

# Vehicle Detection using HOG and HAAR Features

Machine Learning for Cities

By

Akshay Penmatcha  
Priyanshi Singh  
Sunny Kulkarni

# BACKGROUND

---

- › Can we track the vehicle flows in real-time using traffic camera feeds?



- › Possible Uses:

- › Detecting and Tracking Congestions in real-time
- › Efficient Traffic Planning and Management

# APPROACH

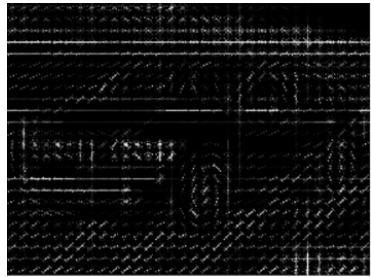
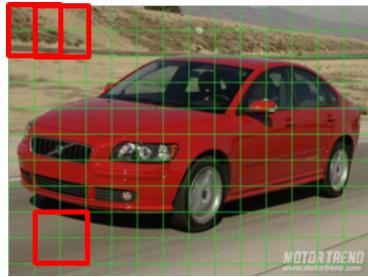
---

- › **Shallow Learning Techniques:**
  - 1) Using HOG Features and applying an SVM Classifier
  - 2) Using HAAR Features and applying a Cascade Classifier
- › **Deep Learning Techniques:**
  - › Convolutional Neural Networks
  - › R-CNN



# HOG - SVM

## Histogram of Oriented Gradients (HOG)



### Parameters

Cell Size = 10x10 Pixels

Stride = 10 Pixels

Block Size = 20x20 Pixels

Bins = 8



Filter masks in x-y direction

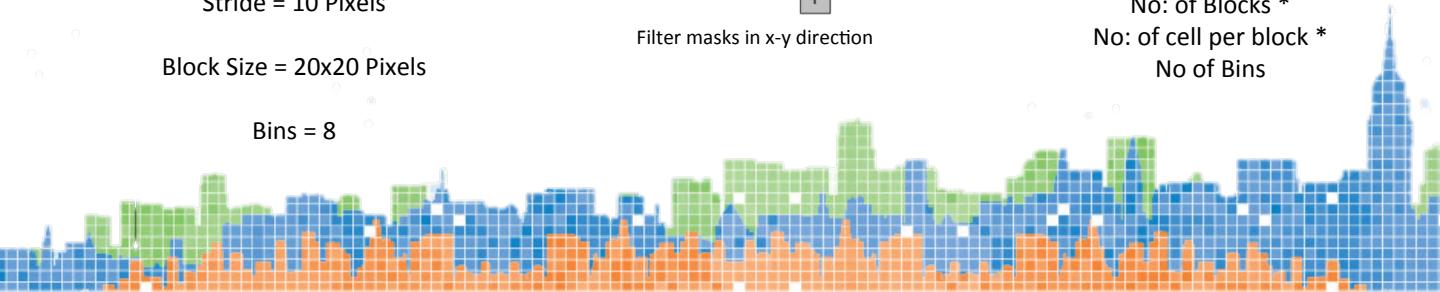
### Output

Flattened vector of size:

No: of Blocks \*

No: of cell per block \*

No of Bins



# HOG - SVM

## Process Flow

Step-1: Labelled Dataset

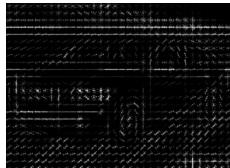


+ve Image

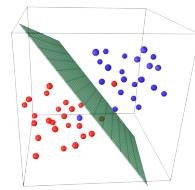


-ve Image

Step-2: HOG Extraction

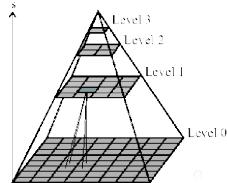


Step-3: Train SVM



\*Source: Google

Step-4: Image Pyramids

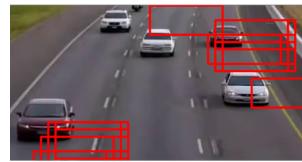


\*Source: Google

Step-5: Sliding Window



Step-6: Non-Max Suppression



# HOG - SVM

## Results

Training Set



Positive: 500



Negative: 500

- Train-Test Split : 20%
- C=1 (inflection point is 0.1)

Confusion Matrix	Predicted: NO	Predicted: YES
Actual: NO	99	1
Actual: YES	1	99

# HOG - SVM

## Results

Training Set



Positive: 500

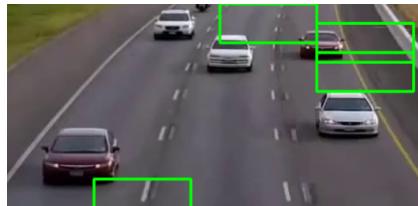
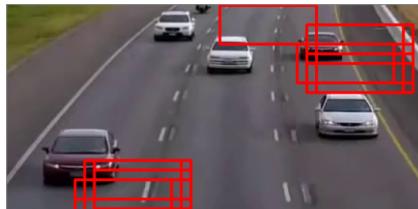


Negative: 500

- Train-Test Split : 20%
- C=1 (inflection point is 0.1)

Confusion Matrix	Predicted: NO	Predicted: YES
Actual: NO	99	1
Actual: YES	1	99

Test Image from Traffic Cam



**Solution:** Hard Negative Mining

# HAAR – CASCADE CLASSIFIER

## HAAR-like Features

Type 1



Type 2



Type 3



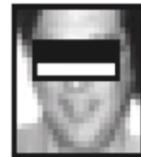
Type 4



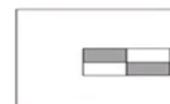
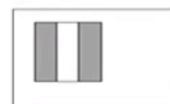
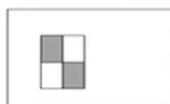
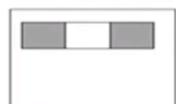
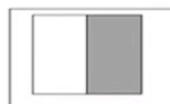
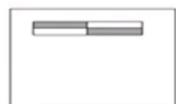
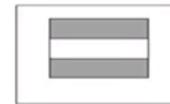
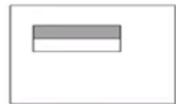
Type 5



Haar Features used in Viola Jones



Applying on a given image



**16,000 + features in a  
24x24 window**

# HAAR – CASCADE CLASSIFIER

## Adaboost Classifier



- Only few set of features useful among 16,000 plus features.
- That's why, We have Adaboost.
- weighted combination of all features, each features is called a weak Classifier

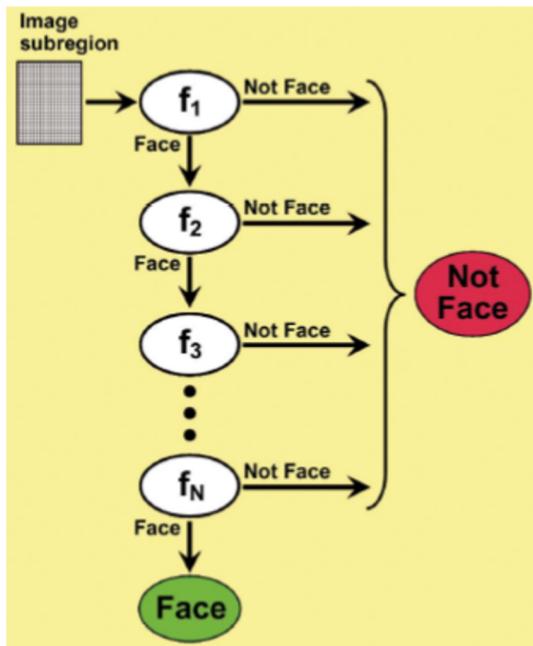
$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \dots$$

 
  
 Strong classifier      Weak classifier

Disadvantage: High Computation Cost

# HAAR – CASCADE CLASSIFIER

## Cascading



# HAAR – CASCADE CLASSIFIER

## Training Function

```
opencv_traincascade -data data -vec positives.vec
-bg bg.txt -numPos 1200 -numNeg 600 -numStages 10
-w 20 -h 20
```

`boostType: GAB => Gentle AdaBoost`

`featureType: HAAR`

`mode: BASIC => Upright rectangle Haar-like features`

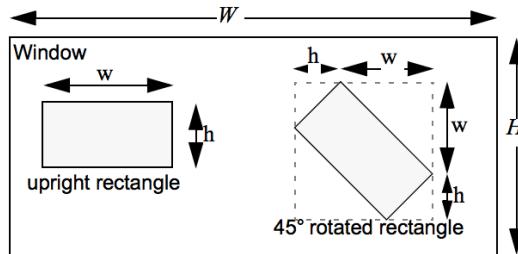


Image Source: Rainer Lienhart, Alexander Kuranov, Vadim Pisarevsky, *Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection* Intel Corporation.

# HAAR – CASCADE CLASSIFIER

## Results

```
===== TRAINING 1-stage =====
<BEGIN
POS count : consumed    1380 : 1384
NEG count : acceptanceRatio   900 : 0.347222
Precalculation time: 6
+---+---+---+
| N |   HR |   FA |
+---+---+---+
| 1|     1|     1|
+---+---+---+
| 2|     1|     1|
+---+---+---+
| 3|     1|     1|
+---+---+---+
| 4| 0.998551| 0.593333|
+---+---+---+
| 5| 0.995652| 0.258889|
+---+---+---+
END>
Training until now has taken 0 days 0 hours 0 minutes 56 seconds.

===== TRAINING 2-stage =====
<BEGIN
POS count : consumed    1380 : 1390
NEG count : acceptanceRatio   900 : 0.0803715
Precalculation time: 5
+---+---+---+
| N |   HR |   FA |
+---+---+---+
| 1|     1|     1|
+---+---+---+
| 2|     1|     1|
+---+---+---+
| 3| 0.995652| 0.788889|
+---+---+---+
| 4| 0.995652| 0.788889|
+---+---+---+
| 5| 0.998551| 0.634444|
+---+---+---+
| 6| 0.995652| 0.603333|
+---+---+---+
| 7| 0.995652|     0.41|
+---+---+---+
END>
Training until now has taken 0 days 0 hours 1 minutes 31 seconds.
```

- N = Current feature for this cascade.
- HR = Hit Rate based on the Stage Threshold (hit rate/numPos)
- FA = False Alarm based on the Stage Threshold (false alarm rate / numNeg)

Stage Threshold = Value set to achieve accuracy of at least 50% in identification of positive vehicle images.

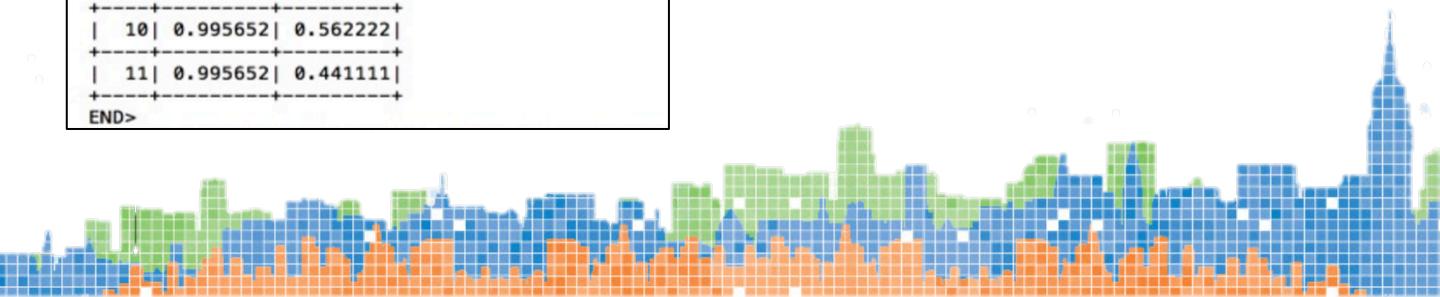
# HAAR – CASCADE CLASSIFIER

## Results

```
===== TRAINING 6-stage =====
<BEGIN
POS count : consumed    1380 : 1414
NEG count : acceptanceRatio   900 : 0.00320153
Precalculation time: 6
+ + + + +
| N | HR | FA |
+ + + + +
| 1| 1| 1|
+ + + + +
| 2| 1| 1|
+ + + + +
| 3| 0.998551| 0.797778|
+ + + + +
| 4| 0.998551| 0.797778|
+ + + + +
| 5| 0.996377| 0.718889|
+ + + + +
| 6| 0.996377| 0.627778|
+ + + + +
| 7| 0.995652| 0.582222|
+ + + + +
| 8| 0.995652| 0.627778|
+ + + + +
| 9| 0.995652| 0.55|
+ + + + +
| 10| 0.995652| 0.562222|
+ + + + +
| 11| 0.995652| 0.441111|
+ + + + +
END>
```

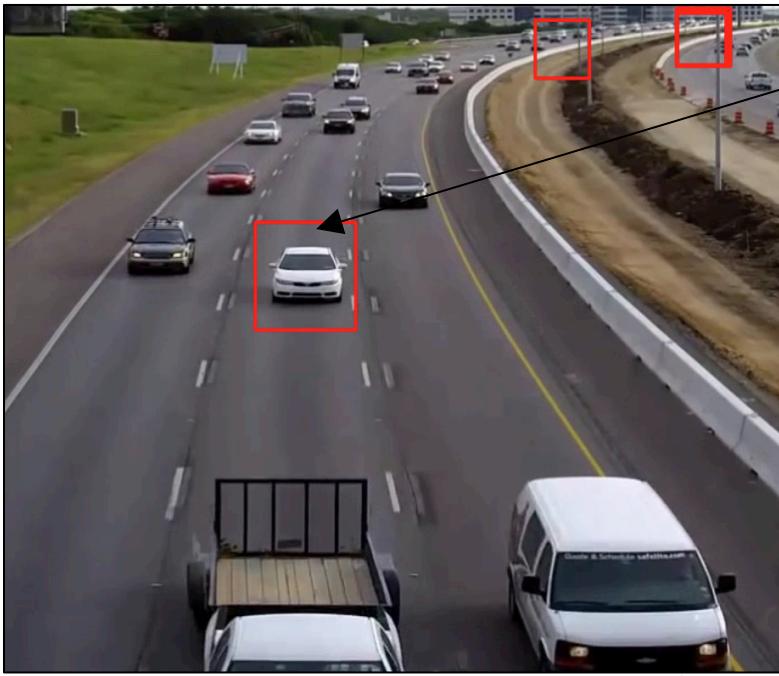
## Training the Classifier - Stage 6

- Acceptance Ratio improves for each subsequent training stage.
- Acceptance Ratio value for -
  - Stage 1 = 0.35
  - Stage 2 = 0.08
  - and Stage 6 = 0.0032



# HAAR – CASCADE CLASSIFIER

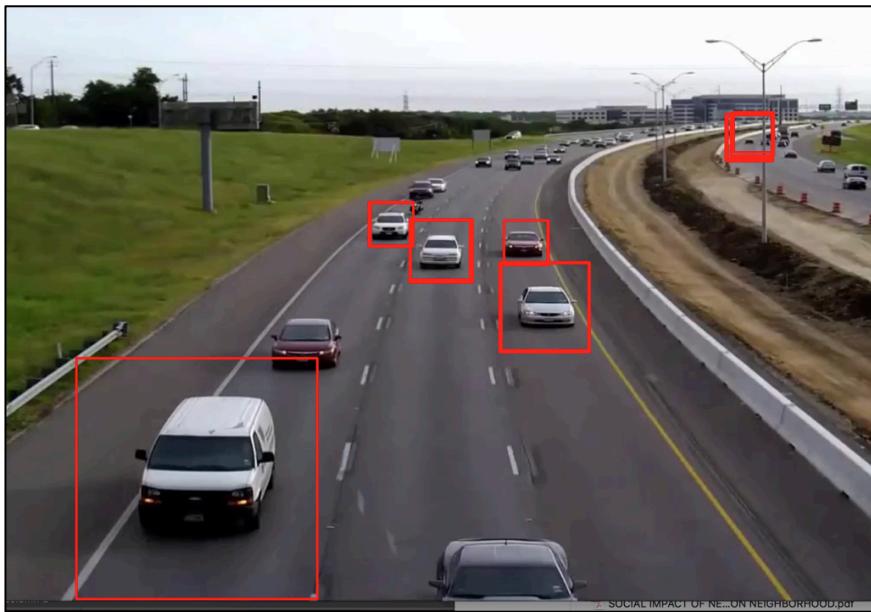
## Results



Vehicle used  
to train the  
classifier

# HAAR – CASCADE CLASSIFIER

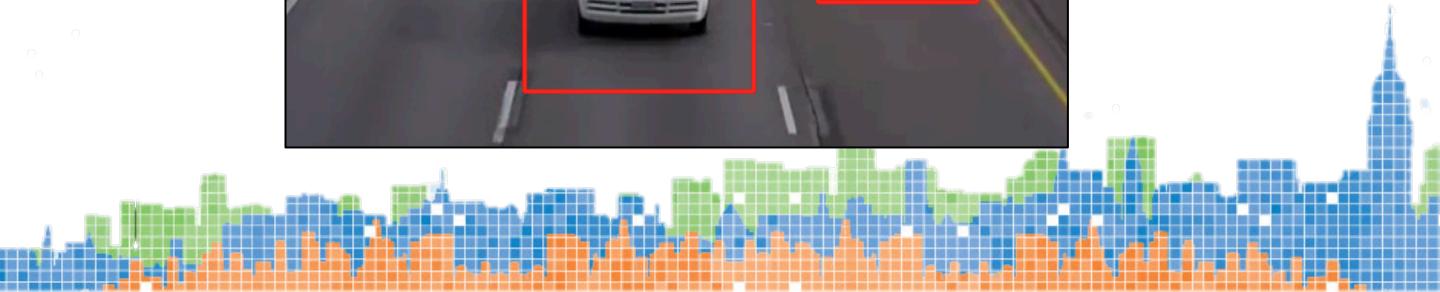
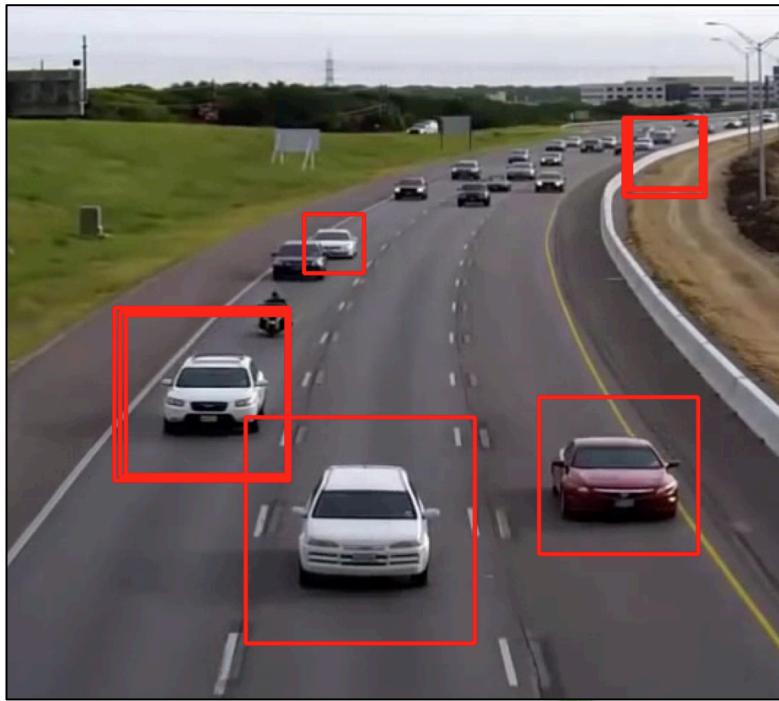
## Results



SOCIAL IMPACT OF NE...ON NEIGHBORHOOD.pdf

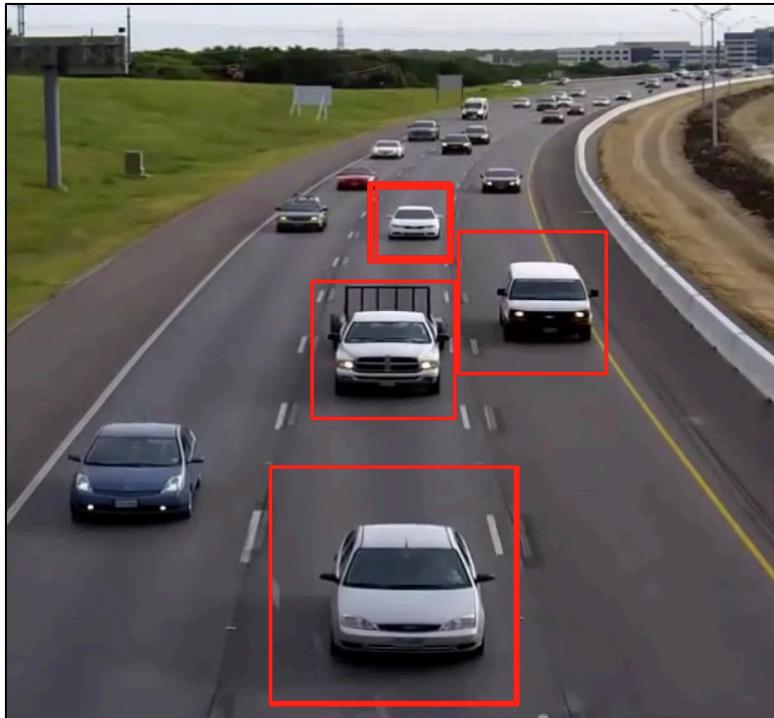
# HAAR – CASCADE CLASSIFIER

## Results



# HAAR – CASCADE CLASSIFIER

## Results



# REFERENCES

---

1. Paul Viola, Microsoft Research, Michael J. Jones, Mitsubishi Electric Research Laboratory, 2003, *Robust Real-Time Face Detection*:  
<http://www.vision.caltech.edu/html-files/EE148-2005-Spring/pprs/viola04ijcv.pdf>
2. Rainer Lienhart, Alexander Kuranov, Vadim Pisarevsky Microprocessor Research Lab, 2003, Intel Labs Intel Corporation, *Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection*:  
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.139.4825>
3. Gary Bradski, Adrian Kaehler, Vadim Pisarevsky, 2005, Vol 09, Issue 02, Intel Corporation, *Learning-Based Computer Vision with Intel's Open Source Computer Vision Library*:  
[http://www.willowgarage.com/sites/default/files/vol09\\_art03.pdf](http://www.willowgarage.com/sites/default/files/vol09_art03.pdf)
4. Youtube: ramsrigouthamg, <https://www.youtube.com/watch?v=WfdYYNamHZ8>

**THANK YOU**



# Appendix

## Basic Introduction to Edge Detection:



-1	-1	-1
2	2	2
-1	-1	-1

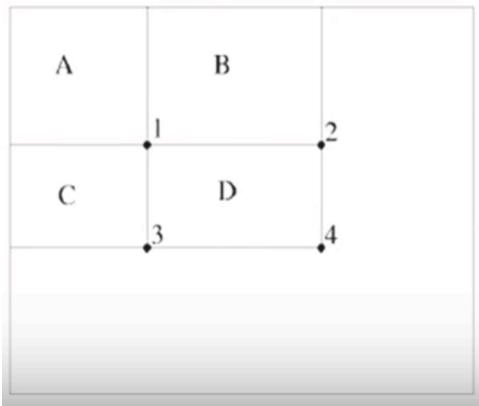
Convolution Kernel



Output Image (right) has higher intensity at pixels where the convolutional kernel Pixel matches perfectly with the input image

# Viola Jones Object Detection Algorithm

## Integral Image



Sum of all pixels in  
 $D = A + (A+B+C+D) - (A+C+A+B)$   
 $= 1 + 4 - (2+3)$



# RESULTS

## Open CV: Train Cascade function parameters -

```
PARAMETERS:  
cascadeDirName: data  
vecFileName: positives.vec  
bgFileName: bg.txt  
numPos: 1380  
numNeg: 900  
numStages: 10  
precalcValBufSize[Mb] : 1024  
precalcIdxBufSize[Mb] : 1024  
acceptanceRatioBreakValue : -1  
stageType: BOOST  
featureType: HAAR  
sampleWidth: 20  
sampleHeight: 20  
boostType: GAB  
minHitRate: 0.995  
maxFalseAlarmRate: 0.5  
weightTrimRate: 0.95  
maxDepth: 1  
maxWeakCount: 100  
mode: BASIC  
Number of unique features given windowSize [20,20] : 78460
```

# DATASETS EXPLORED

---

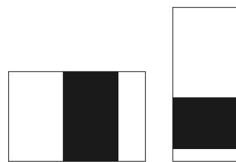


# HAAR – CASCADE CLASSIFIER

## HAAR-like Features



Edge Features

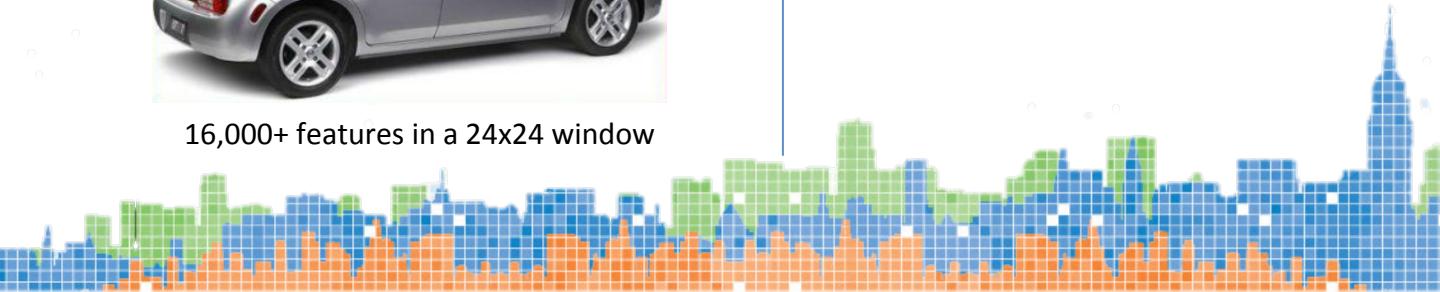


Line Features

60 other different types of kernels

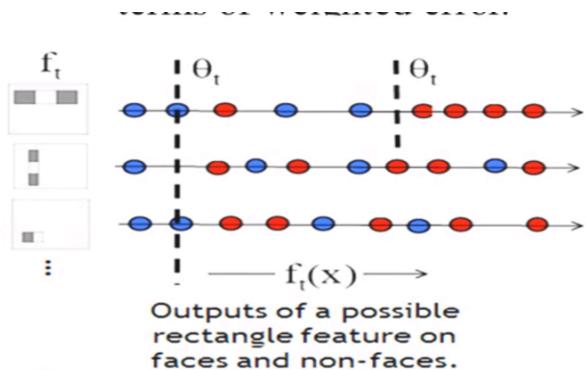


16,000+ features in a 24x24 window

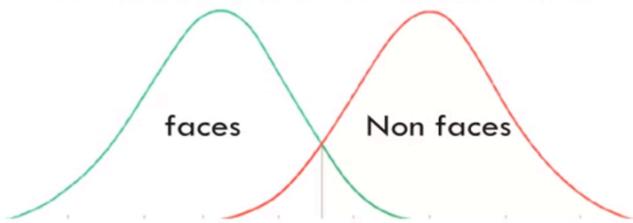


# HAAR – CASCADE CLASSIFIER

## Adaboost Classifier



A Gaussian weak classifier used



- Start with uniform weights on training examples, for N rounds
- Evaluate weighted error for each feature, pick best
- Reweight the examples:
  - Incorrectly classified-> more weights
  - Correctly classified-> less weight

**Final Classifier is the combination of weak ones, weighted according to error they had.**

Image Source: [ramsrigouthamg, https://www.youtube.com/watch?v=WfdYYNamHZ8](https://www.youtube.com/watch?v=WfdYYNamHZ8)