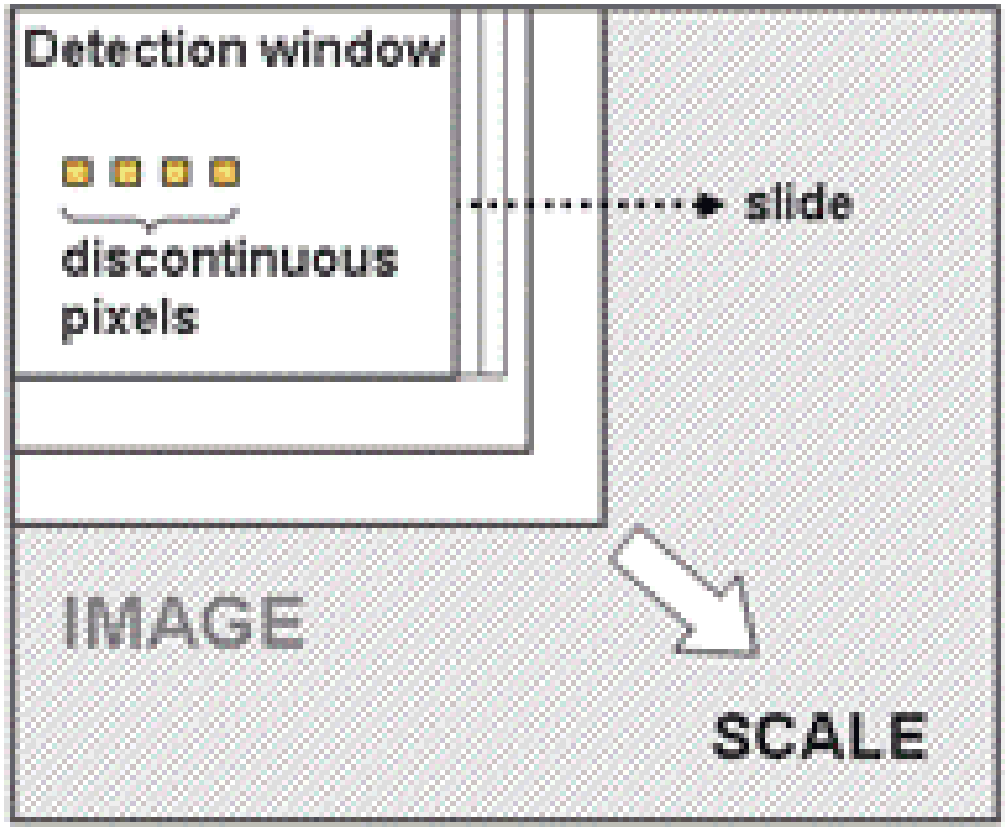
Figure: Example of a typical object detection task in an image

- In object detection, there may be several classes of objects present in an image, many with overlapping bounding boxes. The challenge is to individually identify each object of a class.
- The additional challenges of there being different sources of variation due to viewpoint, illumination, articulation, occlusion, are also present. Additionally the background can be varied and has to be separated in each case.

Outline

Object Detection as Classification

One of the early successful approaches adopted for solving the problem of object detection viewed the task as one of classification. This was through a *sliding window* classifier approach



Object Detection as Classification

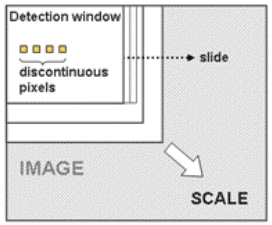


Figure: Illustration of a sliding window, figure courtesy Leron Fliess, Yifat Chernihov and Stav Ashuri

Object Detection as Classification

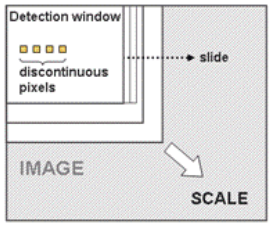


Figure: Illustration of a sliding window, figure courtesy Leron Fliess, Yifat Chernihov and Stav Ashuri

- In a sliding window classifier, one uses a systematic exploration of the image space obtaining individual bounding boxes at each iteration.

Object Detection as Classification

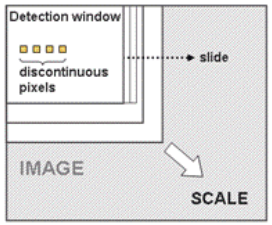
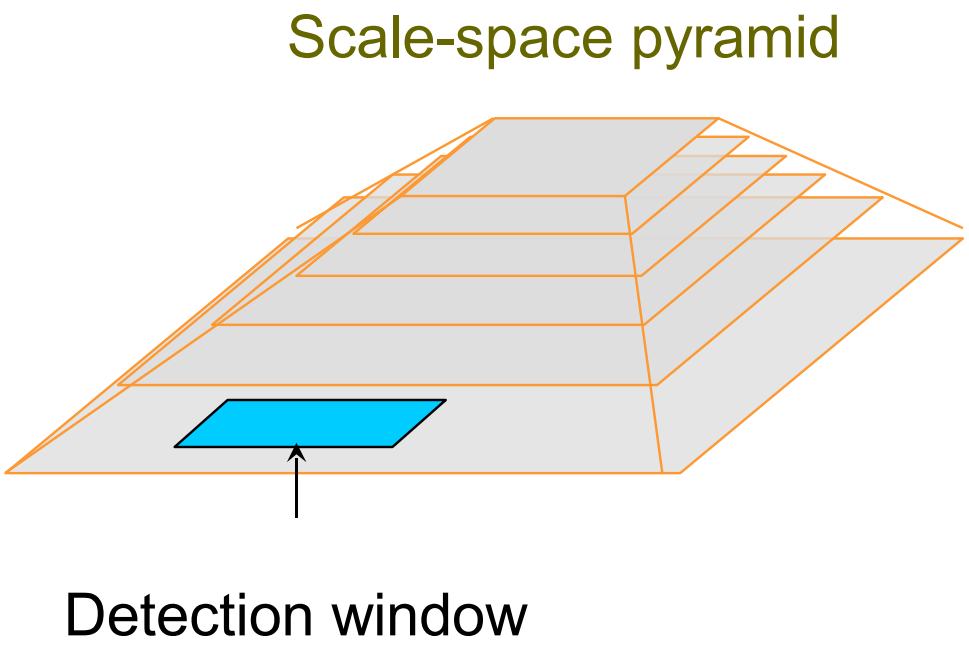


Figure: Illustration of a sliding window, figure courtesy Leron Fliess, Yifat Chernihov and Stav Ashuri

- In a sliding window classifier, one uses a systematic exploration of the image space obtaining individual bounding boxes at each iteration.
- In each iteration, a particular bounding box is evaluated through a classifier. This was done by obtaining the feature vector for the box and using a classifier such as an SVM. This technique was first illustrated using the HoG feature descriptor for pedestrian detection by Dalal and Triggs.

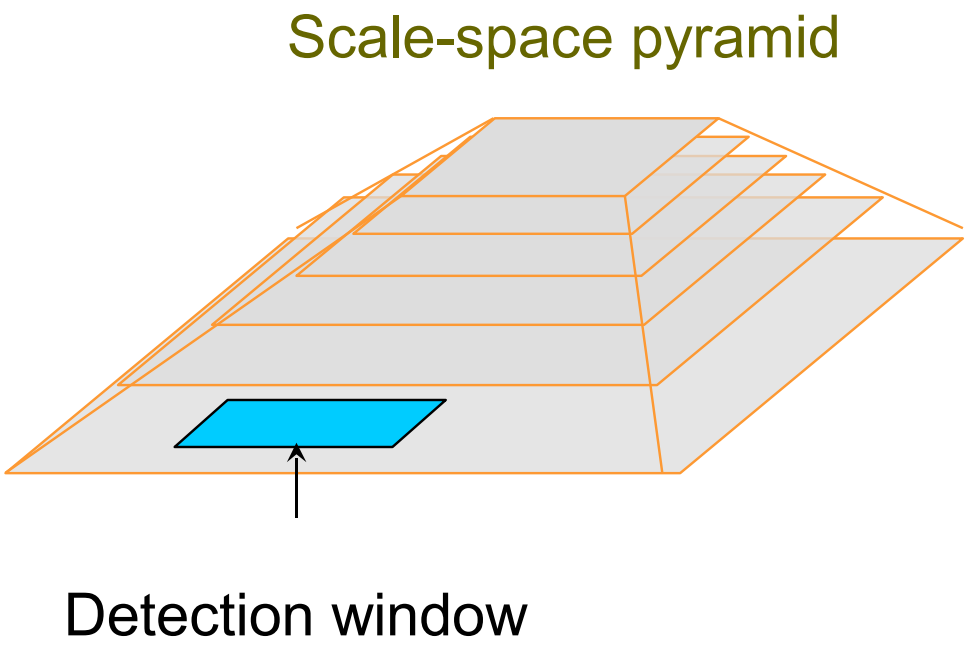
Dalal Triggs algorithm



- Scan image(s) at all scales and locations (with a stride factor and a scale factor)

Figure: Dalal Triggs object detection over a scale space

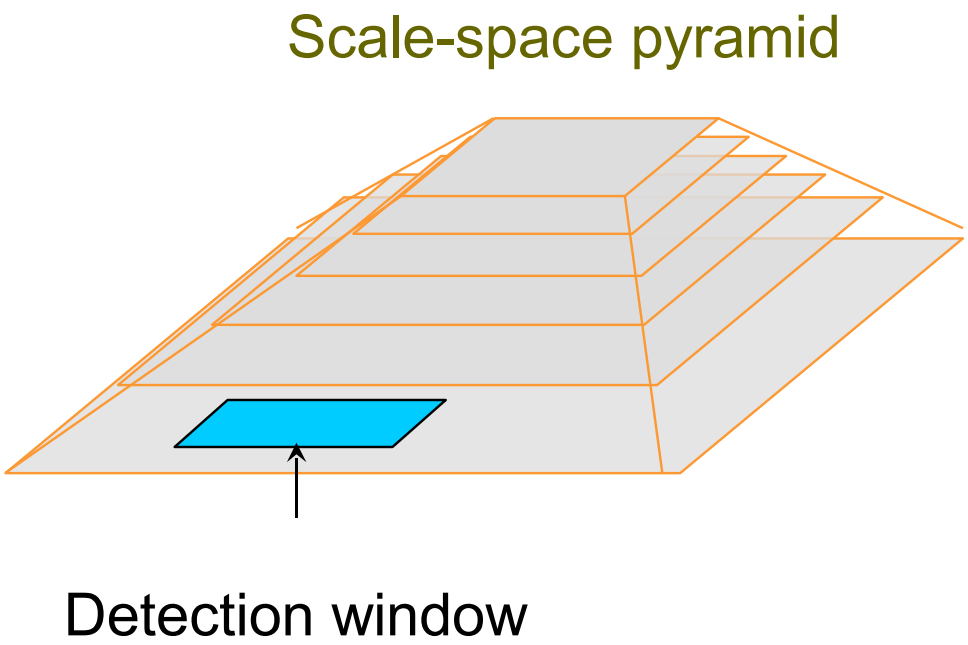
Dalal Triggs algorithm



- Scan image(s) at all scales and locations (with a stride factor and a scale factor)
- Extract features over windows

Figure: Dalal Triggs object detection over a scale space

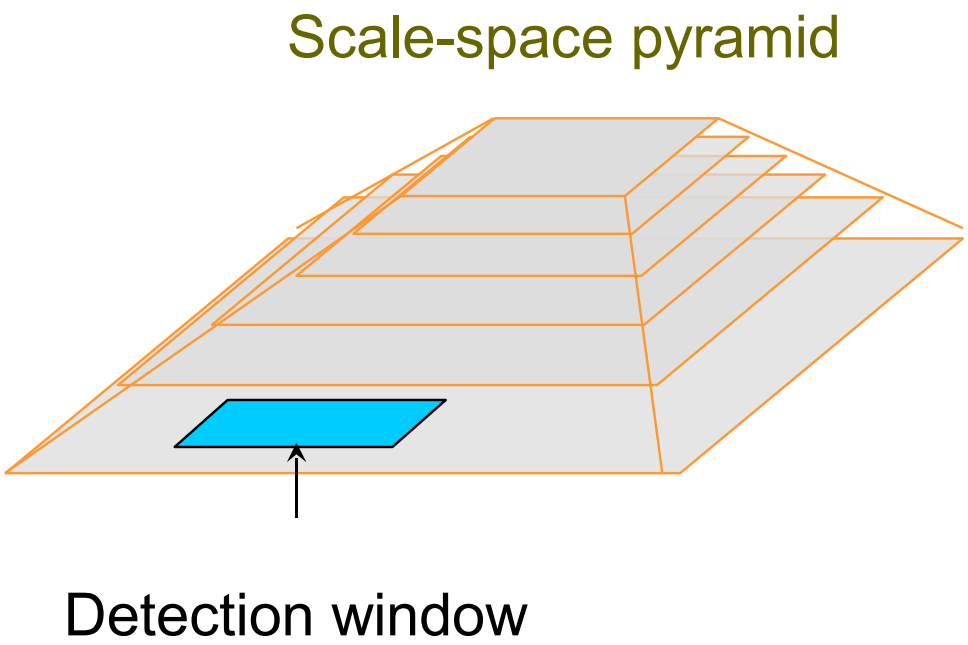
Dalal Triggs algorithm



- Scan image(s) at all scales and locations (with a stride factor and a scale factor)
- Extract features over windows
- Run linear SVM classifier on all windows

Figure: Dalal Triggs object detection over a scale space

Dalal Triggs algorithm



- Scan image(s) at all scales and locations (with a stride factor and a scale factor)
- Extract features over windows
- Run linear SVM classifier on all windows
- Fuse multiple detections in position and scale space

Figure: Dalal Triggs object detection over a scale space

Dalal Triggs algorithm

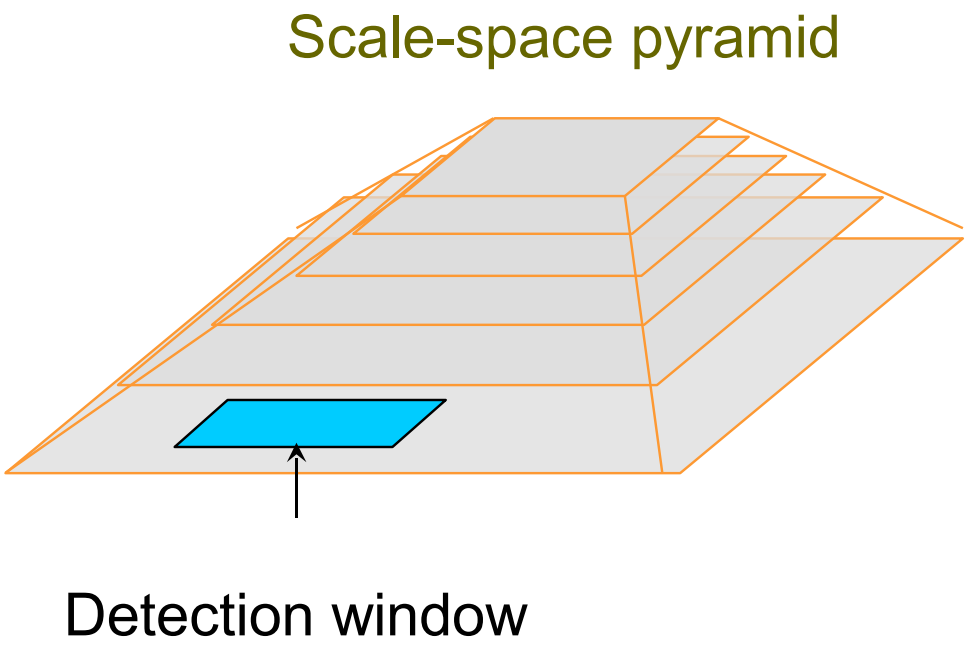


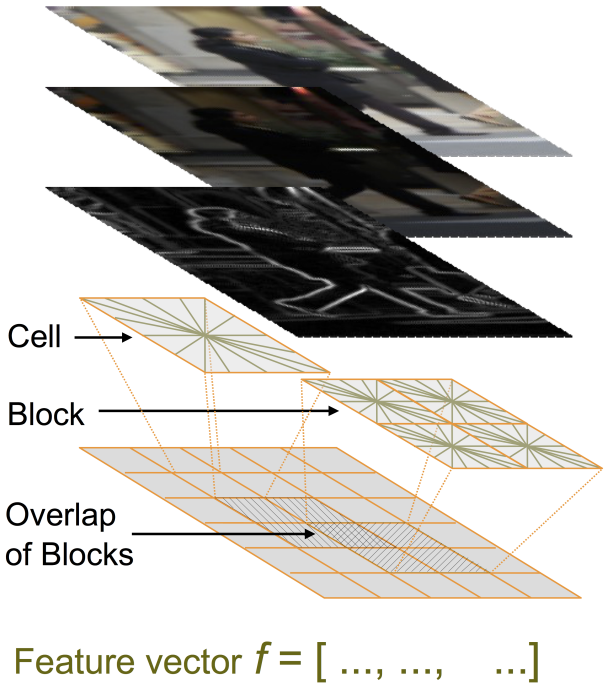
Figure: Dalal Triggs object detection over a scale space

- Scan image(s) at all scales and locations (with a stride factor and a scale factor)
- Extract features over windows
- Run linear SVM classifier on all windows
- Fuse multiple detections in position and scale space
- Output the detections that are obtained after fusion

Outline

Histogram of Gradients (HoG)

The crucial component towards this algorithm is the feature representation used that was the Histogram of Gradients (HoG) by Dalal and Triggs

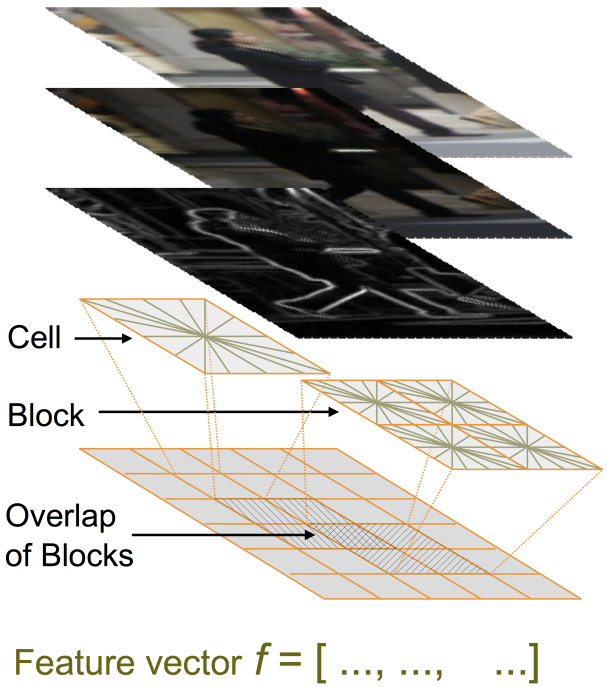


- Given a feature window, the first optional step is to do Gamma Normalisation

Figure: Histogram of Gradients -

Histogram of Gradients (HoG)

The crucial component towards this algorithm is the feature representation used that was the Histogram of Gradients (HoG) by Dalal and Triggs

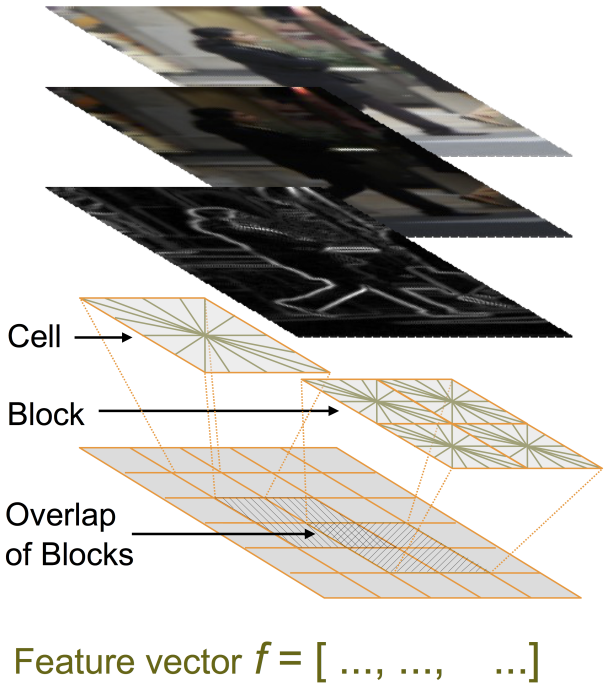


- Given a feature window, the first optional step is to do Gamma Normalisation
- Gradients are computed

Figure: Histogram of Gradients -

Histogram of Gradients (HoG)

The crucial component towards this algorithm is the feature representation used that was the Histogram of Gradients (HoG) by Dalal and Triggs

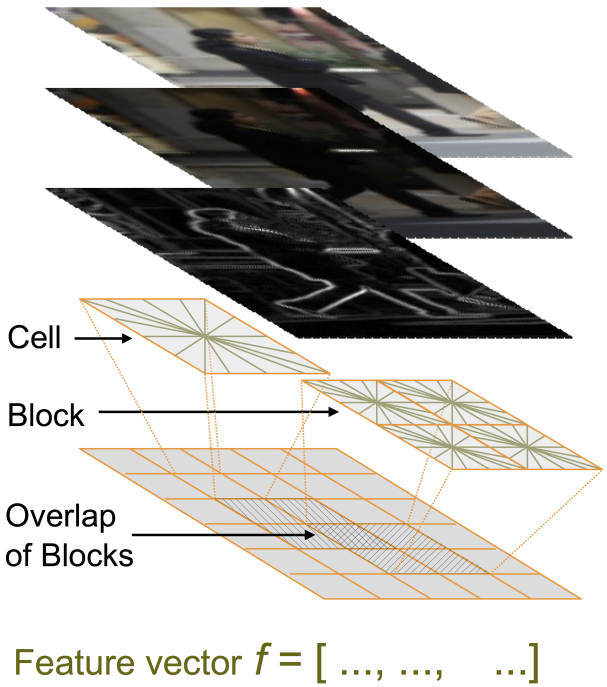


- Given a feature window, the first optional step is to do Gamma Normalisation
- Gradients are computed
- Weighted votes are cast in spatial and orientation cells

Figure: Histogram of Gradients -

Histogram of Gradients (HoG)

The crucial component towards this algorithm is the feature representation used that was the Histogram of Gradients (HoG) by Dalal and Triggs



- Given a feature window, the first optional step is to do Gamma Normalisation
- Gradients are computed
- Weighted votes are cast in spatial and orientation cells
- These cells are aggregated into blocks and normalisation is done for each block

Figure: Histogram of Gradients -

Histogram of Gradients (HoG)

The crucial component towards this algorithm is the feature representation used that was the Histogram of Gradients (HoG) by Dalal and Triggs

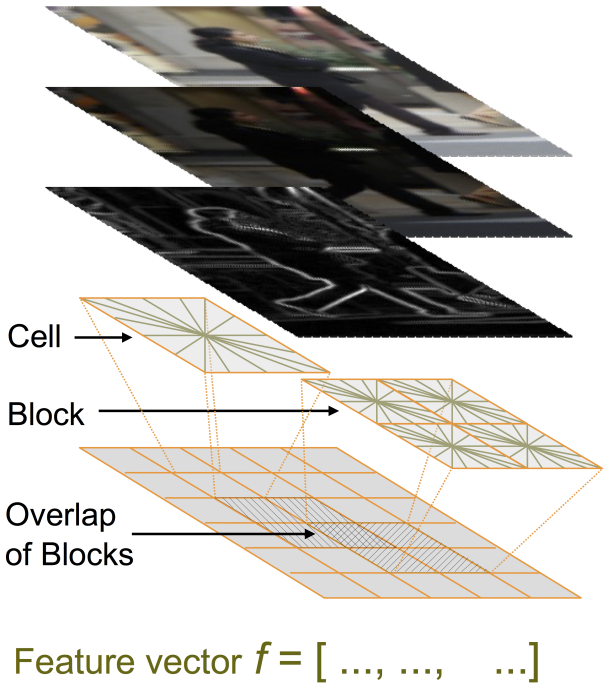


Figure: Histogram of Gradients -

- Given a feature window, the first optional step is to do Gamma Normalisation
- Gradients are computed
- Weighted votes are cast in spatial and orientation cells
- These cells are aggregated into blocks and normalisation is done for each block
- Given a detection window, the HoG are collected through the blocks present in each detection window

Gamma Normalisation

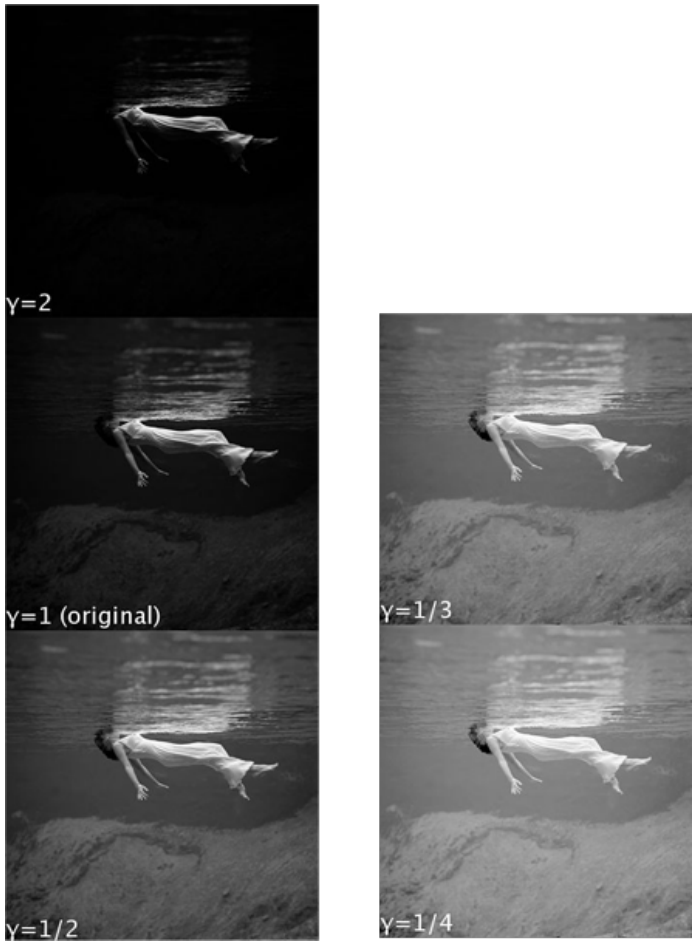


Figure: Gamma Normalisation

- This is an optional step, used to balance the contrast variation. A fixed factor of gamma is used to enhane the distribution of colour.

Gamma Normalisation

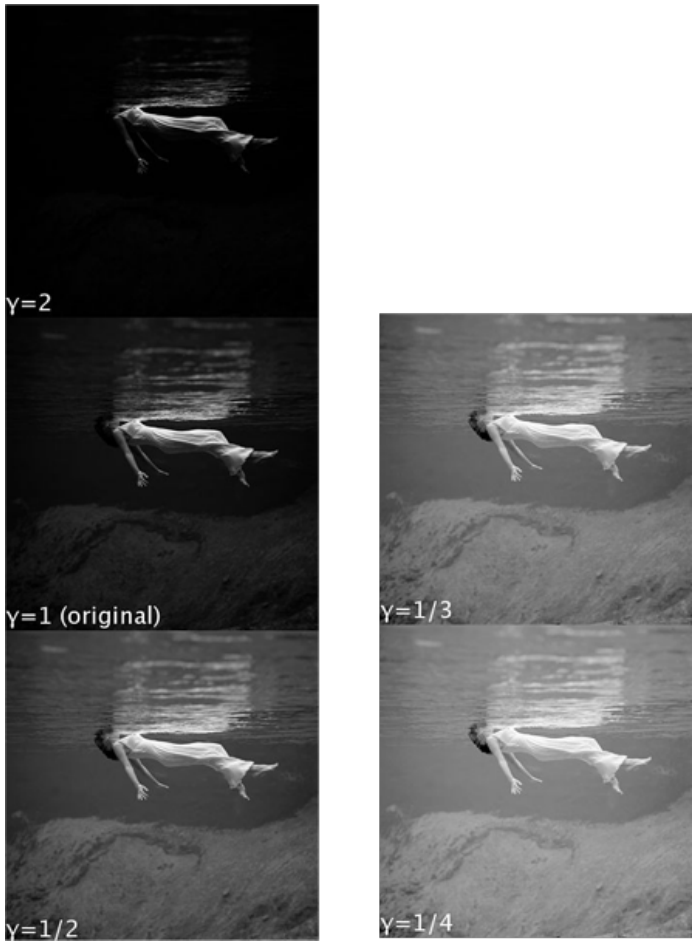


Figure: Gamma Normalisation

- This is an optional step, used to balance the contrast variation. A fixed factor of gamma is used to enhane the distribution of colour.
- The formula used for gamma normalisation is given by

$$V_{\text{out}} = AV_{\text{in}}^{\gamma} \qquad (1)$$

Gamma Normalisation

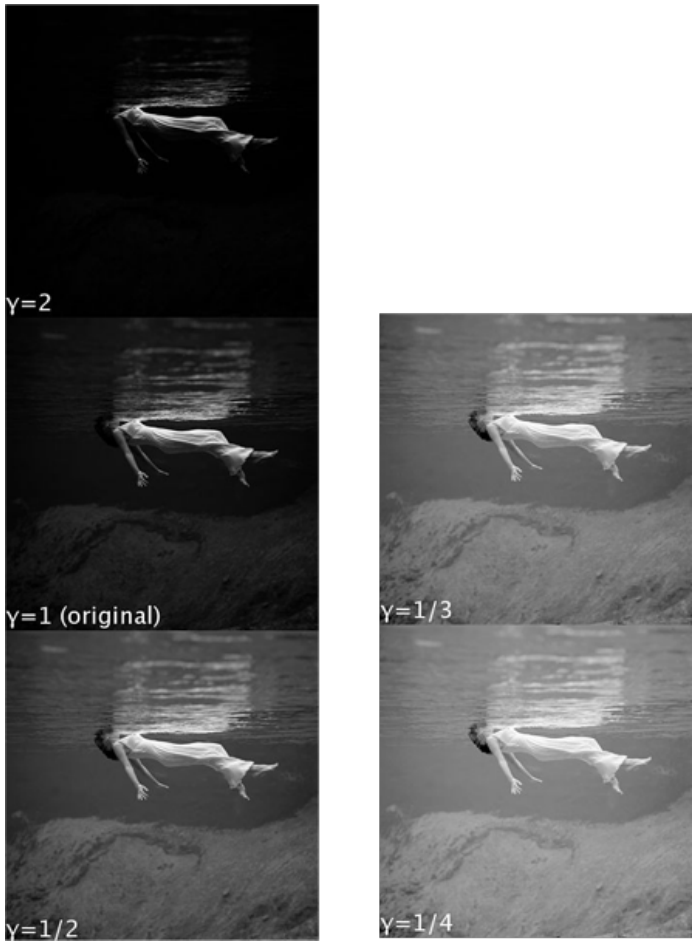


Figure: Gamma Normalisation

- This is an optional step, used to balance the contrast variation. A fixed factor of gamma is used to enhane the distribution of colour.
- The formula used for gamma normalisation is given by

$$V_{\text{out}} = AV_{\text{in}}^{\gamma} \qquad (1)$$

- This step was observed not to significantly affect the results.

Gradient computation

- A number of gradient filters were evaluated by Navneet Dalal in his approach for obtaining gradients.

Gradient computation

- A number of gradient filters were evaluated by Navneet Dalal in his approach for obtaining gradients.
- After evaluation the gradient filters used were given by $[-1, 0, 1]$ and $[-1, 0, 1]^T$

Gradient computation

- A number of gradient filters were evaluated by Navneet Dalal in his approach for obtaining gradients.
- After evaluation the gradient filters used were given by $[-1, 0, 1]$ and $[-1, 0, 1]^T$
- These filters were observed to be better than Sobel and other similar filters.

Gradient computation

- A number of gradient filters were evaluated by Navneet Dalal in his approach for obtaining gradients.
- After evaluation the gradient filters used were given by $[-1, 0, 1]$ and $[-1, 0, 1]^T$
- These filters were observed to be better than Sobel and other similar filters.
- Gaussian smoothing as a preprocessing step was also not found to be necessary

Orientation Binning

- In orientation binning, each pixel calculates a weighted vote for an edge orientation histogram based on the orientation of the gradient centered on it.

Orientation Binning

- In orientation binning, each pixel calculates a weighted vote for an edge orientation histogram based on the orientation of the gradient centered on it.
- The votes are accumulated over local spatial regions called *cells*

Orientation Binning

- In orientation binning, each pixel calculates a weighted vote for an edge orientation histogram based on the orientation of the gradient centered on it.
- The votes are accumulated over local spatial regions called *cells*
- Cells can be either rectangular or radial (sectors in a circle).

Orientation Binning

- In orientation binning, each pixel calculates a weighted vote for an edge orientation histogram based on the orientation of the gradient centered on it.
- The votes are accumulated over local spatial regions called *cells*
- Cells can be either rectangular or radial (sectors in a circle).
- The orientation bins are evenly spaced over 0° - 180° (unsigned gradient) or 0° - 360° signed gradient.

Orientation Binning

- In orientation binning, each pixel calculates a weighted vote for an edge orientation histogram based on the orientation of the gradient centered on it.
- The votes are accumulated over local spatial regions called *cells*
- Cells can be either rectangular or radial (sectors in a circle).
- The orientation bins are evenly spaced over 0° - 180° (unsigned gradient) or 0° - 360° signed gradient.
- Votes are linearly interpolated between neighboring bin centers

Orientation Binning

- In orientation binning, each pixel calculates a weighted vote for an edge orientation histogram based on the orientation of the gradient centered on it.
- The votes are accumulated over local spatial regions called *cells*
- Cells can be either rectangular or radial (sectors in a circle).
- The orientation bins are evenly spaced over 0° - 180° (unsigned gradient) or 0° - 360° signed gradient.
- Votes are linearly interpolated between neighboring bin centers
- The vote is a function of the gradient magnitude such as the magnitude, its square, square-root, or a clipped form of the magnitude. The magnitude itself gave the best results

Descriptor Blocks

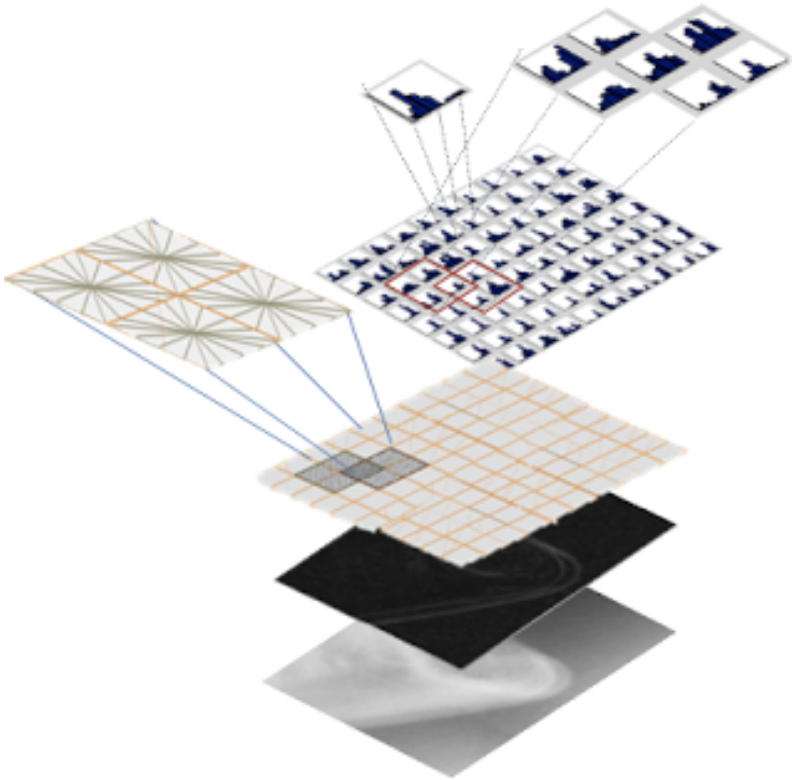


Figure: Descriptor Block

- Cells are accumulated into blocks. Each block accumulates the histograms of multiple cells.

Descriptor Blocks

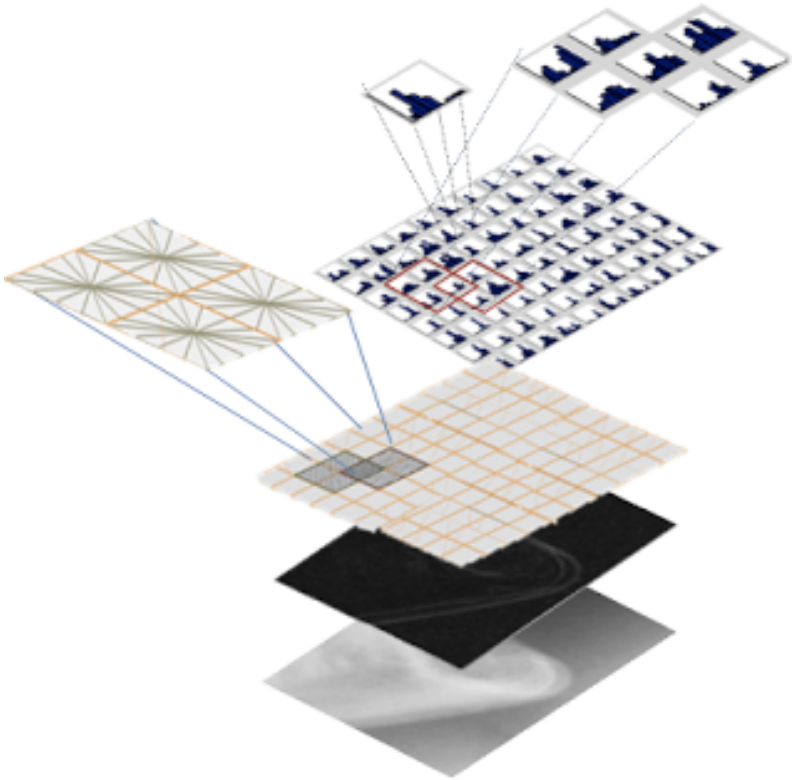


Figure: Descriptor Block

- Cells are accumulated into blocks. Each block accumulates the histograms of multiple cells.
- The blocks are either rectangular blocks R-HoG or circular block HoG C-HoG.

Descriptor Blocks

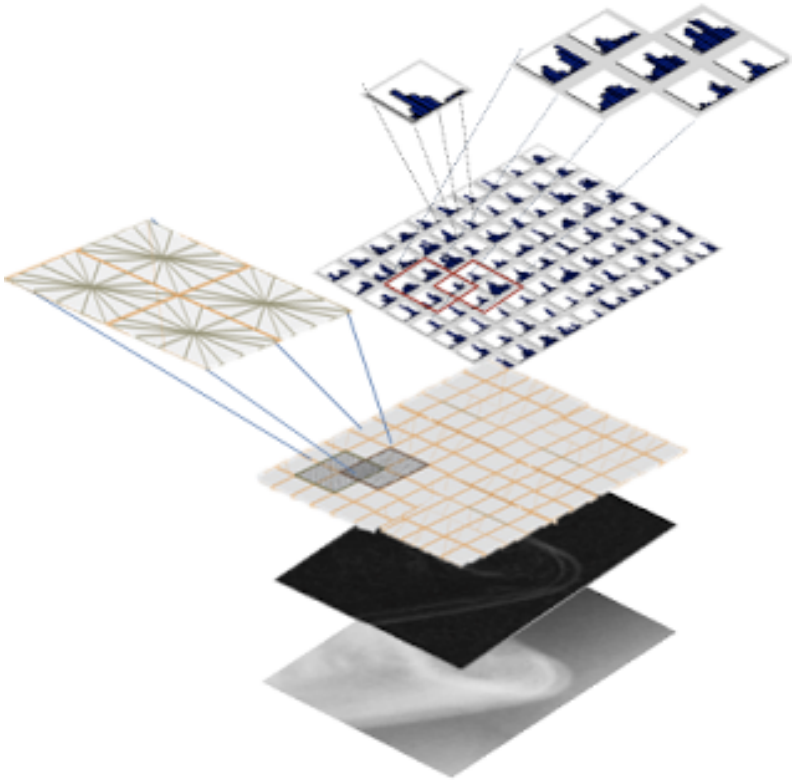


Figure: Descriptor Block

- Cells are accumulated into blocks. Each block accumulates the histograms of multiple cells.
- The blocks are either rectangular blocks R-HoG or circular block HoG C-HoG.
- For humans blocks obtained from 3×3 blocks of 6 pixel cells gave the best performance

Descriptor Blocks

The blocks are normalised using one of the following equations

- L2 norm

$$f = \frac{v}{\sqrt{||v||_2^2 + \epsilon}} \tag{2}$$

Descriptor Blocks

The blocks are normalised using one of the following equations

- L2 norm

$$f = \frac{v}{\sqrt{||v||_2^2 + \epsilon}} \tag{2}$$

- L2 norm - Hys obtained by using above equation but clipping max value v to be 0.2

Descriptor Blocks

The blocks are normalised using one of the following equations

- L2 norm

$$f = \frac{v}{\sqrt{||v||_2^2 + \epsilon}} \tag{2}$$

- L2 norm - Hys obtained by using above equation but clipping max value v to be 0.2
- L1 norm

$$f = \frac{v}{||v||_1 + \epsilon} \tag{3}$$

Descriptor Blocks

The blocks are normalised using one of the following equations

- L2 norm

$$f = \frac{v}{\sqrt{||v||_2^2 + \epsilon}} \tag{2}$$

- L2 norm - Hys obtained by using above equation but clipping max value v to be 0.2
- L1 norm

$$f = \frac{v}{||v||_1 + \epsilon} \tag{3}$$

- L1- sqrt norm

$$f = \frac{v}{\sqrt{||v||_1 + \epsilon}} \tag{4}$$

Descriptor Blocks

The blocks are normalised using one of the following equations

- L2 norm

$$f = \frac{v}{\sqrt{||v||_2^2 + \epsilon}} \tag{2}$$

- L2 norm - Hys obtained by using above equation but clipping max value v to be 0.2
- L1 norm

$$f = \frac{v}{||v||_1 + \epsilon} \tag{3}$$

- L1- sqrt norm

$$f = \frac{v}{\sqrt{||v||_1 + \epsilon}} \tag{4}$$

- Of these L2-norm, L2-Hys, L1-sqrt norm gave equivalent performance whereas L1-norm reduced performance

Evaluation

Parameters

- Gradient scale
- Orientation bins
- Percentage of block overlap

Schemes

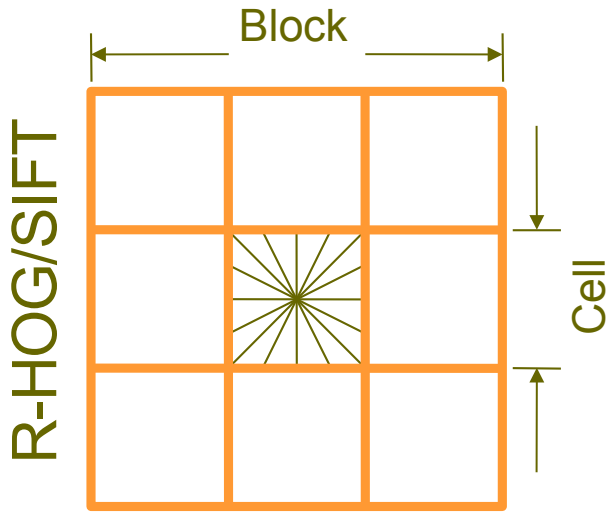
- RGB or Lab, colour/gray-spac
- Block normalisation
  - *L2*-norm,

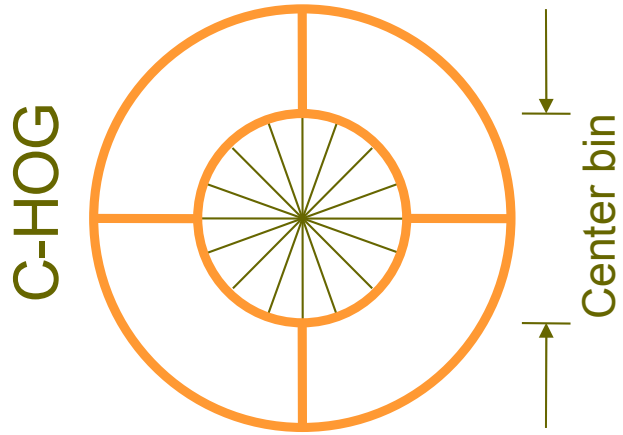
$$v \leftarrow v / \sqrt{\|v\|_2^2 + \varepsilon}$$

or

  - *L1*-norm,

$$v \leftarrow \sqrt{v / (\|v\|_1 + \varepsilon)}$$





Evaluation

Learning phase

Input: Annotations on training images

Create fixed-resolution normalised training image data set

Encode images into feature spaces

Learn binary classifier

Resample negative training images to create hard examples

Encode images into feature spaces

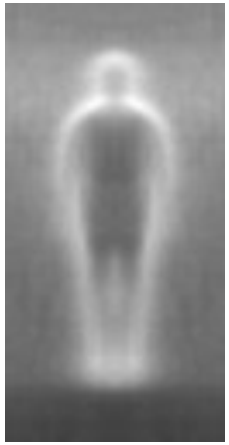
Learn binary classifier

Object/Non-object decision

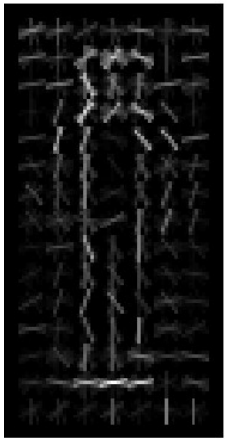
Retraining reduces false positives by an order of magnitude!

Evaluation

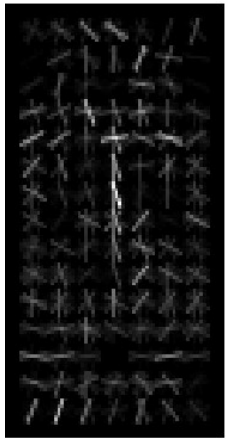
Input example



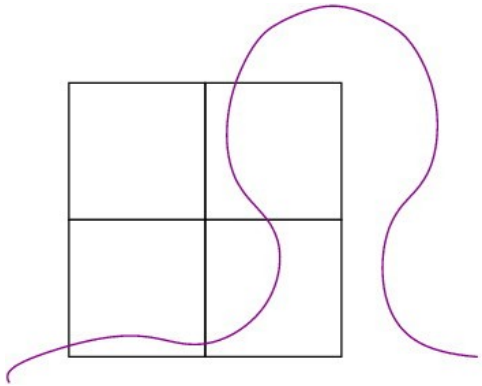
Average gradients



Weighted pos wts



Weighted neg wts



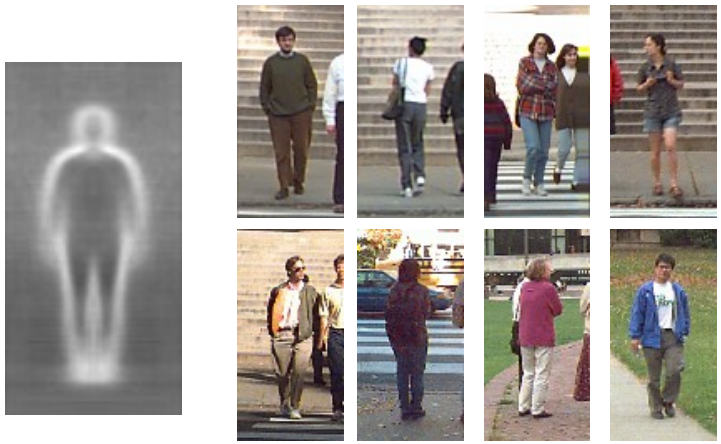
Outside-in weights

Most important cues are head, shoulder, leg silhouettes

Vertical gradients inside a person are counted as negative

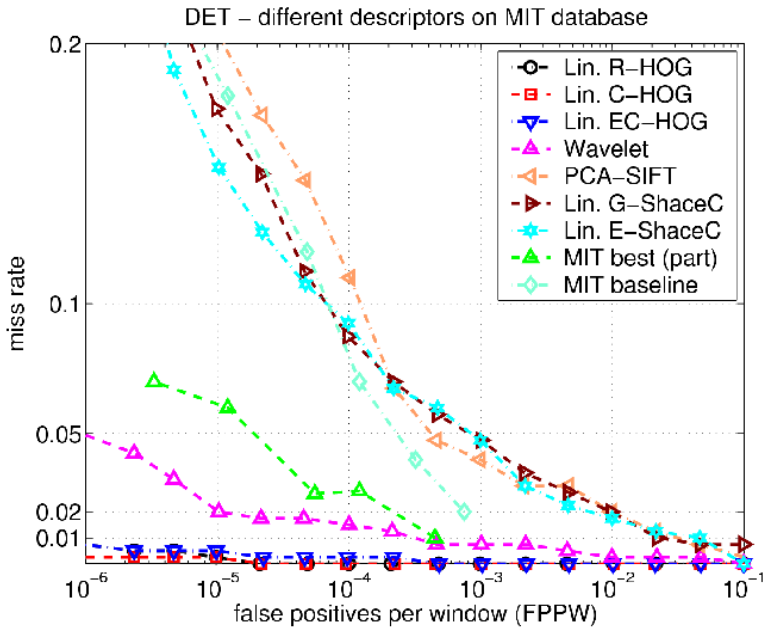
Overlapping blocks just outside the contour are most important

Evaluation

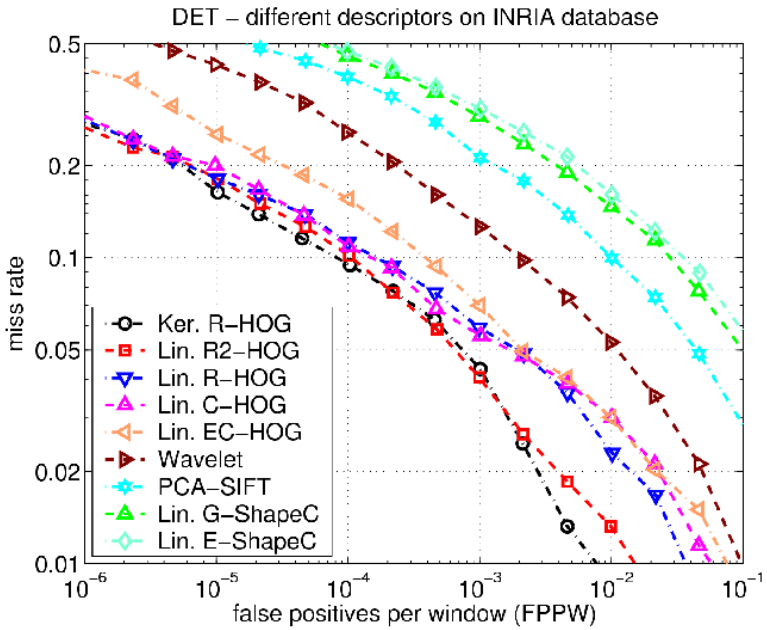
MIT pedestrian database		INRIA person database	
			
Train	507 positive windows Negative data unavailable	Train	1208 positive windows 1218 negative images
Test	200 positive windows Negative data unavailable	Test	566 positive windows 453 negative images
Overall 709 annotations+ reflections		Overall 1774 annotations+ reflections	

Evaluation

MIT pedestrian database



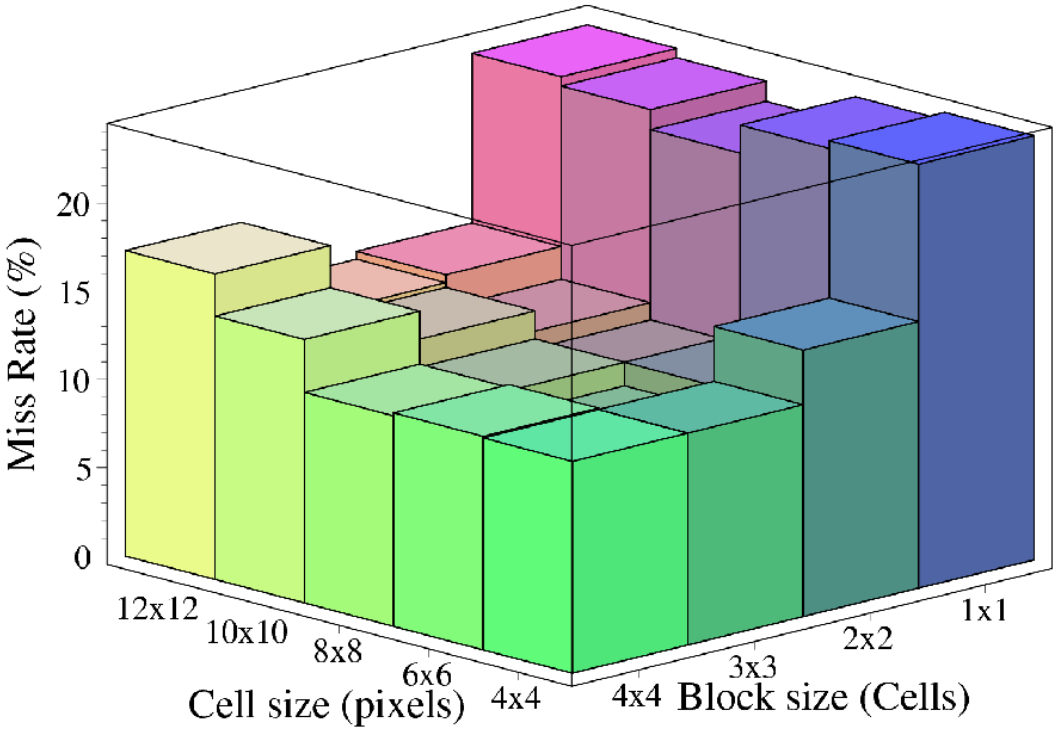
INRIA person database

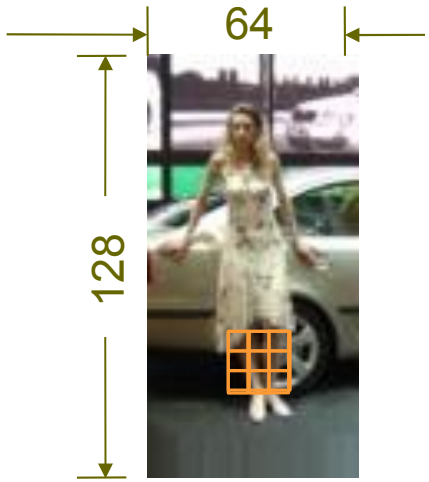


R/C-HOG give near perfect separation on MIT database

Have 1-2 order lower false positives than other descriptors

Evaluation



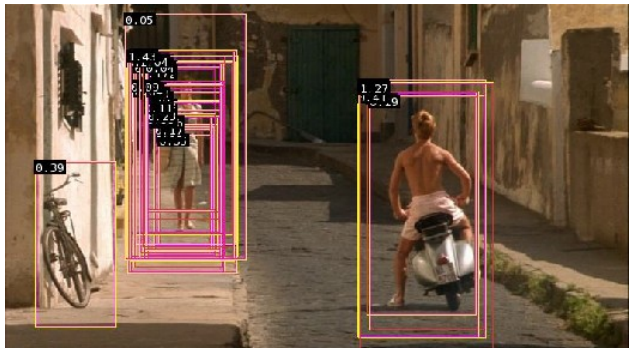


Trade off between need for local spatial invariance and need for finer spatial resolution

Outline

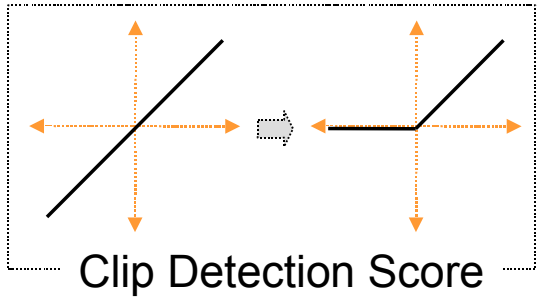
Non Maxima Suprression

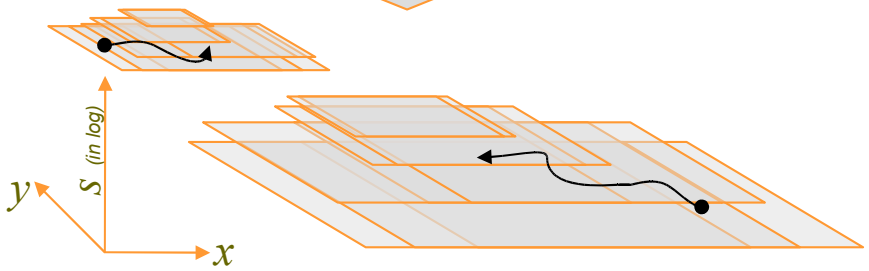
Multi-scale object localisation is obtained through non-maxima suppression

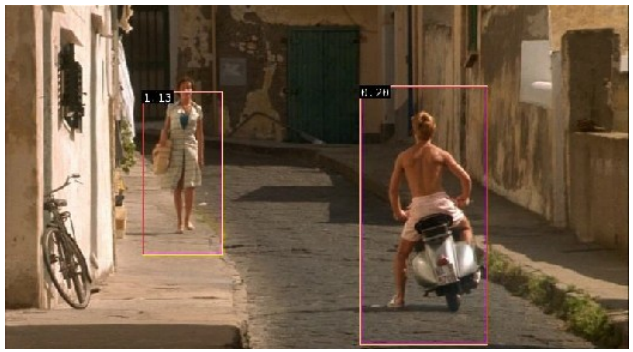


Multi-scale dense scan of detection window

Bias







Final detections

Threshold

$$\mathbf{H}_i = [\exp(s_i)\sigma_x, \exp(s_i)\sigma_y, \sigma_s]$$

$$f(\mathbf{x}) = \sum_i^n w_i \exp\left(-\left\|(\mathbf{x}-\mathbf{x}_i)/\mathbf{H}_i^{-1}\right\|^2/2\right)$$

Apply robust mode detection, like mean shift

Outline

HoG based Methods for Object Detection

To summarize have considered a Histogram of Gradients (HoG) approach by Dalal and Triggs for object detection. We now consider briefly the other approaches that were proposed for solving this problem using HoG based features and their accuracies on Pascal VOC 2007 benchmark that was used for evaluation of object detection.

HoG based Methods for Object Detection

To summarize have considered a Histogram of Gradients (HoG) approach by Dalal and Triggs for object detection. We now consider briefly the other approaches that were proposed for solving this problem using HoG based features and their accuracies on Pascal VOC 2007 benchmark that was used for evaluation of object detection.

- With a linear kernel the HoG based approach obtained an accuracy of 14.7 % mAP over all classes (cf. Harzallah *et al.*, ICCV 2009) and with a χ^2 kernel they obtained an accuracy of 21.9% mAP

Deformable Part Models

- A substantial progress in this approach was made by the Deformable Part Model (DPM) by Felzenszwalb *et al.* (PAMI 2010, CVPR 2008, CVPR 2010, NIPS 2011). In this model parts of an object are discriminatively trained and detected in combination with detecting the whole object.

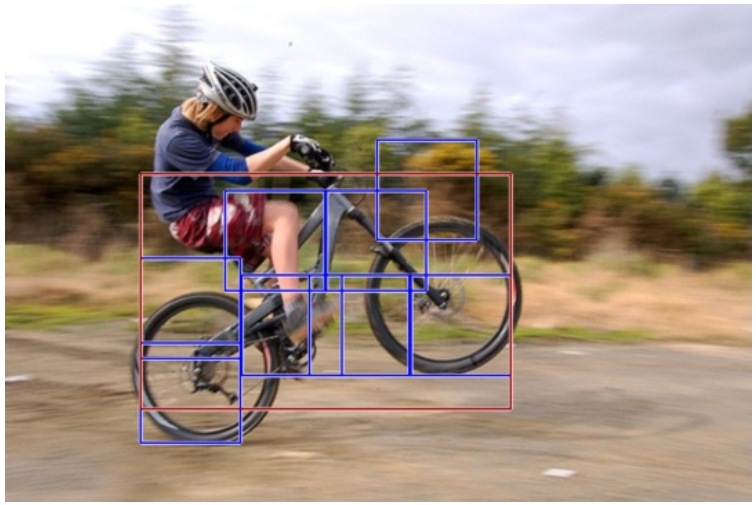


Figure: Deformable Part Model

Deformable Part Models

- A substantial progress in this approach was made by the Deformable Part Model (DPM) by Felzenszwalb *et al.* (PAMI 2010, CVPR 2008, CVPR 2010, NIPS 2011). In this model parts of an object are discriminatively trained and detected in combination with detecting the whole object.
- The best such model when combined with context resulted in a performance of around 35.4% mAP

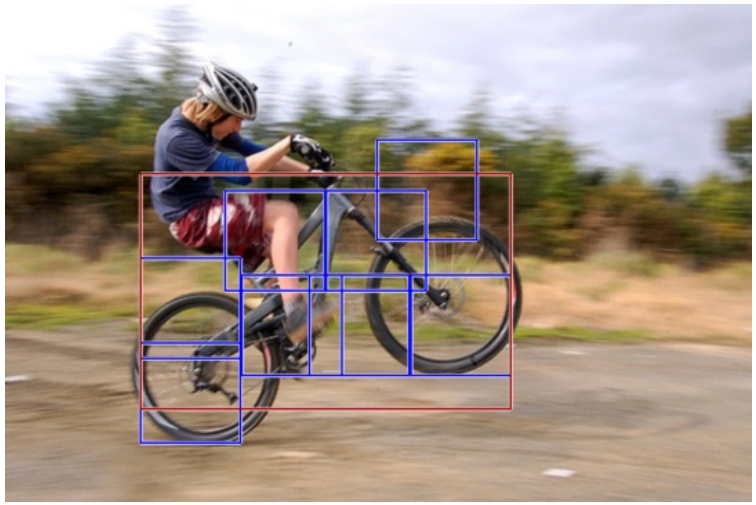


Figure: Deformable Part Model

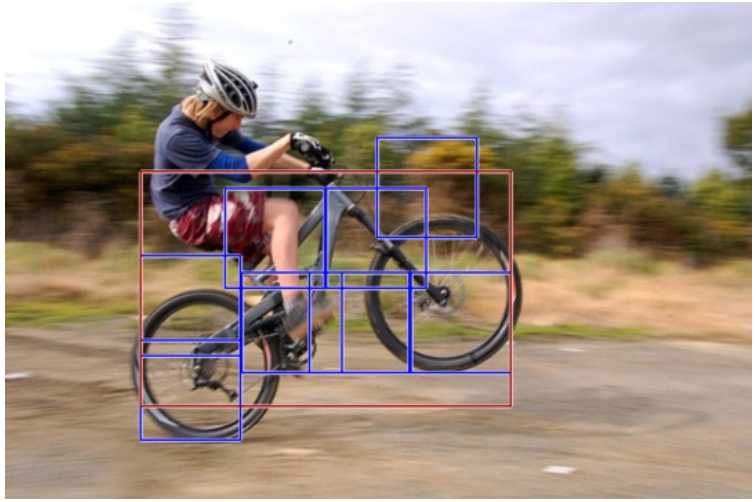
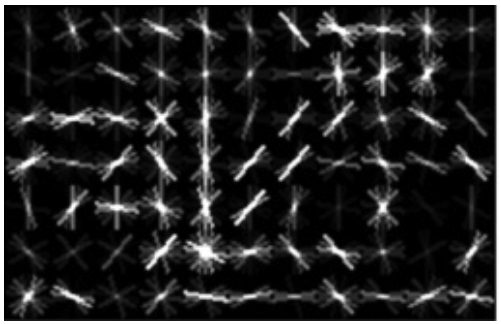
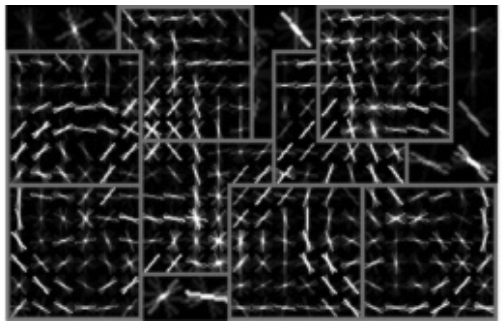


Figure: Deformable Part Model





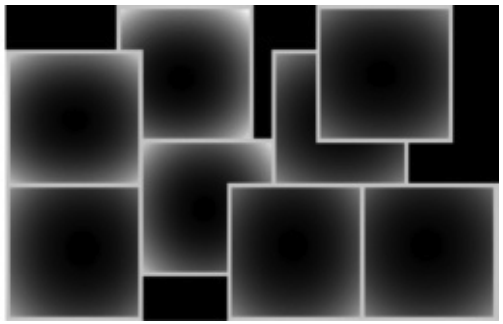


Figure: Deformable Part Model

Outline

Overview of RCNN

- The advent of Alexnet introduced deep learning methods for image classification.

Overview of RCNN

- The advent of Alexnet introduced deep learning methods for image classification.
- However, their use for object detection was not common as object detection methods like DPM worked well with limited data available in Pascal VOC.

Overview of RCNN

- The advent of Alexnet introduced deep learning methods for image classification.
- However, their use for object detection was not common as object detection methods like DPM worked well with limited data available in Pascal VOC.
- Region based CNN was the first such method that showed that deep learning techniques could be successfully used for object detection as well.

Overview of RCNN

- The advent of Alexnet introduced deep learning methods for image classification.
- However, their use for object detection was not common as object detection methods like DPM worked well with limited data available in Pascal VOC.
- Region based CNN was the first such method that showed that deep learning techniques could be successfully used for object detection as well.
- The method still used deep learning more as a means for obtaining better features that were then used by an SVM for classification

Overview of RCNN

- The advent of Alexnet introduced deep learning methods for image classification.
- However, their use for object detection was not common as object detection methods like DPM worked well with limited data available in Pascal VOC.
- Region based CNN was the first such method that showed that deep learning techniques could be successfully used for object detection as well.
- The method still used deep learning more as a means for obtaining better features that were then used by an SVM for classification
- An additional contribution of this method is it advocated the use of classification and regression for solving object detection

Overview of RCNN

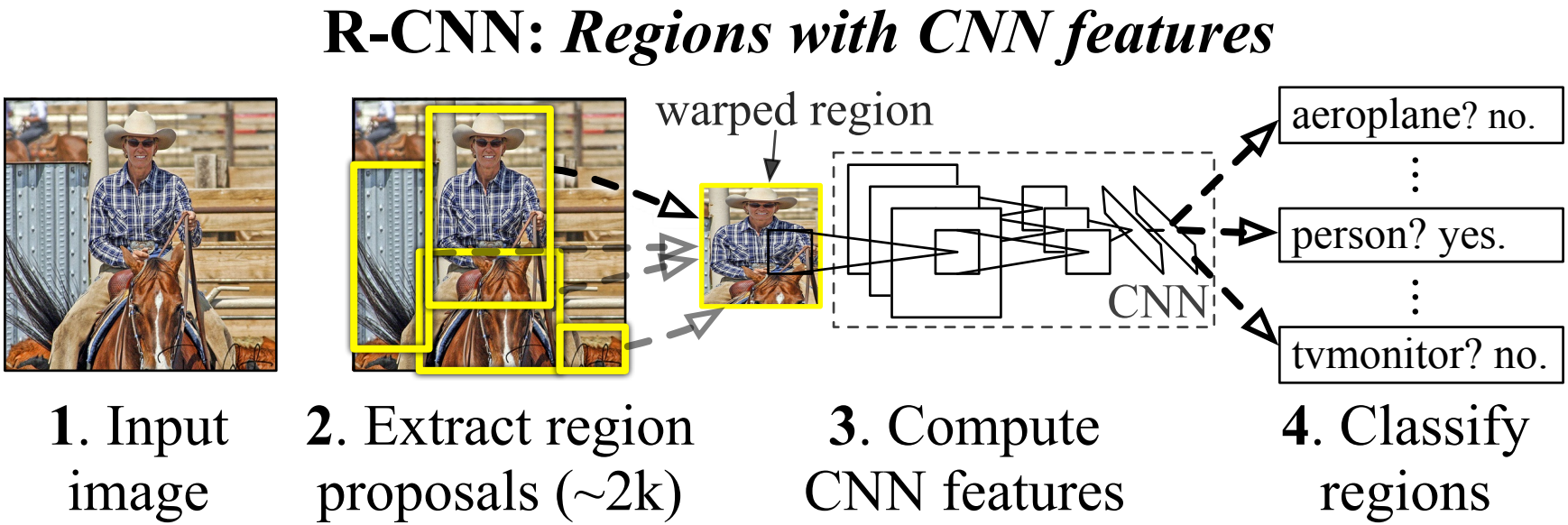


Figure: An overview of the RCNN detection pipeline

Overview of RCNN

- RCNN advocated an approach that uses region proposals extracted through an unsupervised technique that provided proposals that were likely to contain an object.

Overview of RCNN

- RCNN advocated an approach that uses region proposals extracted through an unsupervised technique that provided proposals that were likely to contain an object.
- These proposals were warped to obtain regular sized regions

Overview of RCNN

- RCNN advocated an approach that uses region proposals extracted through an unsupervised technique that provided proposals that were likely to contain an object.
- These proposals were warped to obtain regular sized regions
- These regions were encoded using a fine-tuned CNN network (AlexNet)

Overview of RCNN

- RCNN advocated an approach that uses region proposals extracted through an unsupervised technique that provided proposals that were likely to contain an object.
- These proposals were warped to obtain regular sized regions
- These regions were encoded using a fine-tuned CNN network (AlexNet)
- These features were then classified using a 21 class classifier into one of the 20 Pascal VOC classes and a background class

The End