Robust Detection in Presence of Hard Examples

Subhabrata Debnath

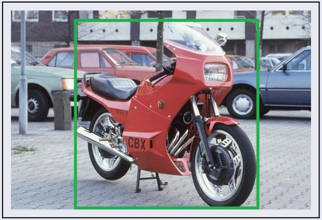
Co-founder & Computer Vision Researcher
VisageMap Inc.
subhabrata.debnath@visagemap.com
s.debnath1989@gmail.com

March 11, 2016

Outline

Object Detection

What is Detection ?



- Detecting an object in an image involves predicting the location of the bounding box containing it, if it is present.

Outline

1. Problem Statement
   - Training a Detector
   - Hard Examples
   - Aim
2. Background
3. Method Description
4. Results
5. Conclusion and Future Scope

Training a Detector

Weakly supervised setting

- Set of images as input, where each image has an associated label.

Training a Detector

Weakly supervised setting

- Set of images as input, where each image has an associated label.
- Labels only denote the presence or absence of the object of interest.

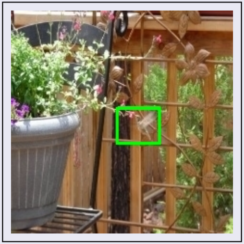
Training a Detector

Weakly supervised setting

- Set of images as input, where each image has an associated label.
- Labels only denote the presence or absence of the object of interest.
- No explicit information about location of the object in the image.

Outline

Problem with hard examples ?



- Including such examples in the training data naively may deteriorate the performance of the classifier, as these hardly have any structural resemblance to actual positives.

Outline

What is our input ?

- A set of images with weak supervision, where some examples are hard.

What is our input ?

- A set of images with weak supervision, where some examples are hard.

What do we want to achieve ?

- Some measure of the hardness for each training example.

What is our input ?

- A set of images with weak supervision, where some examples are hard.

What do we want to achieve ?

- Some measure of the hardness for each training example.
- Careful training using the hardness information.

What is our input ?

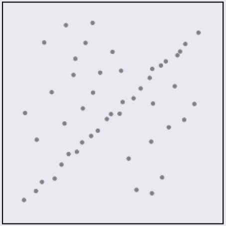
- A set of images with weak supervision, where some examples are hard.

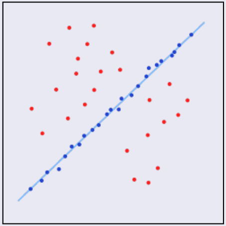
What do we want to achieve ?

- Some measure of the hardness for each training example.
- Careful training using the hardness information.
- Ultimately, building a detector model which is robust to hard examples.

Outline

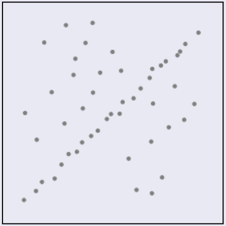
RANDOM SAMPLE CONSENSUS, Fischler and Bolles, 1981

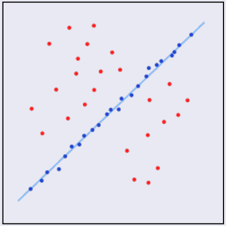




- Learning technique to estimate model parameters by random sampling of observed data.

RANDOM SAMPLE CONSENSUS, Fischler and Bolles, 1981





- Learning technique to estimate model parameters by random sampling of observed data.
- Highly robust to outliers.

RANSAC Algorithm : Fischler and Bolles, SRI 1981

1. Select a random subset of the original data called probable inliers.

RANSAC Algorithm : Fischler and Bolles, SRI 1981

1. Select a random subset of the original data called probable inliers.
2. Build a model using the above inliers.

RANSAC Algorithm : Fischler and Bolles, SRI 1981

1. Select a random subset of the original data called probable inliers.
2. Build a model using the above inliers.
3. Test the rest of the data using the model above.

RANSAC Algorithm : Fischler and Bolles, SRI 1981

1. Select a random subset of the original data called probable inliers.
2. Build a model using the above inliers.
3. Test the rest of the data using the model above.
4. If majority of the data agree with the model then accept it, else reject and repeat from 1 to 4.

RANSAC Algorithm : Fischler and Bolles, SRI 1981

1. Select a random subset of the original data called probable inliers.
2. Build a model using the above inliers.
3. Test the rest of the data using the model above.
4. If majority of the data agree with the model then accept it, else reject and repeat from 1 to 4.
5. Rebuild model using all accepted data points.

Outline

RANSAC SVM : Nishida and Kurita, ICPR 2008

1. Select a random subset of the original data called probable inliers.

RANSAC SVM : Nishida and Kurita, ICPR 2008

1. Select a random subset of the original data called probable inliers.
2. Build a SVM classifier model using the above inliers.

RANSAC SVM : Nishida and Kurita, ICPR 2008

1. Select a random subset of the original data called probable inliers.
2. Build a SVM classifier model using the above inliers.
3. Classify the rest of the data using the model above.

RANSAC SVM : Nishida and Kurita, ICPR 2008

1. Select a random subset of the original data called probable inliers.
2. Build a SVM classifier model using the above inliers.
3. Classify the rest of the data using the model above.
4. If majority of the data get properly classified by the model then accept it, else reject and repeat from 1 to 4.

RANSAC SVM : Nishida and Kurita, ICPR 2008

1. Select a random subset of the original data called probable inliers.
2. Build a SVM classifier model using the above inliers.
3. Classify the rest of the data using the model above.
4. If majority of the data get properly classified by the model then accept it, else reject and repeat from 1 to 4.
5. Rebuild model using all accepted examples.

Thus, RANSAC SVM

- Tries to find the "best model" which agrees with majority of the training data.

Thus, RANSAC SVM

- Tries to find the "best model" which agrees with majority of the training data.
- Examples being misclassified by the "best model" can be considered as outliers.

Thus, RANSAC SVM

- Tries to find the "best model" which agrees with majority of the training data.
- Examples being misclassified by the "best model" can be considered as outliers.
- Thus uses the score of just one model to decide the set of outliers.

Outline

Outlier Robust SVM : Subhabrata Debnath, Anjan Banerjee, Vinay Namboodiri, BMVC 2015

1. Select a random subset of the original data called probable inliers.

Outlier Robust SVM : Subhabrata Debnath, Anjan Banerjee, Vinay Namboodiri, BMVC 2015

1. Select a random subset of the original data called probable inliers.
2. Build a SVM classifier model using the above inliers.

Outlier Robust SVM : Subhabrata Debnath, Anjan Banerjee, Vinay Namboodiri, BMVC 2015

1. Select a random subset of the original data called probable inliers.
2. Build a SVM classifier model using the above inliers.
3. Classify the rest of the data using the model above.

Outlier Robust SVM : Subhabrata Debnath, Anjan Banerjee, Vinay Namboodiri, BMVC 2015

1. Select a random subset of the original data called probable inliers.
2. Build a SVM classifier model using the above inliers.
3. Classify the rest of the data using the model above.
4. Increment the misclassification score of each misclassified example.

Outlier Robust SVM : Subhabrata Debnath, Anjan Banerjee, Vinay Namboodiri, BMVC 2015

1. Select a random subset of the original data called probable inliers.
2. Build a SVM classifier model using the above inliers.
3. Classify the rest of the data using the model above.
4. Increment the misclassification score of each misclassified example.
5. Repeat 1 to 5 enough number of times.

Outlier Robust SVM : Subhabrata Debnath, Anjan Banerjee, Vinay Namboodiri, BMVC 2015

1. Select a random subset of the original data called probable inliers.
2. Build a SVM classifier model using the above inliers.
3. Classify the rest of the data using the model above.
4. Increment the misclassification score of each misclassified example.
5. Repeat 1 to 5 enough number of times.
6. Use the examples with the smallest number of misclassifications to build a SVM classifier model.

Outlier Robust SVM : Subhabrata Debnath, Anjan Banerjee, Vinay Namboodiri, BMVC 2015

1. Select a random subset of the original data called probable inliers.
2. Build a SVM classifier model using the above inliers.
3. Classify the rest of the data using the model above.
4. Increment the misclassification score of each misclassified example.
5. Repeat 1 to 5 enough number of times.
6. Use the examples with the smallest number of misclassifications to build a SVM classifier model.
7. Classify all of the training data again.

Outlier Robust SVM : Subhabrata Debnath, Anjan Banerjee, Vinay Namboodiri, BMVC 2015

1. Select a random subset of the original data called probable inliers.
2. Build a SVM classifier model using the above inliers.
3. Classify the rest of the data using the model above.
4. Increment the misclassification score of each misclassified example.
5. Repeat 1 to 5 enough number of times.
6. Use the examples with the smallest number of misclassifications to build a SVM classifier model.
7. Classify all of the training data again.
8. Declare all the misclassified examples as outliers.

Thus, Outlier Robust SVM

- Tries to find the "best subset" which agrees with majority of the training data.

Thus, Outlier Robust SVM

- Tries to find the "best subset" which agrees with majority of the training data.
- Uses misclassification as a parameter to approximate outlierness.

Thus, Outlier Robust SVM

- Tries to find the "best subset" which agrees with majority of the training data.
- Uses misclassification as a parameter to approximate outlierness.
- Thus uses the score of many small models to decide the set of outliers.

Thus, Outlier Robust SVM

- Tries to find the "best subset" which agrees with majority of the training data.
- Uses misclassification as a parameter to approximate outlierness.
- Thus uses the score of many small models to decide the set of outliers.
- Finally, exploits the property that hard examples behave like outliers as they differ in their feature space as compared to ordinary examples.

Outline

Latent SVM : Yu, Joachims, ICML 2009

In a typical latent svm framework, the model parameter w is learnt by solving the following optimization problem:

$$\min_{w,\xi_i \geq 0} \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i \tag{1}$$

s.t.

$$\max_{h_i \in H} w^T \Phi(x_i, y_i, h_i) - \max_{\hat{h}_i \in H, \hat{y}_i \in Y} w^T \Phi(x_i, \hat{y}_i, \hat{h}_i) \geq \Delta(y_i, \hat{y}_i) - \xi_i,$$

$$\forall \hat{y}_i \in Y, \forall \hat{h}_i \in H, i = 1, \ldots, n.$$

Classification rule :

$$\arg\max_{y,h} \langle w, \phi(x_i, y, h) \rangle$$

Latent SVM : Aim

- To learn a bounding box detector using image features and class labels

Latent SVM : Aim

- To learn a bounding box detector using image features and class labels

Latent SVM : Basic steps

1. Using the training image and labels learn an initial model w.

Latent SVM : Aim

- To learn a bounding box detector using image features and class labels

Latent SVM : Basic steps

1. Using the training image and labels learn an initial model w.
2. For each image, find the highest scoring bounding box using current w.

Latent SVM : Aim

- To learn a bounding box detector using image features and class labels

Latent SVM : Basic steps

1. Using the training image and labels learn an initial model w.
2. For each image, find the highest scoring bounding box using current w.
3. Using these bounding boxes re-learn model w.

Latent SVM : Aim

- To learn a bounding box detector using image features and class labels

Latent SVM : Basic steps

1. Using the training image and labels learn an initial model w.
2. For each image, find the highest scoring bounding box using current w.
3. Using these bounding boxes re-learn model w.
4. Repeat 2 to 4 till desired precision.

Latent SVM : Aim

- To learn a bounding box detector using image features and class labels

Latent SVM : Basic steps

1. Using the training image and labels learn an initial model w.
2. For each image, find the highest scoring bounding box using current w.
3. Using these bounding boxes re-learn model w.
4. Repeat 2 to 4 till desired precision.

Latent SVM : Test

- Find the highest scoring bounding box in the test image.

Latent SVM : Aim

- To learn a bounding box detector using image features and class labels

Latent SVM : Basic steps

1. Using the training image and labels learn an initial model w.
2. For each image, find the highest scoring bounding box using current w.
3. Using these bounding boxes re-learn model w.
4. Repeat 2 to 4 till desired precision.

Latent SVM : Test

- Find the highest scoring bounding box in the test image.
- If score is >threshold, output the bounding box as positive or output negative.

Latent SVM : Yu, Joachims, ICML 2009

- For a particular w, the value of ξ_i is an upper bound on the loss $\Delta(y_i, \hat{y}_i)$.
- Equation 1 is basically minimizing the difference of two convex functions or equivalently minimizing a concave-convex sum.
- Can be solved by the Concave Convex Procedure

The Concave Convex Procedure : Yuille and Rangarajan, NIPS 2002

- Solves the optimization :

$$\min_{w,\xi_i \geq 0} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}[\max_{\hat{h}_i \in H, \hat{y}_i \in Y}(w^T\Phi(x_i, \hat{y}_i, \hat{h}_i) + \Delta(y_i, \hat{y}_i))$$

$$- \max_{h_i \in H} w^T\Phi(x_i, y_i, h_i)] (2)$$

- Assumes an initial value of the model parameter.

The Concave Convex Procedure : Yuille and Rangarajan, NIPS 2002

- Solves the optimization :

$$\min_{w,\xi_i \geq 0} \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} [\max_{\hat{h}_i \in H, \hat{y}_i \in Y} (w^T \Phi(x_i, \hat{y}_i, \hat{h}_i) + \Delta(y_i, \hat{y}_i))$$

$$- \max_{h_i \in H} w^T \Phi(x_i, y_i, h_i)] (2)$$

- Assumes an initial value of the model parameter.
- Solves for the model and latent parameters alternatively by fixing the other.

The Concave Convex Procedure : Yuille and Rangarajan, NIPS 2002

- Solves the optimization :

$$\min_{w,\xi_i \geq 0} \frac{1}{2}\|w\|^2 + \frac{C}{n}\sum_{i=1}^{n}[\max_{\hat{h}_i \in H, \hat{y}_i \in Y}(w^T\Phi(x_i,\hat{y}_i,\hat{h}_i) + \Delta(y_i,\hat{y}_i))$$

$$-\max_{h_i \in H} w^T\Phi(x_i,y_i,h_i)] \quad (2)$$

- Assumes an initial value of the model parameter.
- Solves for the model and latent parameters alternatively by fixing the other.
- High dependence on initialization of latent variables.

Outline

Better initialization ?

- Problem

Better initialization ?

- Problem
  - Due to presence of hard examples in data, initialization becomes even more important.

Better initialization ?

- Problem
  - Due to presence of hard examples in data, initialization becomes even more important.
- Solution

Better initialization ?

- Problem
  - Due to presence of hard examples in data, initialization becomes even more important.
- Solution
  - Exclude outliers declared by Outlier Robust SVM from model initialization.

Better initialization ?

- Problem
  - Due to presence of hard examples in data, initialization becomes even more important.
- Solution
  - Exclude outliers declared by Outlier Robust SVM from model initialization.
- Can we improve any further ?

Better initialization ?

- Problem
  - Due to presence of hard examples in data, initialization becomes even more important.
- Solution
  - Exclude outliers declared by Outlier Robust SVM from model initialization.
- Can we improve any further ?
  - Impose an ordering on training.

Self Paced Learning : P. Kumar, B. Packer, and D. Koller, NIPS 2010

- Teach easy examples first, followed by harder examples gradually.

Self Paced Learning : P. Kumar, B. Packer, and D. Koller, NIPS 2010

- Teach easy examples first, followed by harder examples gradually.
- Easiness directly proportional to distance from the hyperplane.

Self Paced Learning : P. Kumar, B. Packer, and D. Koller, NIPS 2010

- Teach easy examples first, followed by harder examples gradually.
- Easiness directly proportional to distance from the hyperplane.
- New objective function :

$$\min_{w,\xi_i \geq 0, v \in \{0,1\}} \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} vi\xi_i - \frac{1}{K} \sum_{i=1}^{n} v_i \tag{3}$$

- Variables v_i indicate whether i^{th} sample is easy or not.

Self Paced Learning : P. Kumar, B. Packer, and D. Koller, NIPS 2010

- Teach easy examples first, followed by harder examples gradually.
- Easiness directly proportional to distance from the hyperplane.
- New objective function :

$$\min_{w,\xi_i \geq 0, v \in \{0,1\}} \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} vi\xi_i - \frac{1}{K} \sum_{i=1}^{n} v_i \tag{3}$$

- Variables v_i indicate whether i^{th} sample is easy or not.
- Solved using alternate convex search.

Self Paced Learning : P. Kumar, B. Packer, and D. Koller, NIPS 2010

- Teach easy examples first, followed by harder examples gradually.
- Easiness directly proportional to distance from the hyperplane.
- New objective function :

$$\min_{w,\xi_i \geq 0, v \in \{0,1\}} \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^{n} v_i \xi_i - \frac{1}{K} \sum_{i=1}^{n} v_i \tag{3}$$

- Variables v_i indicate whether i^{th} sample is easy or not.
- Solved using alternate convex search.
- Considers all samples in the final iteration, thus provides the same guarantees as CCCP.

Thus so far !

Thus so far !

- Initialize latent parameter using non-outliers from Outlier Robust SVM.

Thus so far !

- Initialize latent parameter using non-outliers from Outlier Robust SVM.
- Solve the optimization using Self Paced Learning.

Why initialization helps ?

Thus so far !

- Initialize latent parameter using non-outliers from Outlier Robust SVM.
- Solve the optimization using Self Paced Learning.

Why initialization helps ?

- As self paced learning distance from the hyperplane as a measure of easiness, thus initial approximation may become skewed.

Outline

Revisiting the constraints of latent SVM

$$\max_{h_i \in H} w^T \Phi(x_i, y_i, h_i) - \max_{\hat{h}_i \in H, \hat{y}_i \in Y} w^T \Phi(x_i, \hat{y}_i, \hat{h}_i) \geq \Delta(y_i, \hat{y}_i) - \xi_i,$$

Revisiting the constraints of latent SVM

$$\max_{h_i \in H} w^T \Phi(x_i, y_i, h_i) - \max_{\hat{h}_i \in H, \hat{y}_i \in Y} w^T \Phi(x_i, \hat{y}_i, \hat{h}_i) \geq \Delta(y_i, \hat{y}_i) - \xi_i,$$

- This essentially enforces that, the true detection should score higher than the false detection for each image.

Revisiting the constraints of latent SVM

$$\max_{h_i \in H} w^T \Phi(x_i, y_i, h_i) - \max_{\hat{h}_i \in H, \hat{y}_i \in Y} w^T \Phi(x_i, \hat{y}_i, \hat{h}_i) \geq \Delta(y_i, \hat{y}_i) - \xi_i,$$

- This essentially enforces that, the true detection should score higher than the false detection for each image.

- Generally the number of images in which the object is absent $\gg$ than images where the object is present.

Revisiting the constraints of latent SVM

$$\max_{h_i \in H} w^T \Phi(x_i, y_i, h_i) - \max_{\hat{h}_i \in H, \hat{y}_i \in Y} w^T \Phi(x_i, \hat{y}_i, \hat{h}_i) \geq \Delta(y_i, \hat{y}_i) - \xi_i,$$

- This essentially enforces that, the true detection should score higher than the false detection for each image.

- Generally the number of images in which the object is absent $\gg$ than images where the object is present.
- Thus the optimization may focus more on reducing the score of the highest scoring box of the negative class.

Revisiting the constraints of latent SVM

$$\max_{h_i \in H} w^T \Phi(x_i, y_i, h_i) - \max_{\hat{h}_i \in H, \hat{y}_i \in Y} w^T \Phi(x_i, \hat{y}_i, \hat{h}_i) \geq \Delta(y_i, \hat{y}_i) - \xi_i,$$

- This essentially enforces that, the true detection should score higher than the false detection for each image.

- Generally the number of images in which the object is absent $\gg$ than images where the object is present.
- Thus the optimization may focus more on reducing the score of the highest scoring box of the negative class.
- This imbalance can be handled elegantly using Blaschko's ranking constraints.

Ranking constraints : Blaschko and Vedaldi and Zisserman, NIPS 2010

- The constraints can be modified such that, the true detection for each image should score higher than the false detections for all the images.

Ranking constraints : Blaschko and Vedaldi and Zisserman, NIPS 2010

- The constraints can be modified such that, the true detection for each image should score higher than the false detections for all the images.
- This leads to modification in the objective function such that we can simultaneously localize and rank object detections.

Ranking constraints : Blaschko and Vedaldi and Zisserman, NIPS 2010

- The constraints can be modified such that, the true detection for each image should score higher than the false detections for all the images.
- This leads to modification in the objective function such that we can simultaneously localize and rank object detections.
- Thus our final objective function and constraints :-

Ranking constraints : Blaschko and Vedaldi and Zisserman, NIPS 2010

- The constraints can be modified such that, the true detection for each image should score higher than the false detections for all the images.
- This leads to modification in the objective function such that we can simultaneously localize and rank object detections.
- Thus our final objective function and constraints :-

$$\min_{w,\xi_i \geq 0, v \in \{0,1\}} \frac{1}{2} \|w\|^2 + \frac{C}{n.n_+} \sum_{i=1}^{n} v_i \xi_i - \frac{1}{K} \sum_{i=1}^{n} v_i \tag{4}$$

s.t :

$$\sum_{i,j} (\langle w, \phi(x_i, y_i) \rangle - \langle w, \phi(x_j, \hat{y}_j) \rangle) \geq \sum_{i,j} \Delta(y_j, \hat{y}_j) - \xi_i$$

Outline

Setup

- Dataset used
  - Training : PASCAL VOC 2007 train data (including 0 labels, 2501 examples per class)

Setup

- Dataset used
  - Training : PASCAL VOC 2007 train data (including 0 labels, 2501 examples per class)
  - Test : PASCAL VOC 2007 validation data (including 0 labels, 2510 examples per class)

Setup

- Dataset used
  - Training : PASCAL VOC 2007 train data (including 0 labels, 2501 examples per class)
  - Test : PASCAL VOC 2007 validation data (including 0 labels, 2510 examples per class)
  - Classes : Aeroplane, motorbike, person

Setup

- Dataset used
  - Training : PASCAL VOC 2007 train data (including 0 labels, 2501 examples per class)
  - Test : PASCAL VOC 2007 validation data (including 0 labels, 2510 examples per class)
  - Classes : Aeroplane, motorbike, person
- Proposal generation technique

Setup

- Dataset used
  - Training : PASCAL VOC 2007 train data (including 0 labels, 2501 examples per class)
  - Test : PASCAL VOC 2007 validation data (including 0 labels, 2510 examples per class)
  - Classes : Aeroplane, motorbike, person
- Proposal generation technique
  - Segmentation As Selective Search for Object Recognition (Sande, Koen, Uijlings, Jasper, Gevers, Theo, Smeulders, Arnold, ICCV 2011.)

Setup

- Dataset used
  - Training : PASCAL VOC 2007 train data (including 0 labels, 2501 examples per class)
  - Test : PASCAL VOC 2007 validation data (including 0 labels, 2510 examples per class)
  - Classes : Aeroplane, motorbike, person
- Proposal generation technique
  - Segmentation As Selective Search for Object Recognition (Sande, Koen, Uijlings, Jasper, Gevers, Theo, Smeulders, Arnold, ICCV 2011.)
  - Hypotheses : 2500 bounding boxes on an average per image on PASCAL VOC 2007.

Setup

- Dataset used
  - Training : PASCAL VOC 2007 train data (including 0 labels, 2501 examples per class)
  - Test : PASCAL VOC 2007 validation data (including 0 labels, 2510 examples per class)
  - Classes : Aeroplane, motorbike, person
- Proposal generation technique
  - Segmentation As Selective Search for Object Recognition (Sande, Koen, Uijlings, Jasper, Gevers, Theo, Smeulders, Arnold, ICCV 2011.)
  - Hypotheses : 2500 bounding boxes on an average per image on PASCAL VOC 2007.
  - These bounding boxes correspond to the values the latent variables can take for each image.

Setup

- Feature extraction
  - Pre-trained convolutional neural network using caffe framework for feature extraction.

Setup

- Feature extraction
  - Pre-trained convolutional neural network using caffe framework for feature extraction.
  - Network pre-trained on Imagenet Large Scale Visual Recognition Challenge 2012 dataset which contains 1000 classes.

Setup

- Feature extraction
  - Pre-trained convolutional neural network using caffe framework for feature extraction.
  - Network pre-trained on Imagenet Large Scale Visual Recognition Challenge 2012 dataset which contains 1000 classes.
  - Layer 7 responses were used as features, giving a 4096 dimension feature for each input image.

Setup

- Feature extraction
  - Pre-trained convolutional neural network using caffe framework for feature extraction.
  - Network pre-trained on Imagenet Large Scale Visual Recognition Challenge 2012 dataset which contains 1000 classes.
  - Layer 7 responses were used as features, giving a 4096 dimension feature for each input image.
- SVM used for Outlier removal

Setup

- Feature extraction
  - Pre-trained convolutional neural network using caffe framework for feature extraction.
  - Network pre-trained on Imagenet Large Scale Visual Recognition Challenge 2012 dataset which contains 1000 classes.
  - Layer 7 responses were used as features, giving a 4096 dimension feature for each input image.
- SVM used for Outlier removal
  - Linear SVM.

Setup

- Feature extraction
    - Pre-trained convolutional neural network using caffe framework for feature extraction.
    - Network pre-trained on Imagenet Large Scale Visual Recognition Challenge 2012 dataset which contains 1000 classes.
    - Layer 7 responses were used as features, giving a 4096 dimension feature for each input image.
- SVM used for Outlier removal
    - Linear SVM.
    - LibLinear package : Fan, Chang, Hsieh, Wang, Lin. 2008

Outline

Outliers detected by Outlier Robust SVM

Outliers detected by Outlier Robust SVM

Outliers detected by Outlier Robust SVM

Outliers detected by Outlier Robust SVM

Outliers detected by Outlier Robust SVM

Outliers detected by Outlier Robust SVM

Outline

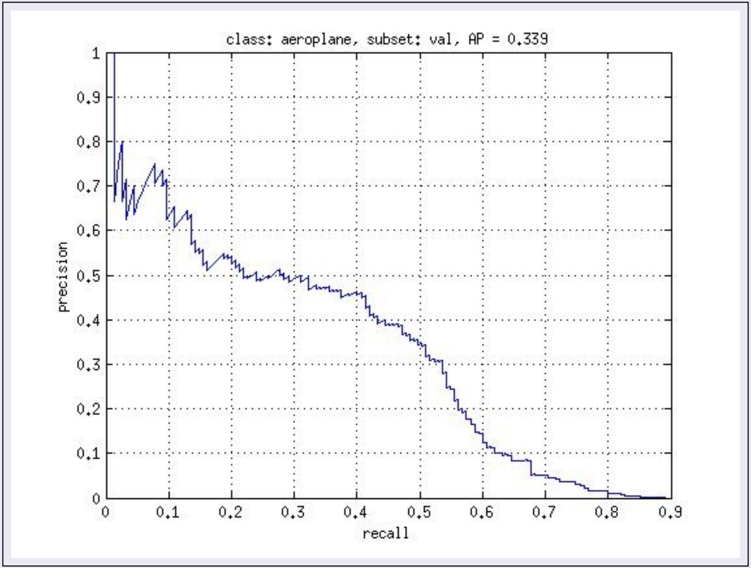
Detection results

Figure: Detection results of aeroplane, motorbike and person class

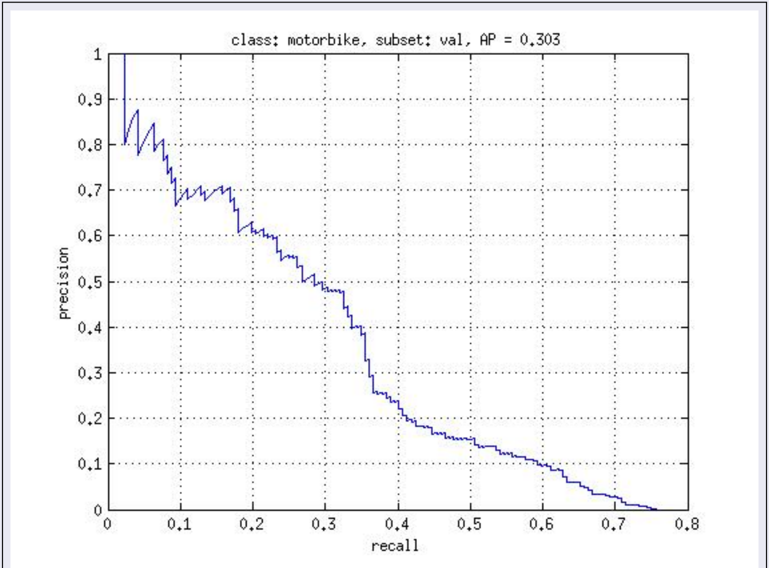
Outline

1. Problem Statement
2. Background
3. Method Description
4. Results
    - Setup
    - Outlier detections
    - Detection results
    - mAP graphs
5. Conclusion and Future Scope

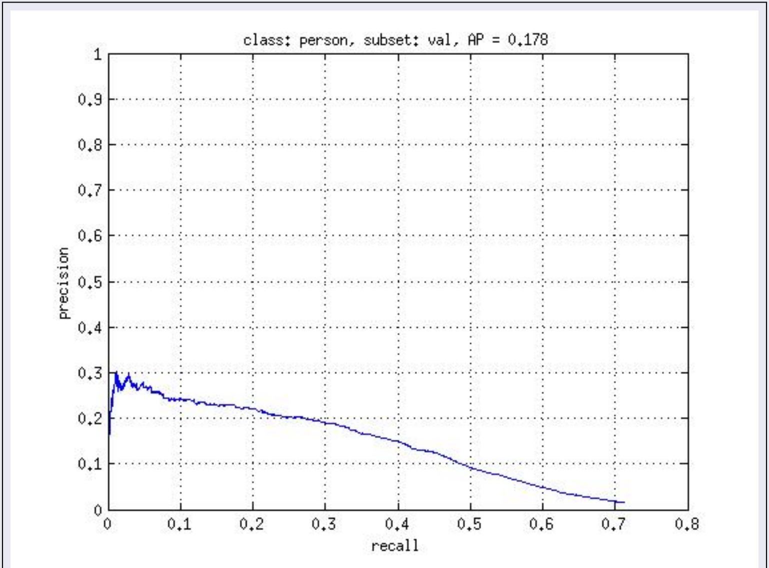
Mean average precision : aeroplane



Mean average precision : motorbike



Mean average precision : person



Mean average precision comparison

Conclusion

- Hard examples can often degrade the performance of the detector and thus should be treated carefully.

Conclusion

- Hard examples can often degrade the performance of the detector and thus should be treated carefully.
- Imposing an ordering on learning may help us to obtain a better solution.

Conclusion

- Hard examples can often degrade the performance of the detector and thus should be treated carefully.
- Imposing an ordering on learning may help us to obtain a better solution.

Limitation

- Hardness in feature space may not correspond to visual hardness.

Conclusion

- Hard examples can often degrade the performance of the detector and thus should be treated carefully.
- Imposing an ordering on learning may help us to obtain a better solution.

Limitation

- Hardness in feature space may not correspond to visual hardness.
- Latent SVM primal objective being non-convex, may still converge to a local optimum.

Conclusion

- Hard examples can often degrade the performance of the detector and thus should be treated carefully.
- Imposing an ordering on learning may help us to obtain a better solution.

Limitation

- Hardness in feature space may not correspond to visual hardness.
- Latent SVM primal objective being non-convex, may still converge to a local optimum.

Future Scope

- The work can be extended to provide robust detection when (i) image labels are noisy (ii) bounding box labels are noisy.

Conclusion

- Hard examples can often degrade the performance of the detector and thus should be treated carefully.
- Imposing an ordering on learning may help us to obtain a better solution.

Limitation

- Hardness in feature space may not correspond to visual hardness.
- Latent SVM primal objective being non-convex, may still converge to a local optimum.

Future Scope

- The work can be extended to provide robust detection when (i) image labels are noisy (ii) bounding box labels are noisy.
- Application of Outlier Robust SVM to other ML problems.

THANK YOU