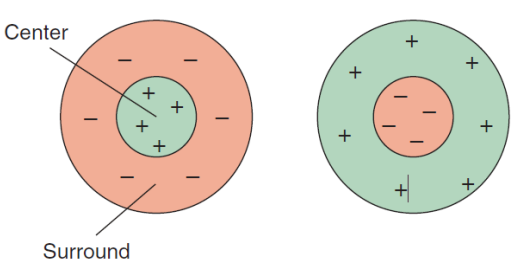
Transformada Hough

PROF. CESAR HENRIQUE COMIN

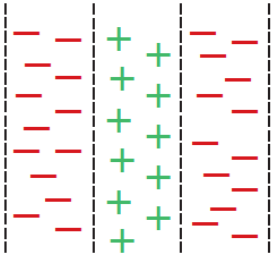
Bordas

Como já vimos, bordas usualmente fornecem importantes informações sobre um objeto

Por causa disso, nossa visão possui diversos mecanismos dedicados a detectar bordas



Campo receptivo da retina



Campo receptivo de uma célula cortical

Bordas

Bordas

Retas

Em muitos casos, retas podem ser entendidas como um tipo especial de borda, na qual todos os pontos são colineares.

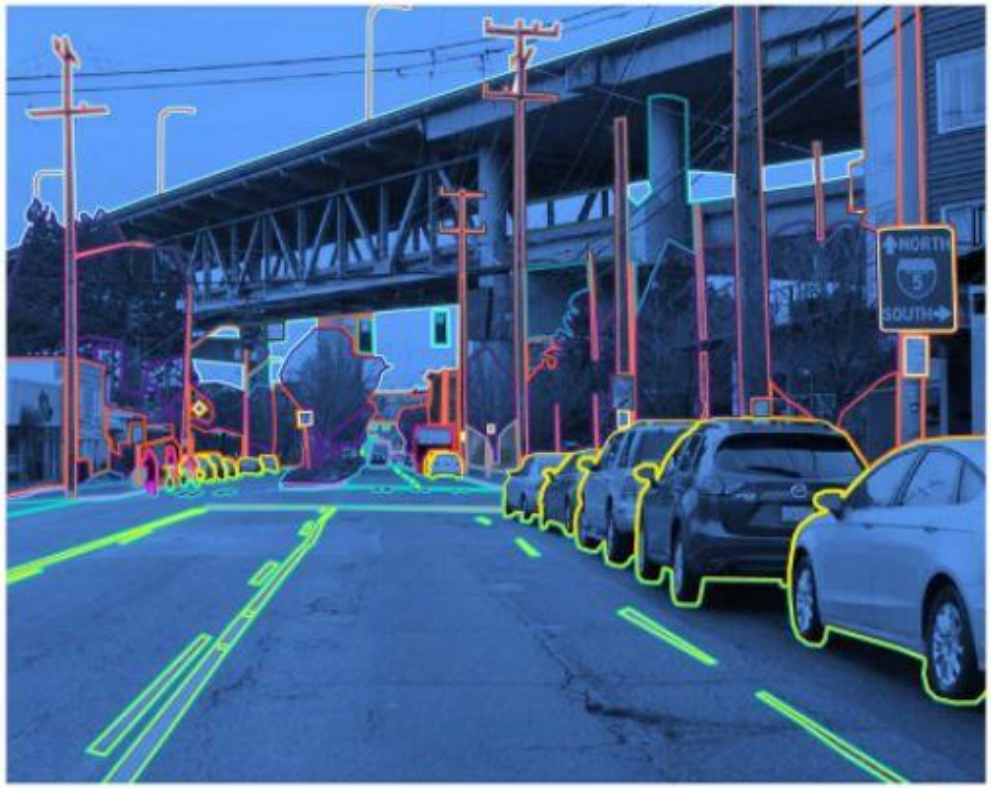
Humanos adoram construir retas

Humanos adoram construir retas

Humanos adoram construir retas

Retas

- Encontrar retas em uma imagem ou vídeo é uma tarefa muito importante, especialmente em cenários urbanos.

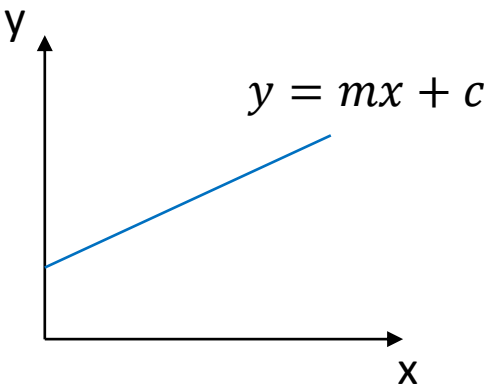


A Transformada Hough

- A transformada Hough é uma técnica que pode ser utilizada para a identificação de retas em uma imagem

A Transformada Hough

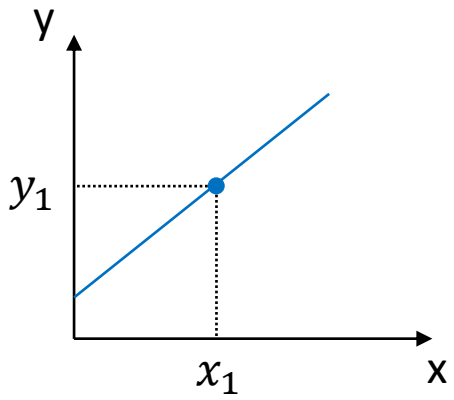
- A transformada Hough é uma técnica que pode ser utilizada para a identificação de retas em uma imagem
- Uma reta é descrita pela equação $y = mx + c$



A Transformada Hough

- A transformada Hough é uma técnica que pode ser utilizada para a identificação de retas em uma imagem

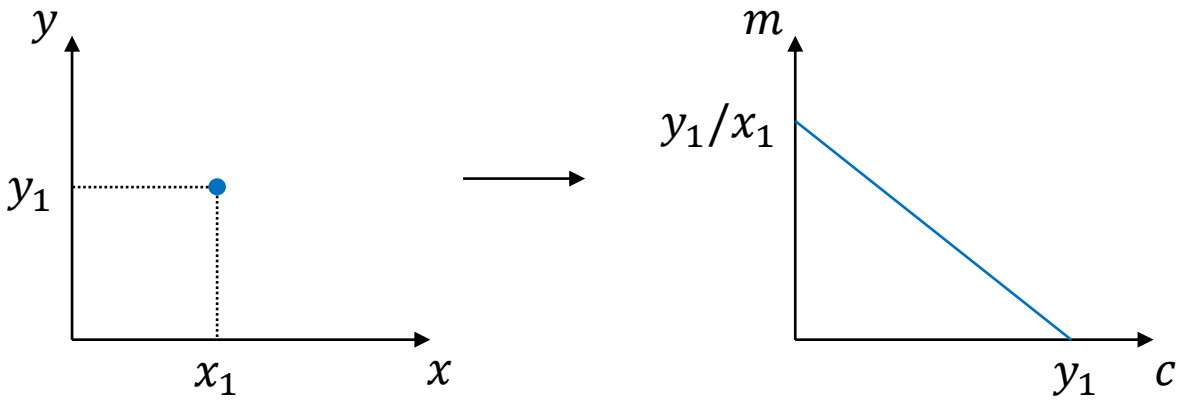
- Uma reta é descrita pela equação $y = mx + c$
- Ou seja, cada ponto na reta possui a restrição $y_i = mx_i + c$



A Transformada Hough

- A transformada Hough é uma técnica que pode ser utilizada para a identificação de retas em uma imagem

- Uma reta é descrita pela equação $y = mx + c$
- Ou seja, cada ponto na reta possui a restrição $y_i = mx_i + c$
- Portanto, para cada ponto temos que $m = -c/x_i + y_i/x_i$

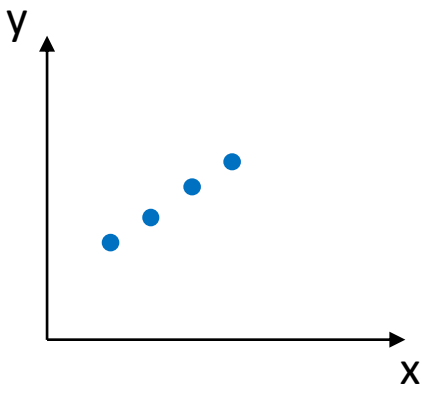


Espaço de coordenadas da imagem $x \times y$

Espaço de parâmetros de reta $c \times m$

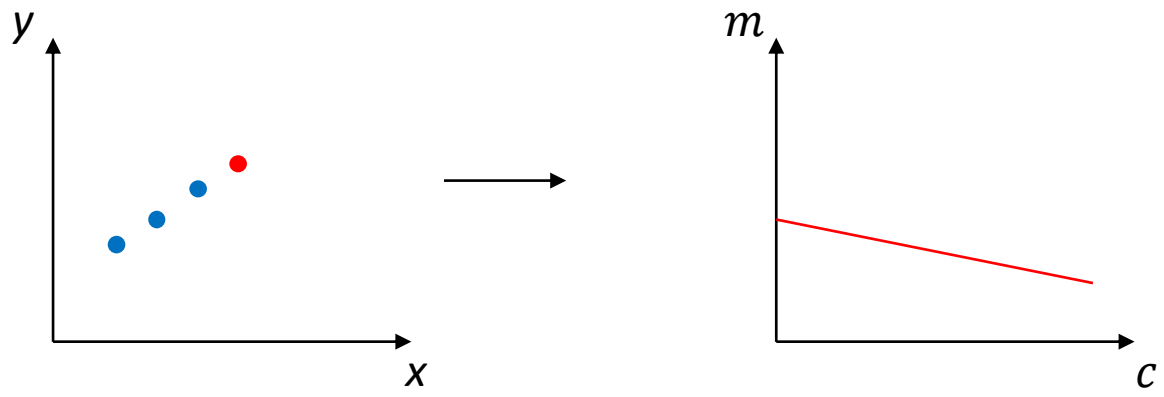
A Transformada Hough

- Para cada ponto da imagem, podemos encontrar a reta que ele define no espaço de parâmetros



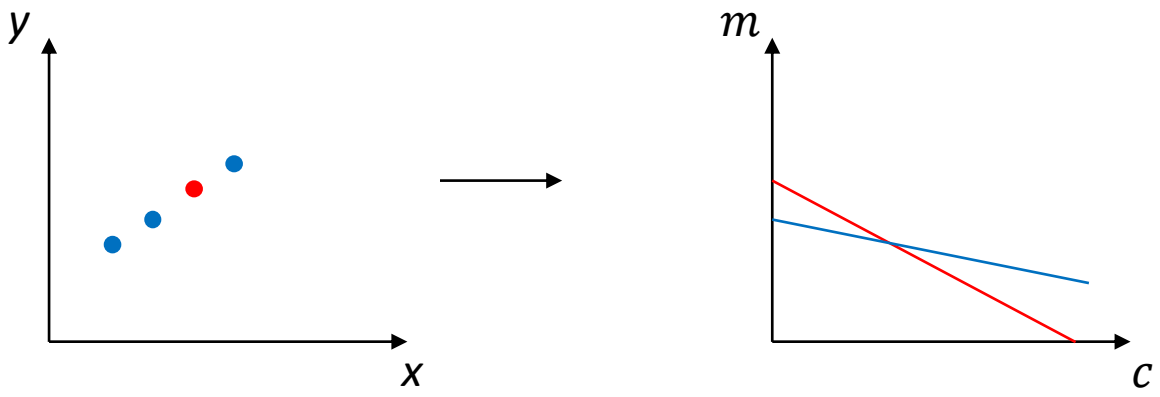
A Transformada Hough

- Para cada ponto da imagem, podemos encontrar a reta que ele define no espaço de parâmetros



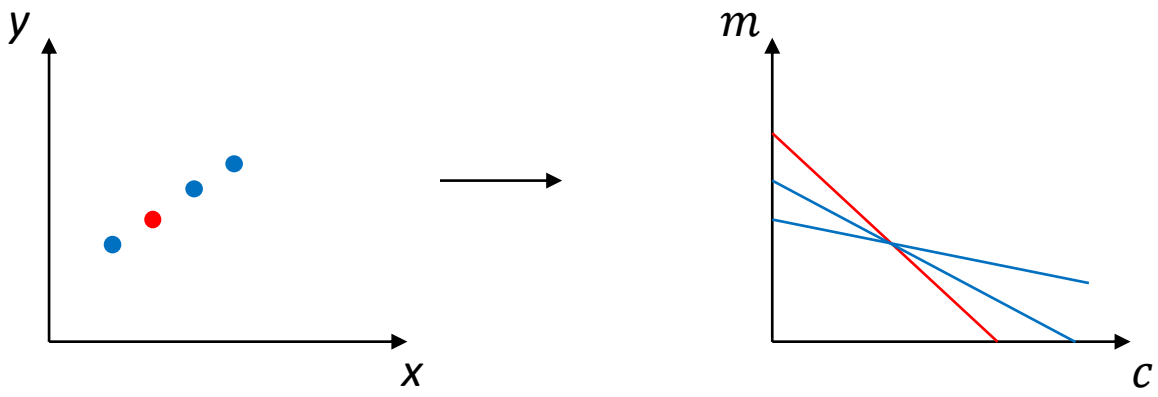
A Transformada Hough

- Para cada ponto da imagem, podemos encontrar a reta que ele define no espaço de parâmetros



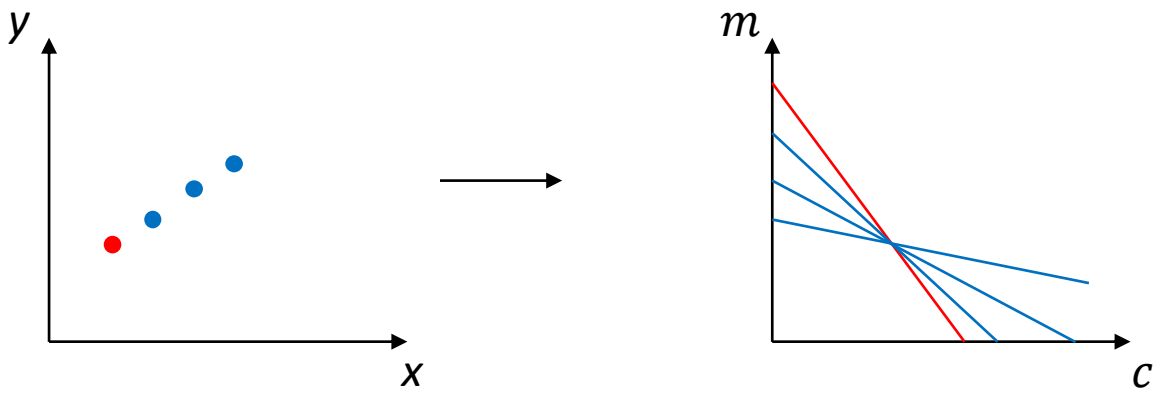
A Transformada Hough

- Para cada ponto da imagem, podemos encontrar a reta que ele define no espaço de parâmetros



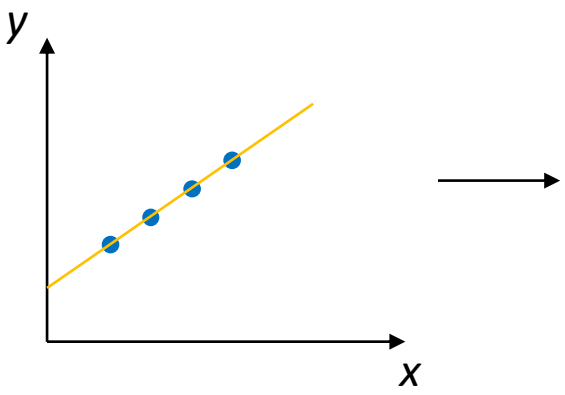
A Transformada Hough

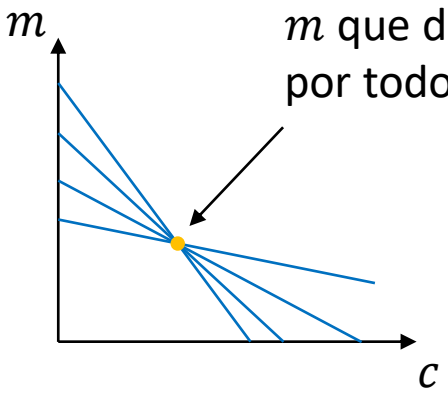
- Para cada ponto da imagem, podemos encontrar a reta que ele define no espaço de parâmetros



A Transformada Hough

- Para cada ponto da imagem, podemos encontrar a reta que ele define no espaço de parâmetros
- A reta que passa por todos os pontos terá os mesmos parâmetros





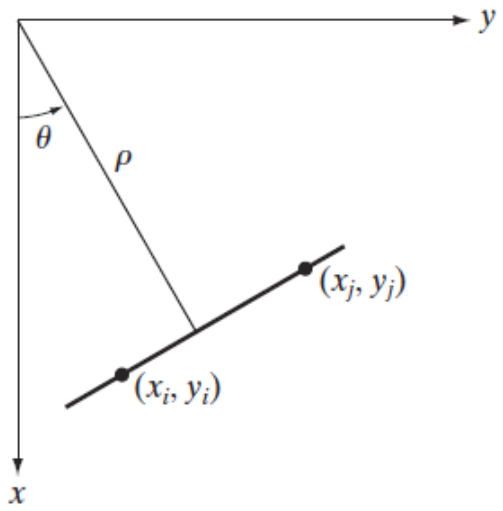
c e m igual para todos os pontos, ou seja, o valor de c e m que define a reta passando por todos os pontos

A Transformada Hough

Um problema de representar retas utilizando a equação $y = mx + c$ é que retas verticais não podem ser representadas

Portanto, utilizamos a representação normal (ou polar) de uma reta:

$$\rho = xcos\theta + ysin\theta$$



ρ: distância da origem ao ponto da reta mais próximo à origem. Equivalentemente, tamanho do vetor perpendicular à reta e que começa na origem e termina em um ponto da reta
θ: ângulo entre o vetor acima e o eixo vertical

A Transformada Hough

- A transformada Hough consiste em utilizar cada ponto branco de uma imagem binária para definir uma curva no espaço $\theta \times \rho$
- Os cruzamentos entre diferentes curvas representam pontos colineares na imagem

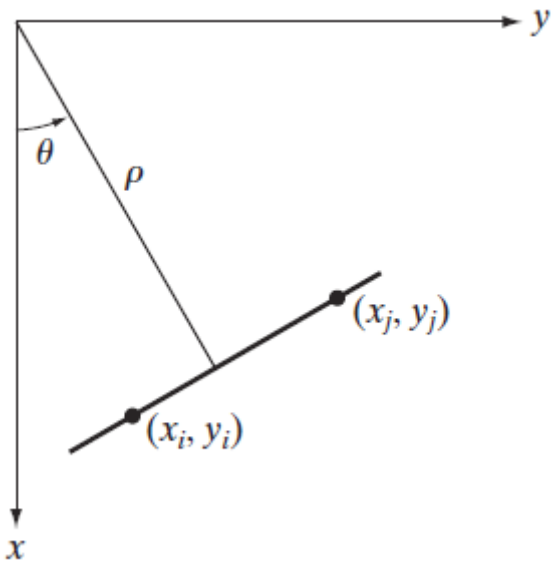
A Transformada Hough

Na representação normal, cada ponto (x_i, y_i) define a função

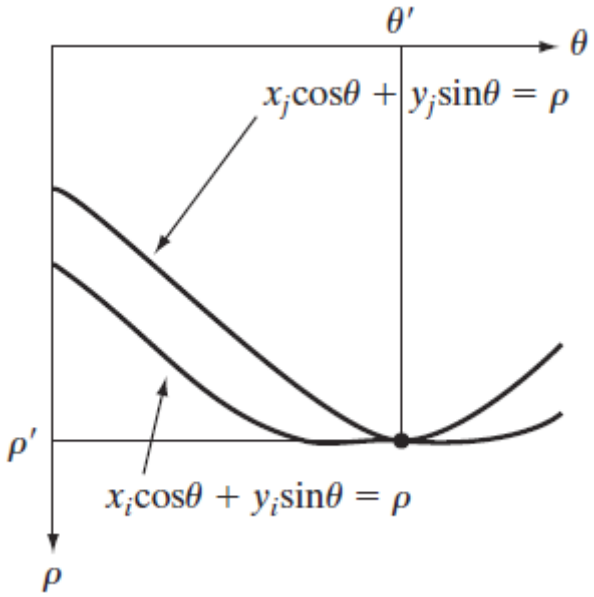
$$\rho = x_i cos\theta + y_i sin\theta$$

no espaço de parâmetros

Espaço de coordenadas da imagem

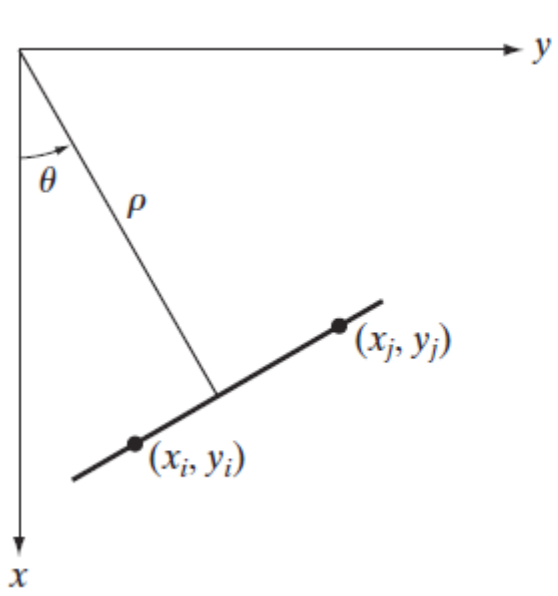


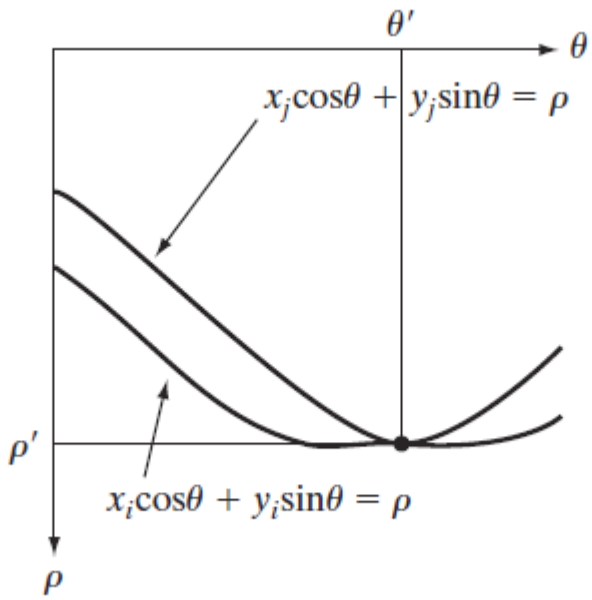
Espaço de parâmetros de reta

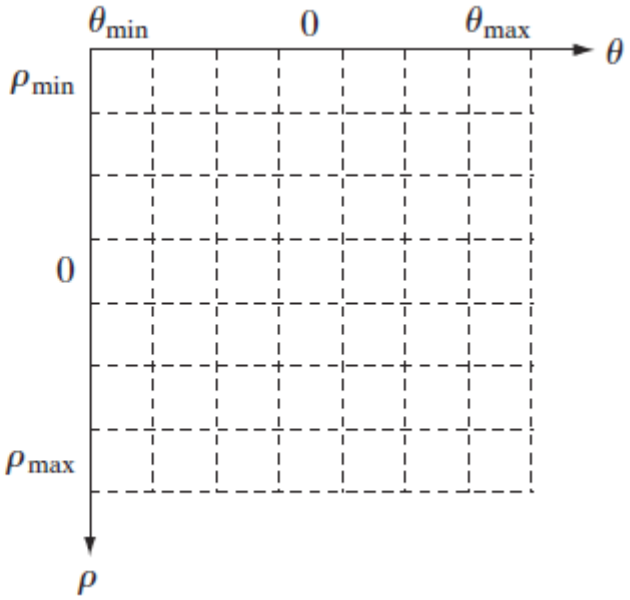


A Transformada Hough

- Para cada pixel branco de uma imagem binária, temos uma equação $\rho = x_i cos\theta + y_i sin\theta$
- Para armazenar os valores dessa equação, construímos um espaço de parâmetros $\rho \times \theta$ (um array 2D) implementado na forma de um acumulador
- Os principais parâmetros desse espaço são a resolução $\Delta\rho$ e $\Delta\theta$ do acumulador.

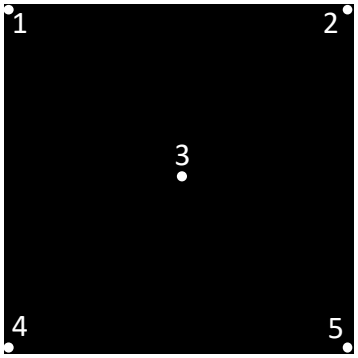




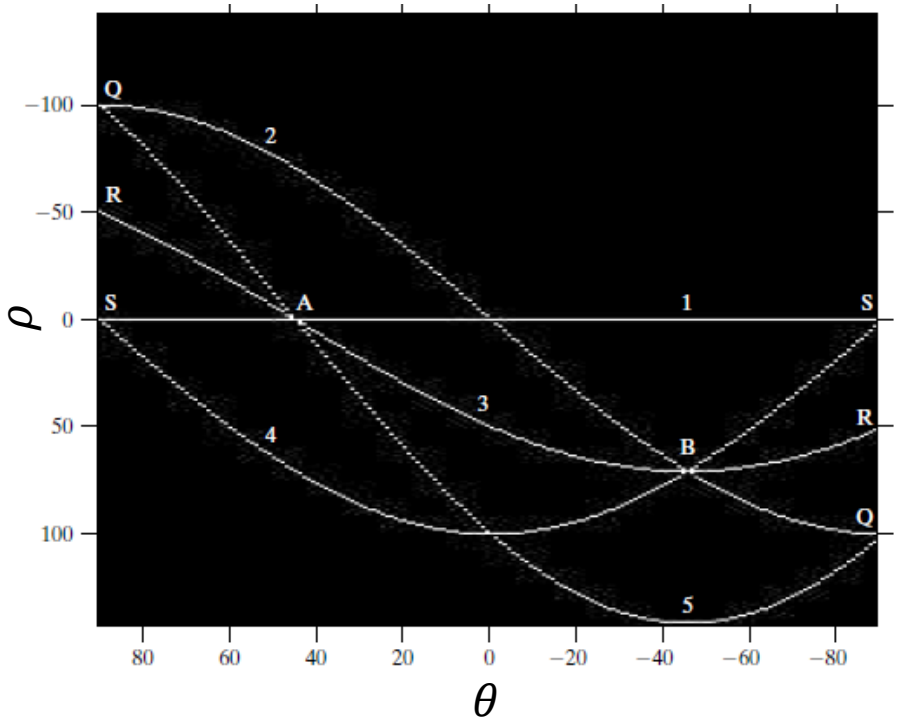


A Transformada Hough - Exemplo

Imagem contendo 5 pontos



Transformada Hough



A Transformada Hough

Algoritmo:

1. Obtenha uma imagem binária na qual pixels brancos representam bordas de objetos;
2. Especifique o espaço $\rho \times \theta$, e defina a respectiva matriz acumuladora M
3. Para cada pixel branco na imagem
   - 3a. Encontre a equação normal definida pela posição do pixel
   - 3b. Para θ variando no intervalo $[-\pi/2, \pi/2]$, com passo de $\Delta\theta$, encontre os valores de ρ correspondentes. Adicione 1 para cada posição (ρ_i, θ_i) em M
4. Encontre picos na matriz M

A Transformada Hough

Note que a escolha da resolução do espaço $\rho \times \theta$ (ajustada por $\Delta\rho$ e $\Delta\theta$) é muito importante:

- Se $\Delta\rho$ e $\Delta\theta$ forem alto, retas distintas serão consideradas iguais
- Se $\Delta\rho$ e $\Delta\theta$ forem pequeno, ruídos na imagem terão maior influência no resultado. Adicionalmente, o custo do algoritmo aumenta

A Transformada Hough - Implementação

Notebook “**Transformada Hough**”

A Transformada Hough – detectando círculos

A transformada Hough também pode ser utilizada na detecção de círculos (e diversos outros tipos de formas).

Equação do círculo

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

Podemos construir um acumulador 3D e, para cada ponto de borda (x_i, y_i), encontrar os valores correspondentes de x_0, y_0 e r.

Introdução à Classificação de Imagens

PROF. CESAR HENRIQUE COMIN

Tarefas de Visão Computacional

Classificação	Classificação + localização	Detecção de objetos	Segmentação semântica	Segmentação de instâncias
				
Gato	Gato, [(43, 22), (78, 97)]	Cachorro, Cachorro, Gato	Céu, Árvore, Grama, Gato	Cachorro, Cachorro, Gato

Aprendizado de máquina

- Classificação supervisionada:

- Classificação não-supervisionada:

Aprendizado de máquina

- Classificação supervisionada:
  - O classificador é treinado em uma base de dados na qual a classe dos objetos é conhecida
  - Portanto, cada imagem utilizada no treinamento precisa ser associada a uma classe
  - Isso requer a rotulação manual de todas as imagens na base de treinamento

- Classificação não-supervisionada:

Aprendizado de máquina

- Classificação supervisionada:
  - O classificador é treinado em uma base de dados na qual a classe dos objetos é conhecida
  - Portanto, cada imagem utilizada no treinamento precisa ser associada a uma classe
  - Isso requer a rotulação manual de todas as imagens na base de treinamento

- Classificação não-supervisionada:
  - O classificador é treinado sem possuir informação sobre os rótulos dos objetos
  - O classificador automaticamente identifica a “definição” do que é cada classe
  - Classificação não-supervisionada é mais difícil do que classificação supervisionada
  - Exemplo: k-médias, mean shift, mistura de gaussianas

Classificação de imagens supervisionada

- Classificação de imagens supervisionada lida com a seguinte situação:
  Dada uma imagem I e um conjunto de possíveis classes $c_1, c_2, \dots, c_k$ que essa imagem pode pertencer, qual a classe mais provável de I?

Classificação de imagens supervisionada

Exemplo:

Temos a classe cachorro e gato. Qual a classe da imagem abaixo?

classe=?

* Para nós, essa é uma tarefa muito simples. Para o computador, é uma tarefa muito complicada

Classificação de imagens supervisionada

Para podermos classificar uma imagem como cachorro ou gato, precisamos primeiro construir uma base de dados contendo os dois tipos de imagens


Cachorro


Cachorro


Gato


Gato


Gato


Gato


Cachorro


Cachorro


Gato


Gato


Cachorro


Cachorro

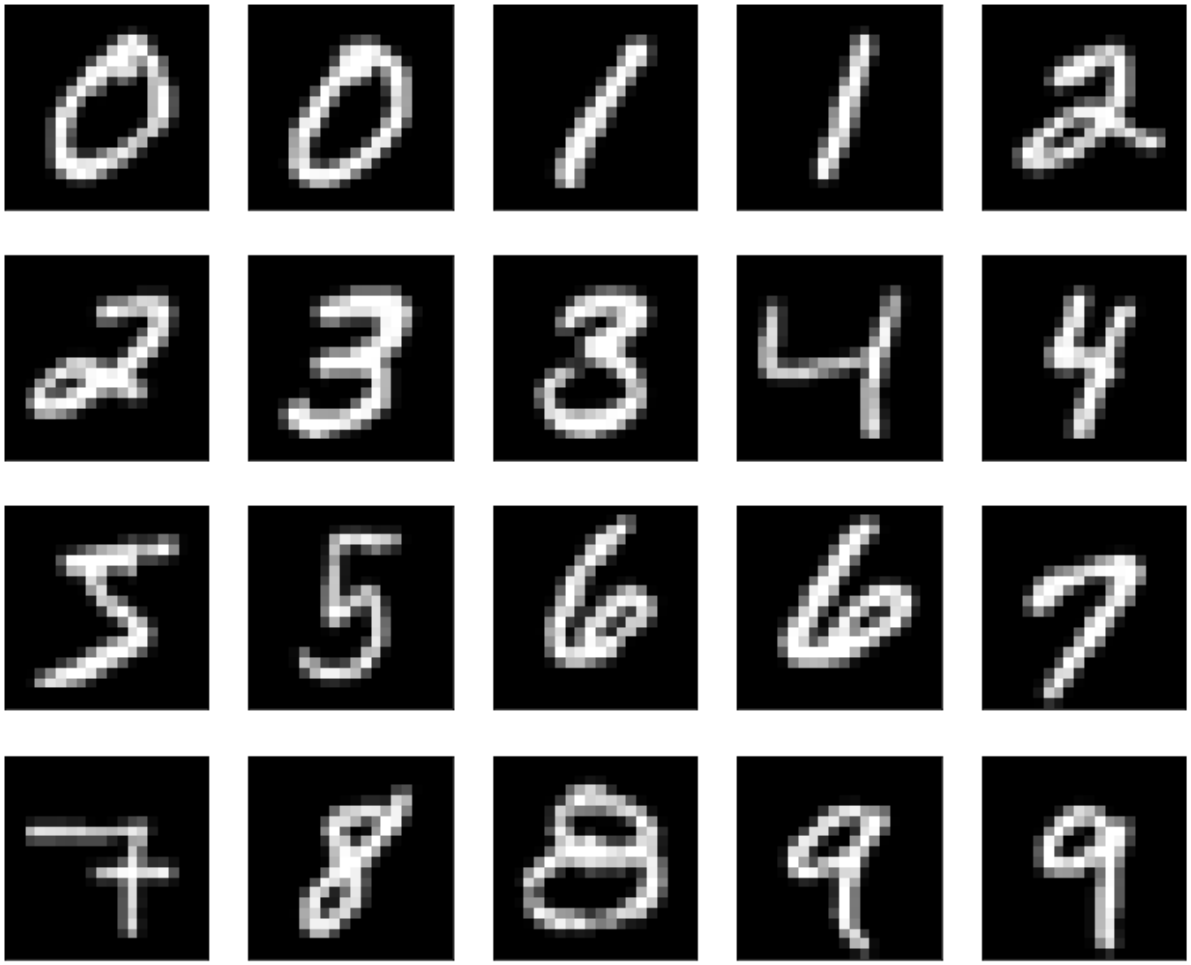
Classificação de imagens supervisionada

- Dependendo do número de possíveis variações das imagens a serem classificadas, pode ser que precisemos de um grande número de imagens
- Adicionalmente, muitas vezes precisamos que as imagens tenham o mesmo tamanho. Se isso não ocorrer, podemos redimensionar as imagens para um tamanho padrão
- O pré-processamento da imagem (normalização de brilho, remoção de ruído, etc) pode ajudar bastante na tarefa de classificação

Classificação de imagens

- Para entendermos a tarefa de classificação de imagens, utilizaremos a base de imagens MNIST
- Essa base possui um grande número de imagens de dígitos escritos manualmente
- As imagens possuem tamanho 20x20

Classificação de imagens

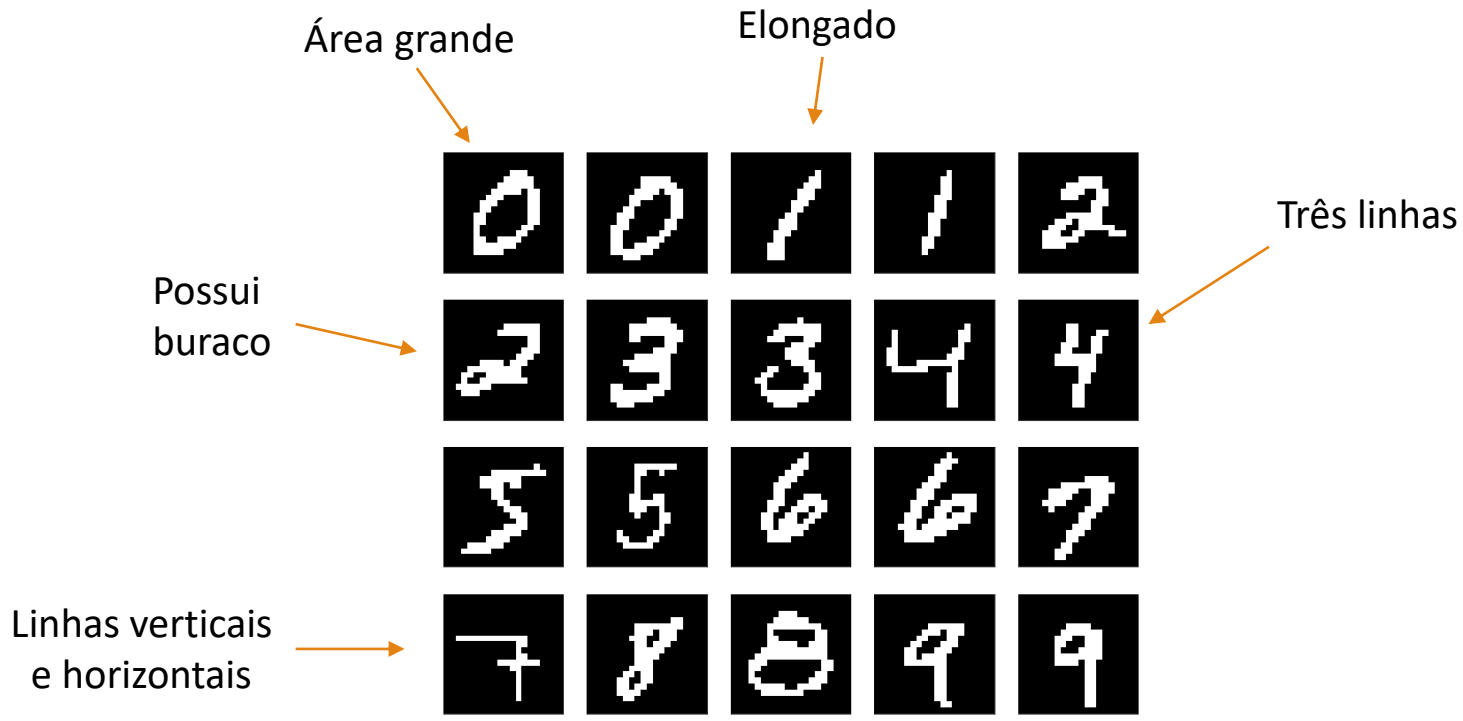


Classificação de imagens

- O primeiro passo é definirmos e extrairmos atributos para cada imagem
- Uma coleção de atributos extraídos para uma dada imagem é chamada de vetor de atributos

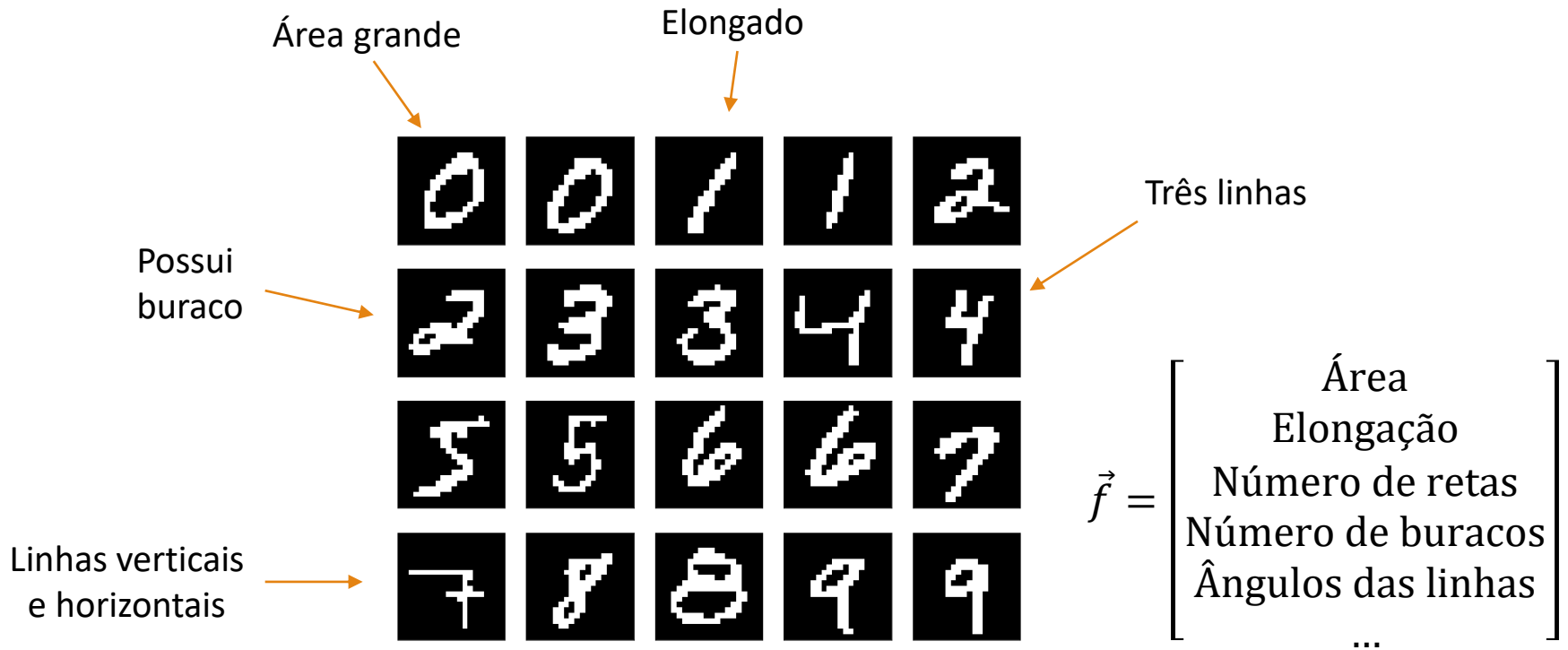
Classificação de imagens – Engenharia de atributos

- Uma abordagem para caracterizar os dígitos poderia ser limiarizar a imagem e então extrair algumas propriedades de forma



Classificação de imagens – Engenharia de atributos

- Uma abordagem para caracterizar os dígitos poderia ser limiarizar a imagem e então extrair algumas propriedades de forma



Classificação de imagens – Geração automática de atributos

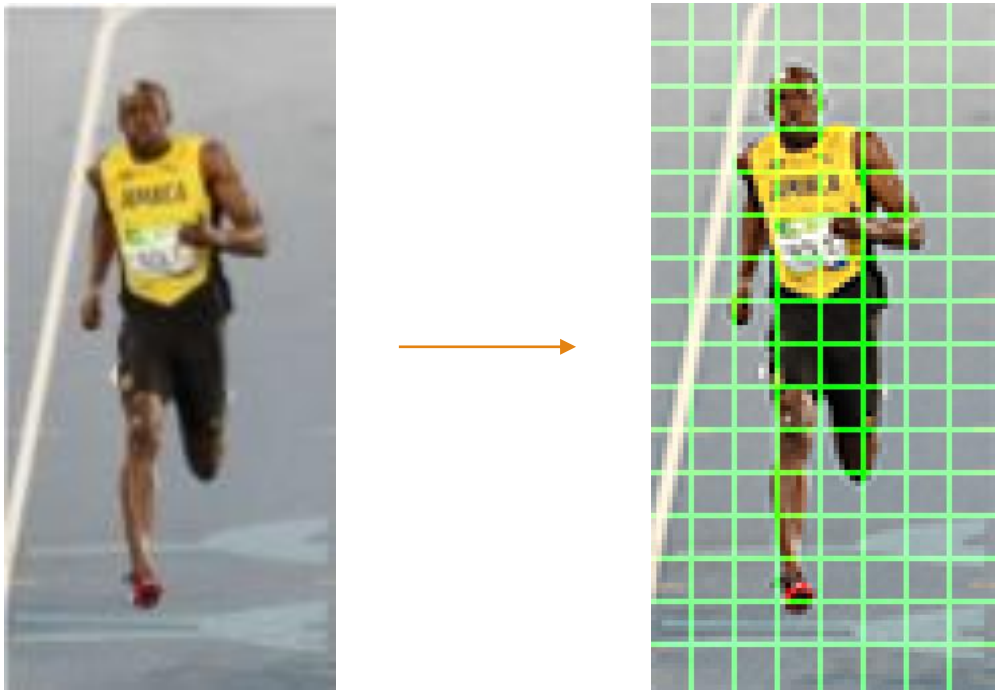
- Mas iremos estudar uma forma mais geral de extrair atributos, chamada Histogram of Oriented Gradients

Conceitos básicos do Histogram of Oriented Gradients - HOG

- O descritor Histogram of Oriented Gradients (HOG) é uma metodologia que pode ser utilizada para associar um grande número de propriedades a uma dada imagem
- As propriedades são definidas de forma automática, utilizando o ângulo do vetor gradiente obtido em diferentes regiões da imagem
- O gradiente é utilizado com o objetivo de criar um descritor que seja aproximadamente invariante à diferenças nas intensidade dos pixels

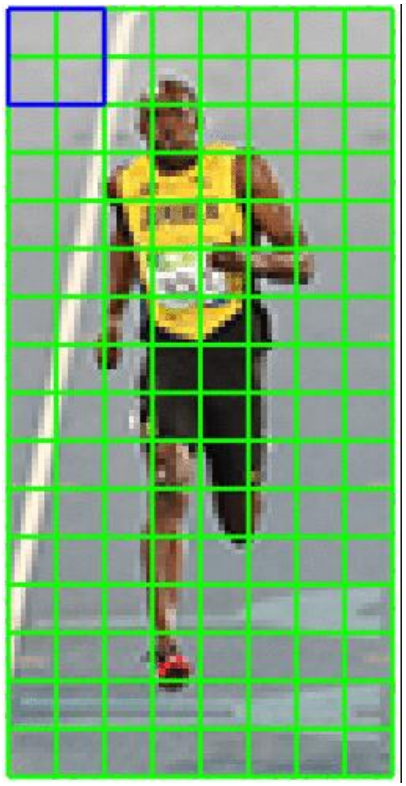
Conceitos básicos do Histogram of Oriented Gradients - HOG

- A imagem é dividida em células
- Histogramas dos ângulos dos vetores gradientes dos pixels são calculados para cada célula



HOG

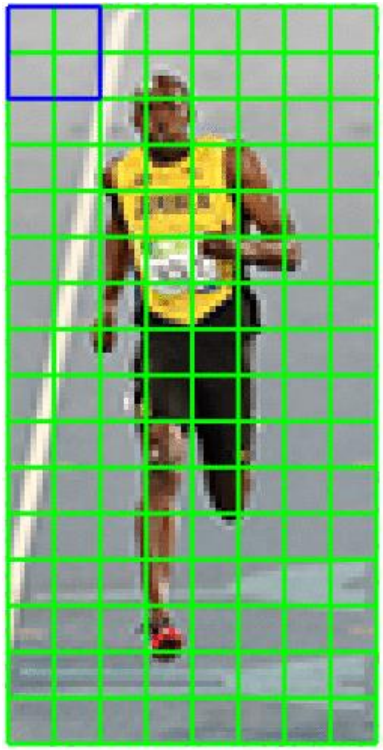
- As células são então agrupadas em blocos (quadrado azul na imagem abaixo)
- Os histogramas das células de cada bloco são concatenados para formar $n \times n \times ncaixas$ atributos associados com cada bloco, onde n é o número de células dentro do bloco e $ncaixas$ o número de caixas utilizado para o histograma
- Os valores são normalizados pelas soma de cada bloco



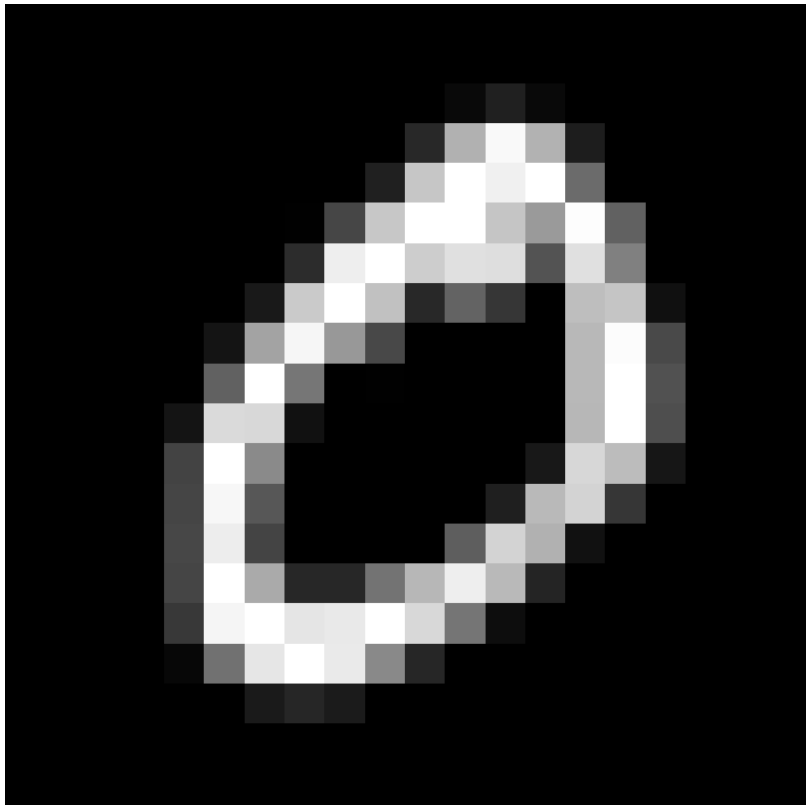
HOG

- As células são então agrupadas em blocos (quadrado azul na imagem abaixo)
- Os histogramas das células de cada bloco são concatenados para formar $n \times n \times ncaixas$ atributos associados com cada bloco, onde n é o número de células dentro do bloco e $ncaixas$ o número de caixas utilizado para o histograma
- Os valores são normalizados pelas soma de cada bloco

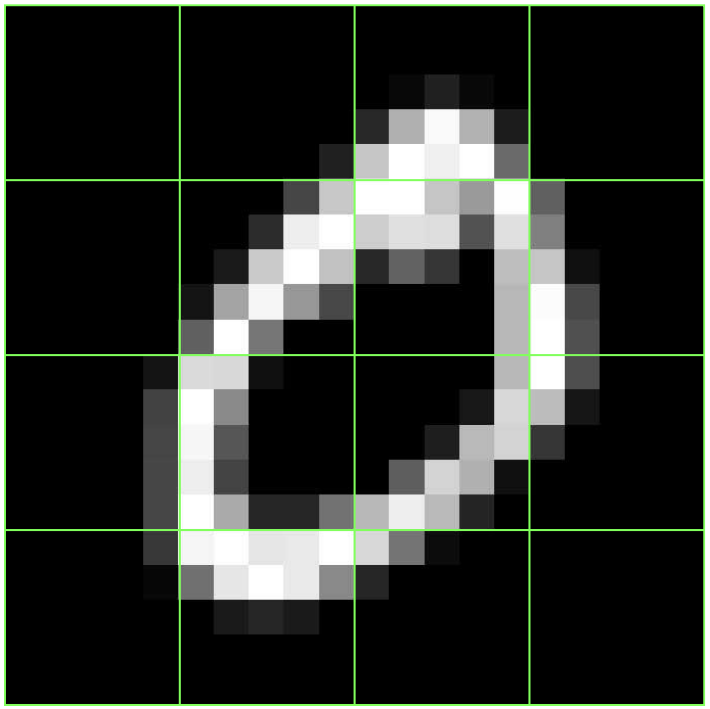
- A janela de bloco é movida, e o processo é repetido



Cálculo do HOG em uma imagem de dígito

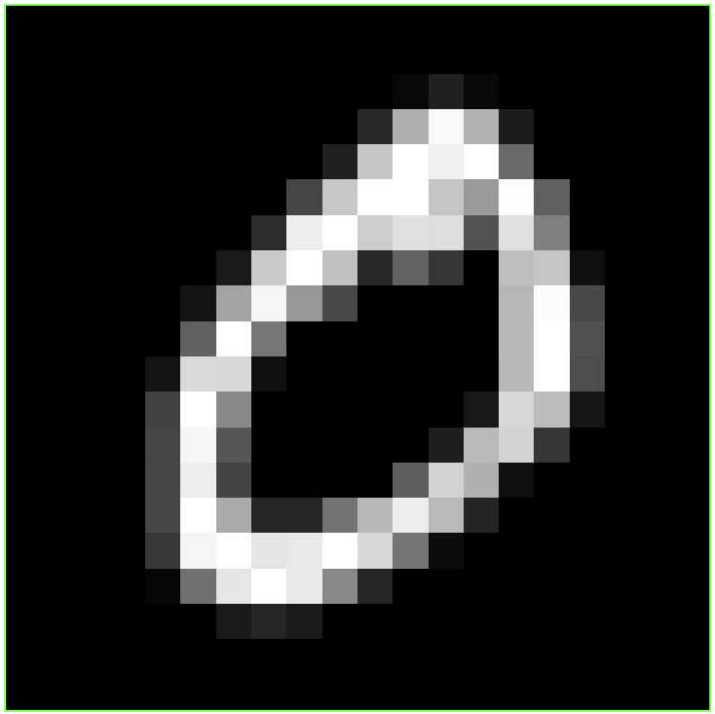


Cálculo do HOG em uma imagem de dígito



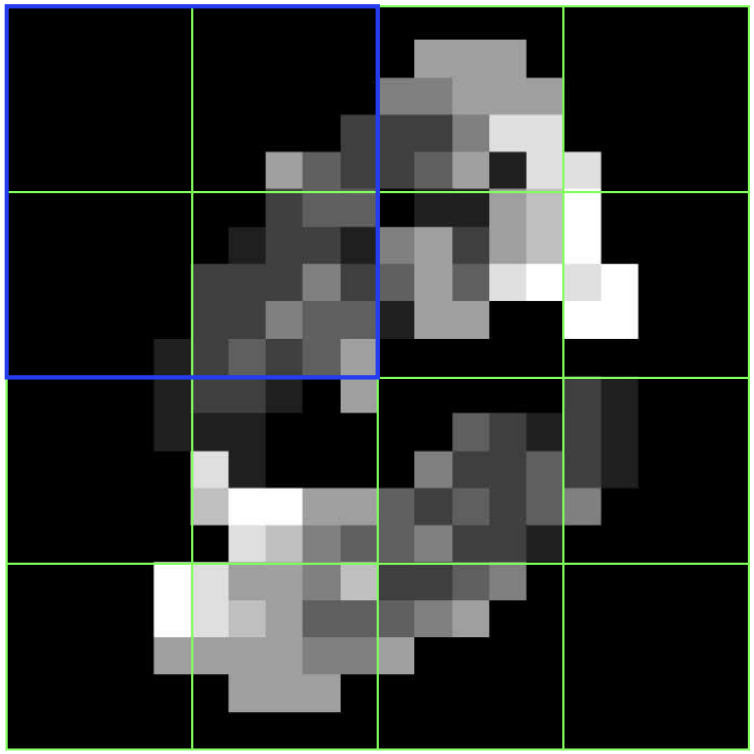
A imagem é dividida em células de tamanho 5×5 pixels (o tamanho da célula é um parâmetro do método)

Cálculo do HOG em uma imagem de dígito



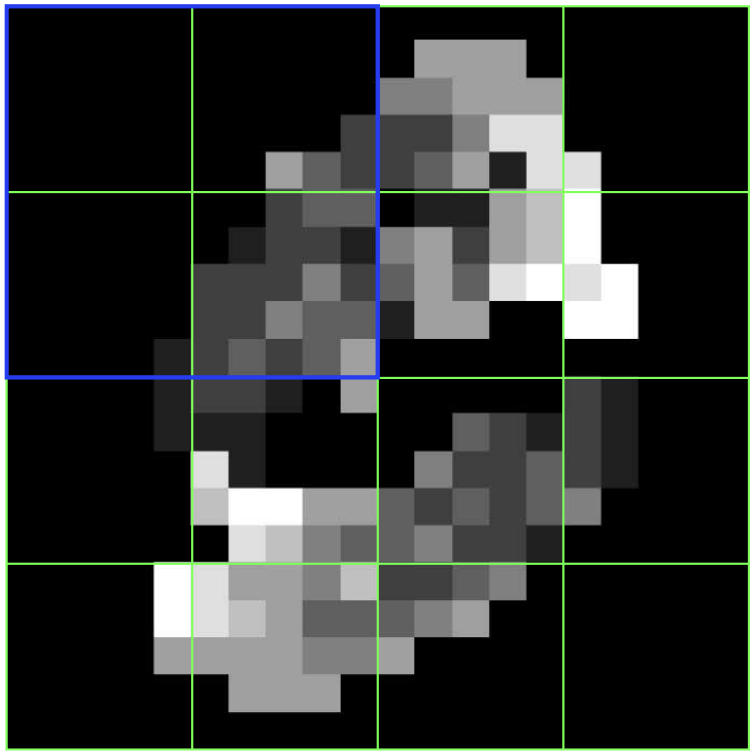
Células são agrupadas em blocos de tamanho 10×10 pixels (parâmetro do método)

Cálculo do HOG em uma imagem de dígito



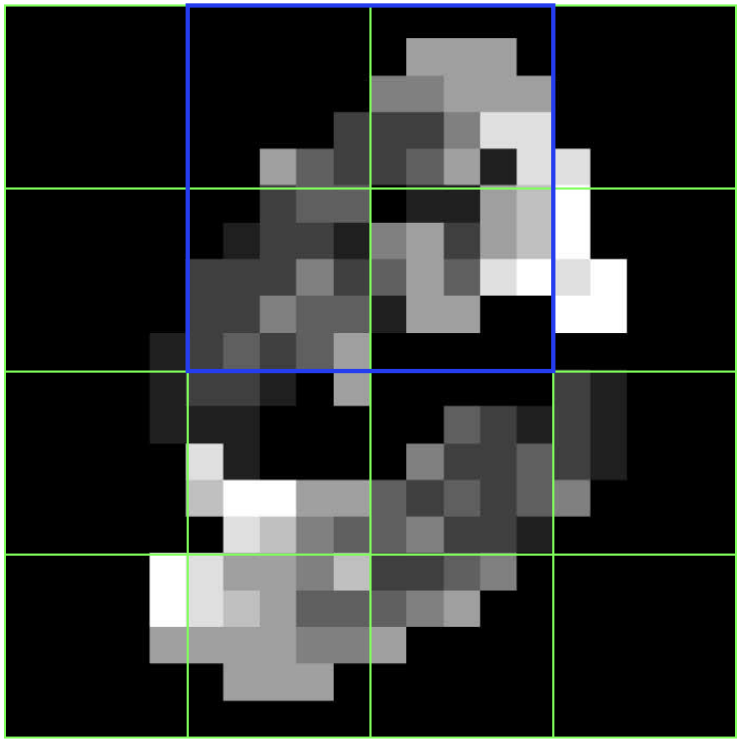
- Os ângulos dos vetores gradientes são calculados para cada pixel

Cálculo do HOG em uma imagem de dígito



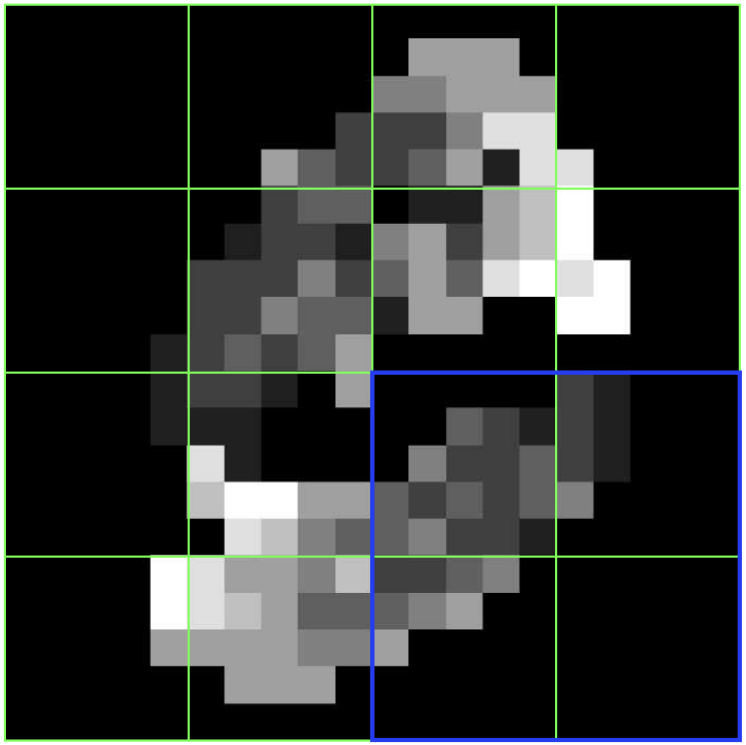
- Um histograma é obtido para cada célula (usualmente 9 caixas são utilizadas para o histograma)
- Os histogramas das células no bloco azul são concatenados e normalizados pela soma de todos os valores

Cálculo do HOG em uma imagem de dígito



O bloco é movido 5 pixels para a direita, e o histograma normalizado é calculado novamente

Cálculo do HOG em uma imagem de dígito

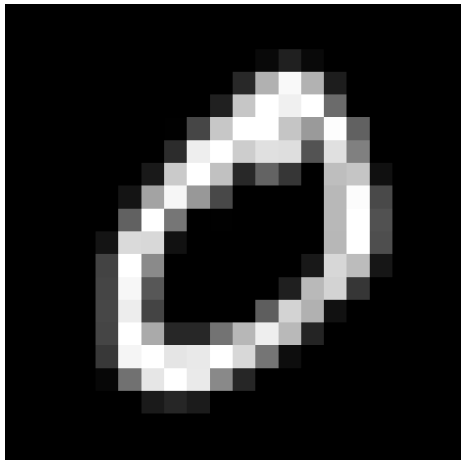


E assim por diante, até o último bloco

Cálculo do HOG em uma imagem de dígito

- Nós temos 9 blocos azuis na imagem, cada um possuindo 4×9 valores de histograma (número de células em cada bloco vezes o número de caixas utilizadas para o histograma).
- Portanto, temos 324 atributos calculados para a imagem

Cálculo do HOG em uma imagem de dígito



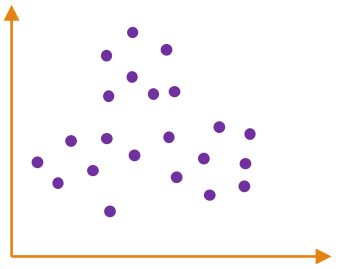
HOG

[0.001, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.001, 0.052, 0.219, 0.055, 0.000, 0.000, 0.000, 0.000, 0.000, 0.009, 0.171, 0.415, 0.235, 0.066, 0.006, 0.000, 0.000, 0.006, 0.026, 0.318, 0.532, 0.532, 0.091, 0.000, 0.000, 0.000, 0.024, 0.047, 0.041, 0.248, 0.081, 0.000, 0.000, 0.000, 0.000, 0.000, 0.008, 0.247, 0.392, 0.129, 0.002, 0.003, 0.000, 0.000, 0.001, 0.115, 0.071, 0.392, 0.392, 0.016, 0.000, 0.000, 0.000, 0.002, 0.009, 0.231, 0.392, 0.392, 0.012, 0.012, 0.007, 0.000, 0.005, 0.086, 0.253, 0.216, 0.035, 0.001, 0.005, 0.003, 0.052, 0.062, 0.253, 0.182, 0.013, 0.000, 0.000, 0.050, 0.079, 0.253, 0.162, 0.244, 0.253, 0.253, 0.185, 0.092, 0.122, 0.035, 0.129, 0.068, 0.242, 0.093, 0.039, 0.167, 0.253, 0.253, 0.253, 0.253, 0.144, 0.198, 0.068, 0.219, 0.169, 0.183, 0.101, 0.001, 0.000, 0.006, 0.008, 0.092, 0.285, 0.285, 0.250, 0.062, 0.000, 0.000, 0.025, 0.020, 0.027, 0.097, 0.130, 0.257, 0.285, 0.218, 0.203, 0.030, 0.003, 0.065, 0.215, 0.285, 0.285, 0.258, 0.092, 0.209, 0.165, 0.107, 0.056, 0.329, 0.329, 0.147, 0.034, 0.000, 0.000, 0.008, 0.009, 0.099, 0.329, 0.329, 0.041, 0.022, 0.005, 0.000, 0.002, 0.027, 0.119, 0.255, 0.322, 0.262, 0.088, 0.012, 0.040, 0.109, 0.150, 0.091, 0.329, 0.329, 0.038, 0.021, 0.000, 0.000, 0.010, 0.058, 0.040, 0.170, 0.353, 0.154, 0.084, 0.021, 0.030, 0.007, 0.040, 0.004, 0.027, 0.353, 0.353, 0.283, 0.077, 0.068, 0.009, 0.024, 0.004, 0.353, 0.353, 0.097, 0.009, 0.006, 0.000, 0.000, 0.003, 0.000, 0.111, 0.353, 0.260, 0.041, 0.011, 0.001, 0.000, 0.000, 0.019, 0.010, 0.021, 0.070, 0.215, 0.266, 0.288, 0.172, 0.006, 0.170, 0.029, 0.126, 0.219, 0.151, 0.136, 0.288, 0.288, 0.288, 0.022, 0.000, 0.000, 0.000, 0.008, 0.033, 0.212, 0.181, 0.020, 0.288, 0.000, 0.000, 0.000, 0.002, 0.037, 0.220, 0.288, 0.288, 0.282, 0.218, 0.072, 0.065, 0.025, 0.015, 0.049, 0.214, 0.305, 0.230, 0.305, 0.305, 0.003, 0.003, 0.000, 0.000, 0.018, 0.128, 0.305, 0.122, 0.014, 0.000, 0.000, 0.007, 0.016, 0.125, 0.305, 0.304, 0.305, 0.163, 0.000, 0.000, 0.000, 0.000, 0.008, 0.184, 0.069, 0.452, 0.452, 0.033, 0.000, 0.002, 0.000, 0.000, 0.021, 0.004, 0.221, 0.452, 0.097, 0.014, 0.003, 0.000, 0.000, 0.003, 0.114, 0.452, 0.293, 0.003, 0.000, 0.000, 0.000, 0.000, 0.046, 0.006, 0.064, 0.079, 0.005, 0.000, 0.000, 0.000, 0.000, 0.004]

Cálculo do HOG em uma imagem de dígito

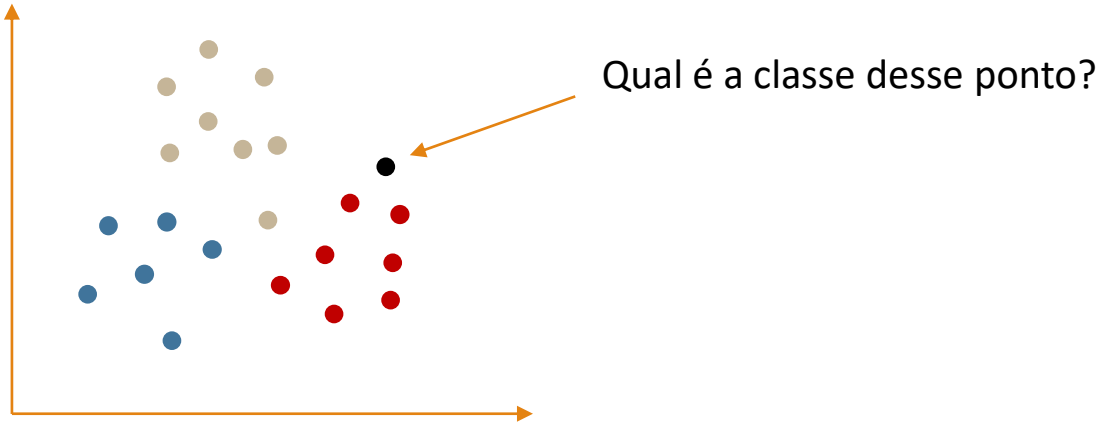
Temos 324 atributos. Podemos agora pensar que cada imagem é representada por um ponto em um espaço de 324 dimensões!

Se tivéssemos apenas 2 atributos:



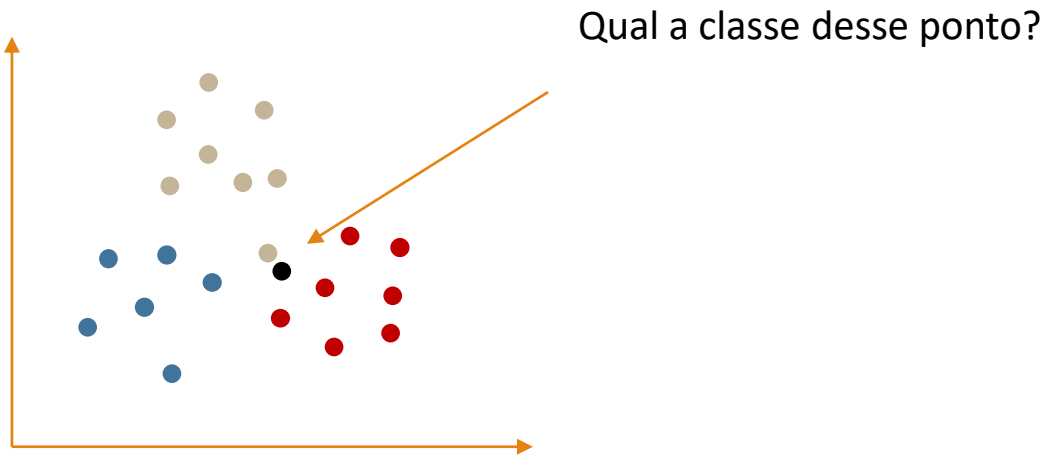
Classificador k-vizinhos

- Um dos classificadores mais simples que podemos utilizar é o k-vizinhos.
- Para um dado ponto que queremos classificar (representando uma imagem com classe desconhecida), analisamos os k pontos mais próximos (k-vizinhos).
- A classe mais frequente nos k pontos analisados é associada ao ponto desconhecido.



Classificador k-vizinhos

- Um dos classificadores mais simples que podemos utilizar é o k-vizinhos.
- Para um dado ponto que queremos classificar (representando uma imagem com classe desconhecida), analisamos os k pontos mais próximos (k-vizinhos).
- A classe mais frequente nos k pontos analisados é associada ao ponto desconhecido.

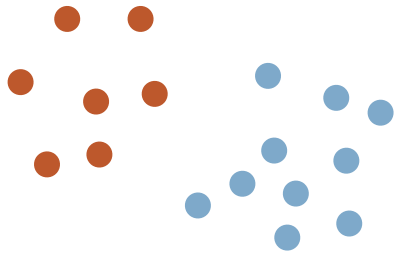


Support Vector Machine (SVM)

- Outro classificador muito utilizado em processamento de imagens é o SVM

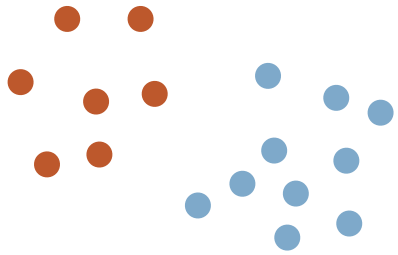
Support Vector Machine (SVM)

Suponha que temos duas classes que são linearmente separáveis:

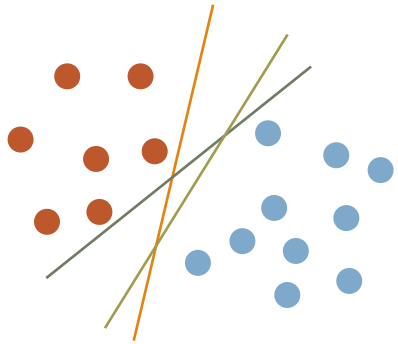


Support Vector Machine (SVM)

Suponha que temos duas classes que são linearmente separáveis:

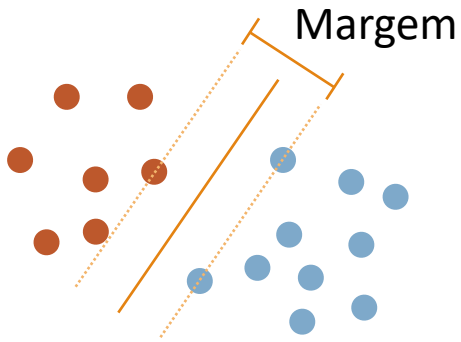


Nesse caso, temos diversas possibilidades para definir a função de separação entre as classes



SVM

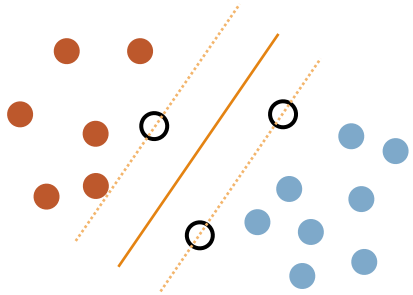
O classificador SVM consiste em selecionar a função de separação que maximiza a margem, definida como a distância entre a reta de decisão e o ponto mais próximo à reta

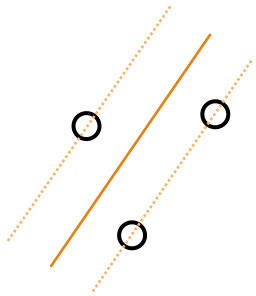


SVM

Uma das principais vantagens do SVM é a sua esparsidade

Support vectors

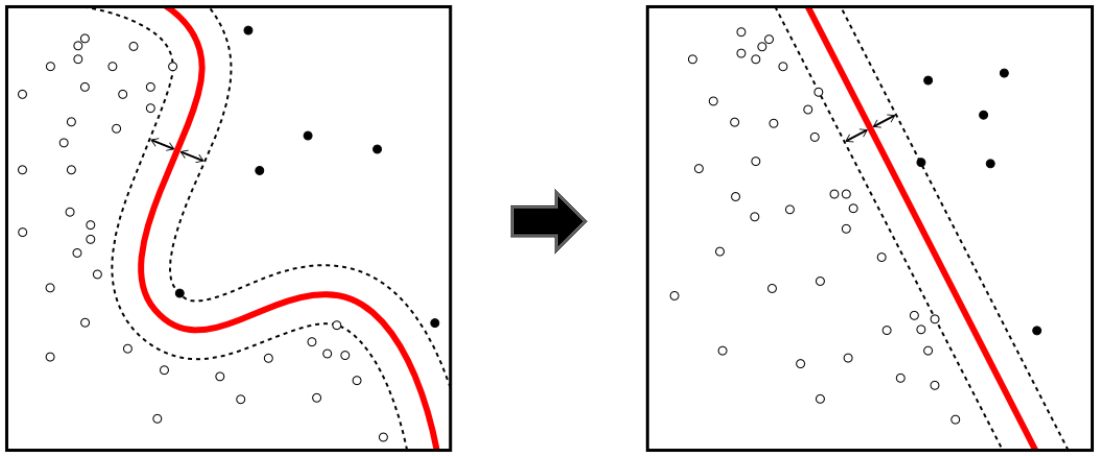




Apenas alguns pontos, chamados de **support vectors**, são necessários para definir a função de decisão e, portanto, para classificar novas imagens

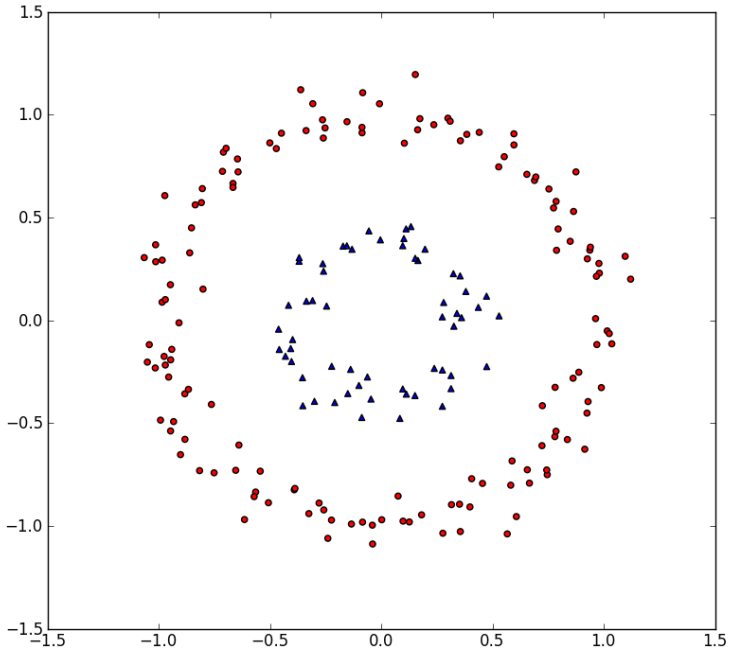
SVM

- Para classes que não são linearmente separáveis, o chamado truque de kernel (kernel trick) é utilizado
- Essa técnica corresponde a aplicar uma transformação não-linear aos atributos, definindo um novo espaço de atributos no qual as classes são linearmente separáveis



SVM – Exemplo de kernel trick

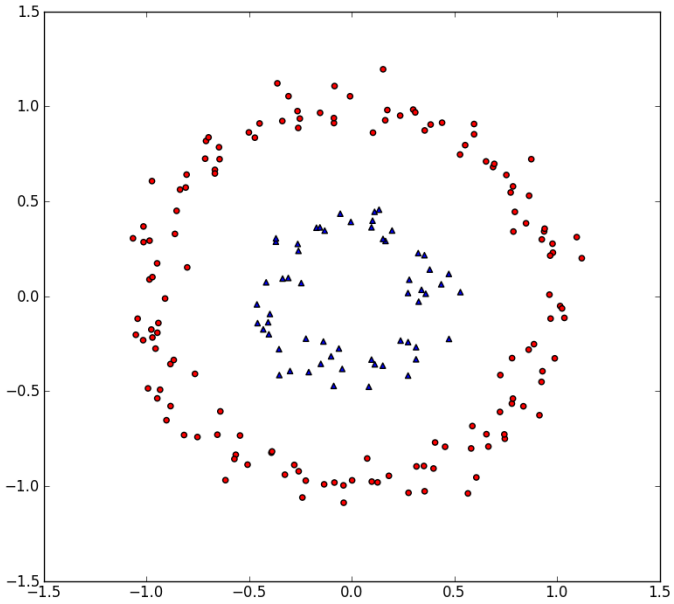
Os pontos não são linearmente separáveis

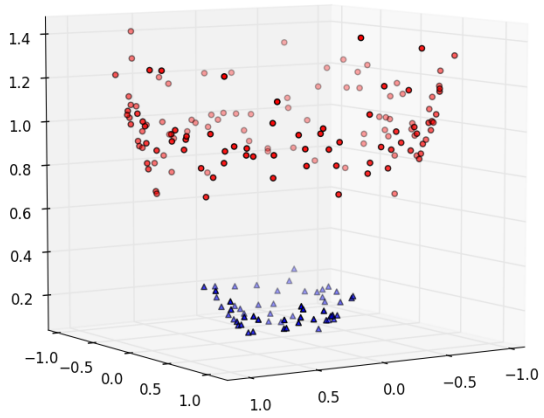


http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

SVM – Exemplo de kernel trick

Um novo atributo (por exemplo, a distância do ponto à origem), em conjunto com os dois atributos originais, resulta em uma distribuição de pontos que é linearmente separável

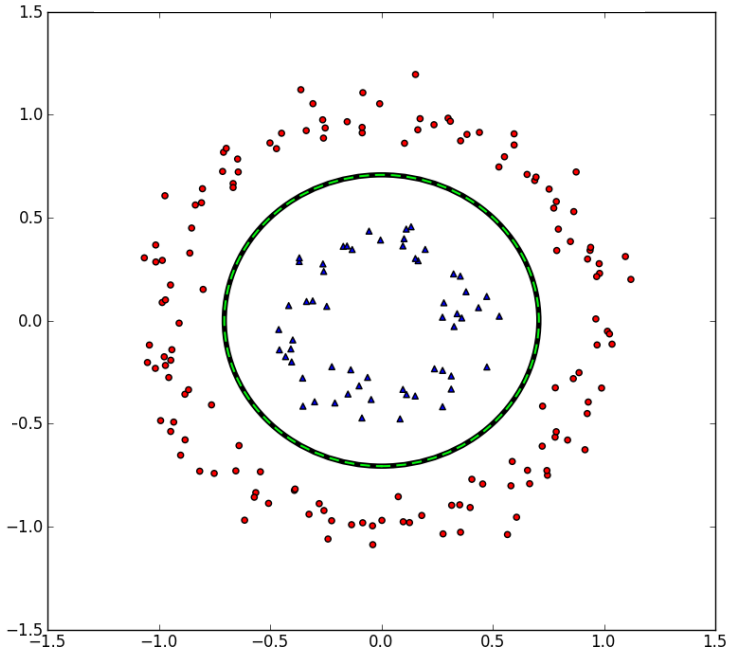


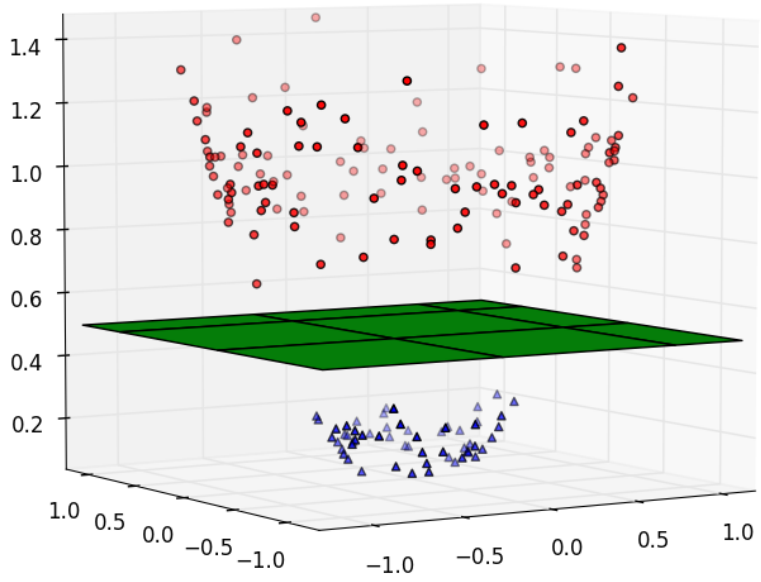


http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

SVM – Exemplo de kernel trick

Um novo atributo (por exemplo, a distância do ponto à origem), em conjunto com os dois atributos originais, resulta em uma distribuição de pontos que é linearmente separável





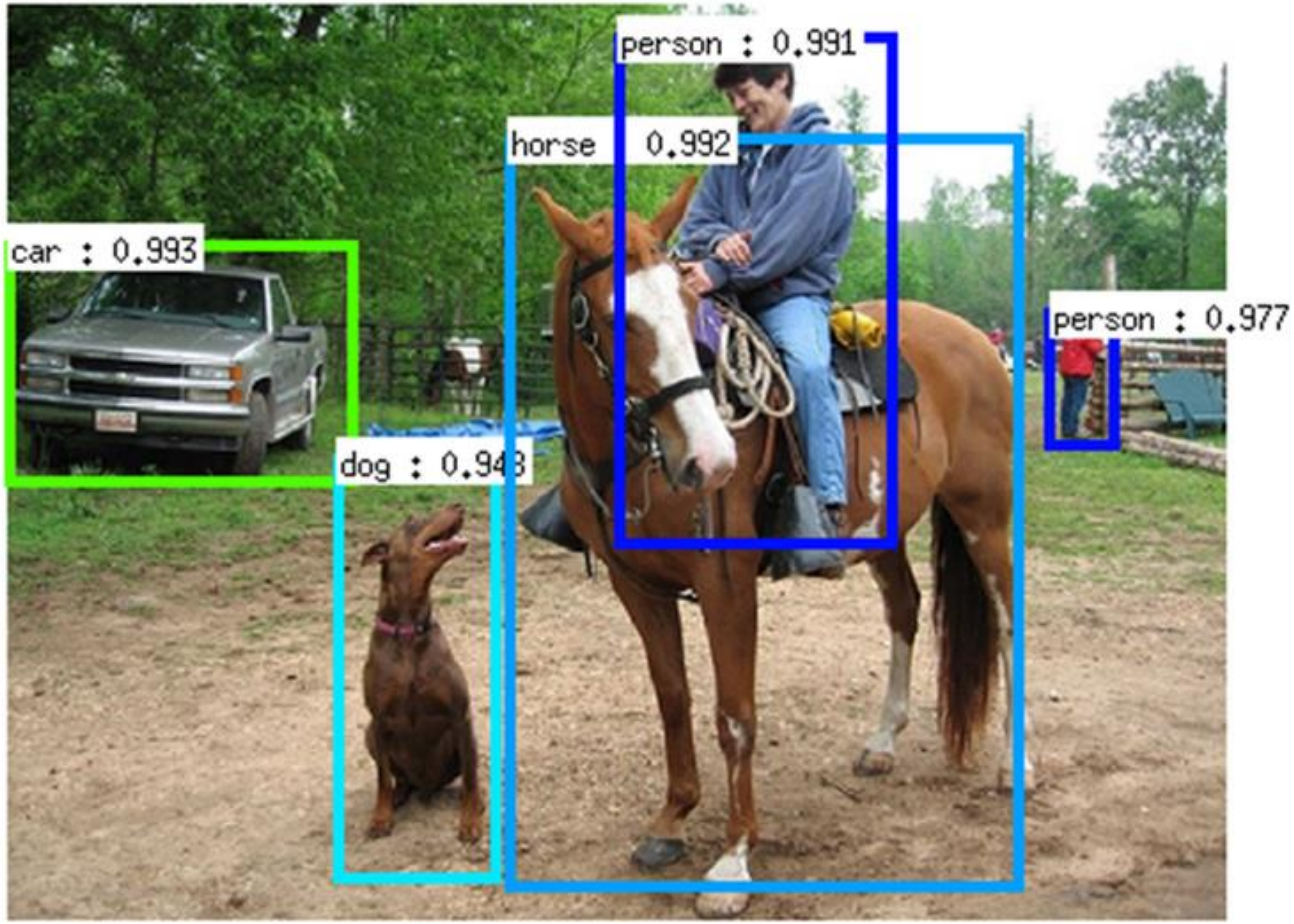
http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Classificação de dígitos utilizando o HOG e k-vizinhos

Notebook “**Classificação de dígitos utilizando HOG e k-vizinhos**”

Detecção e classificação de objetos

Detecção de objetos



Detecção Utilizando Janela Deslizante

Detecção Utilizando Janela Deslizante

Queremos encontrar todas as pessoas nessa cena

Detecção Utilizando Janela Deslizante

A estratégia é deslizar uma janela sobre a imagem, e tentar classificar cada imagem sob a janela, verificando se há uma pessoa na janela

Detecção Utilizando Janela Deslizante

A estratégia é deslizar uma janela sobre a imagem, e tentar classificar cada imagem sob a janela, verificando se há uma pessoa na janela

Detecção Utilizando Janela Deslizante

A estratégia é deslizar uma janela sobre a imagem, e tentar classificar cada imagem sob a janela, verificando se há uma pessoa na janela

Detecção Utilizando Janela Deslizante

A estratégia é deslizar uma janela sobre a imagem, e tentar classificar cada imagem sob a janela, verificando se há uma pessoa na janela

Detecção Utilizando Janela Deslizante

A estratégia é deslizar uma janela sobre a imagem, e tentar classificar cada imagem sob a janela, verificando se há uma pessoa na janela

Detecção Utilizando Janela Deslizante

A estratégia é deslizar uma janela sobre a imagem, e tentar classificar cada imagem sob a janela, verificando se há uma pessoa na janela

Detecção Utilizando Janela Deslizante

A estratégia é deslizar uma janela sobre a imagem, e tentar classificar cada imagem sob a janela, verificando se há uma pessoa na janela

Detecção Utilizando Janela Deslizante

As pessoas possuem diferentes tamanhos, como podemos tratar essa variação de tamanho?

Detecção Utilizando Janela Deslizante

As pessoas possuem diferentes tamanhos, como podemos tratar essa variação de tamanho?
Diminuímos a imagem e repetimos a busca. O tamanho da janela é mantido fixo.

Detecção Utilizando Janela Deslizante

Aplicamos a janela deslizante em cada nível da pirâmide multiresolução da imagem

Detecção Utilizando Janela Deslizante

Para classificarmos as imagens encontradas em cada janela, precisamos primeiro treinar um classificador
Para isso, criamos uma base de imagens de treinamento

Imagens de treinamento contendo pessoas (exemplos positivos)

...

Imagens de treinamento sem pessoas (exemplos negativos)

...

Detecção Utilizando Janela Deslizante

- Na sequência, definimos e calculamos propriedades para caracterizar cada imagem da base.
- Um classificador é treinado nas propriedades calculadas

Detecção Utilizando Janela Deslizante

Dada uma nova imagem, na qual queremos detectar pessoas:

- Para cada região sob a janela deslizante, classificamos a região utilizando o classificador treinado na base de imagens
- No final do procedimento, temos associado a cada região analisada a informação “pessoa” ou “não pessoa”

Detecção Utilizando Janela Deslizante

- O resultado, em geral, inclui uma grande quantidade de janelas, pois diversas janelas próximas acabam sendo associadas a cada pessoa

Detecção Utilizando Janela Deslizante

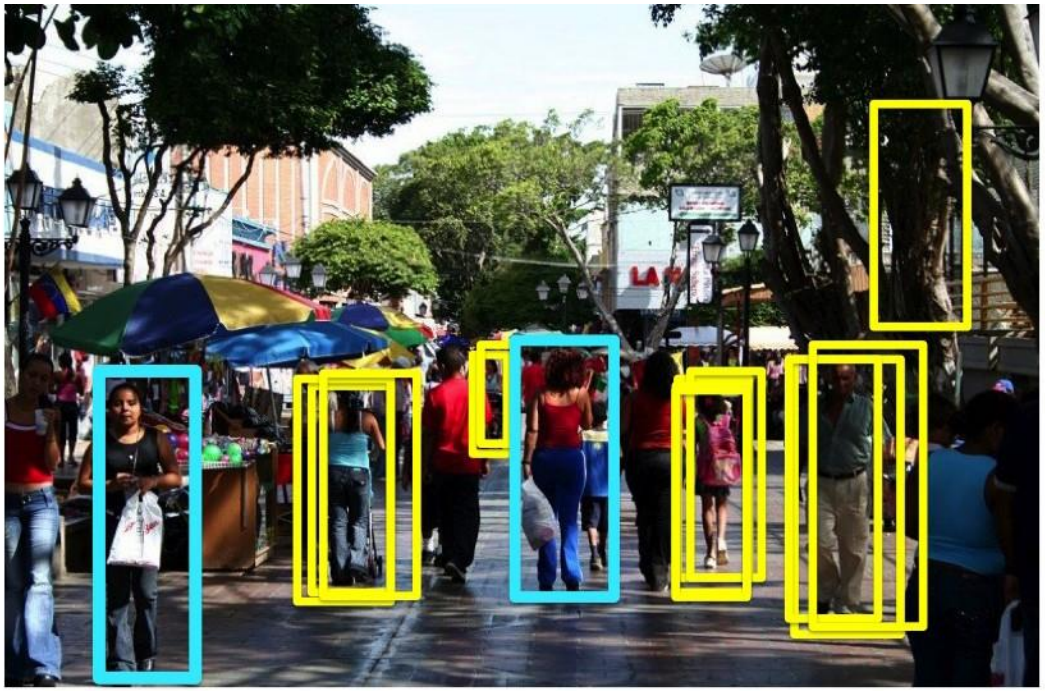
- Podemos utilizar diferentes critérios para eliminar janelas redundantes
- Um critério muito comum:
  1. Obtenha a janela J com maior probabilidade de conter uma pessoa
  2. Elimine as janelas possuindo intersecção com J maior do que um limiar
  3. Armazene J numa lista final de janelas e remova-a da imagem
  4. Repita 1 a 3 até que não haja mais janelas a serem analisadas

Detecção Utilizando Janela Deslizante

- Podemos utilizar diferentes critérios para eliminar janelas redundantes
- Um critério muito comum:
  1. Obtenha a janela J com maior probabilidade de conter uma pessoa
  2. Elimine as janelas possuindo intersecção com J maior do que um limiar
  3. Armazene J numa lista final de janelas e remova-a da imagem
  4. Repita 1 a 3 até que não haja mais janelas a serem analisadas

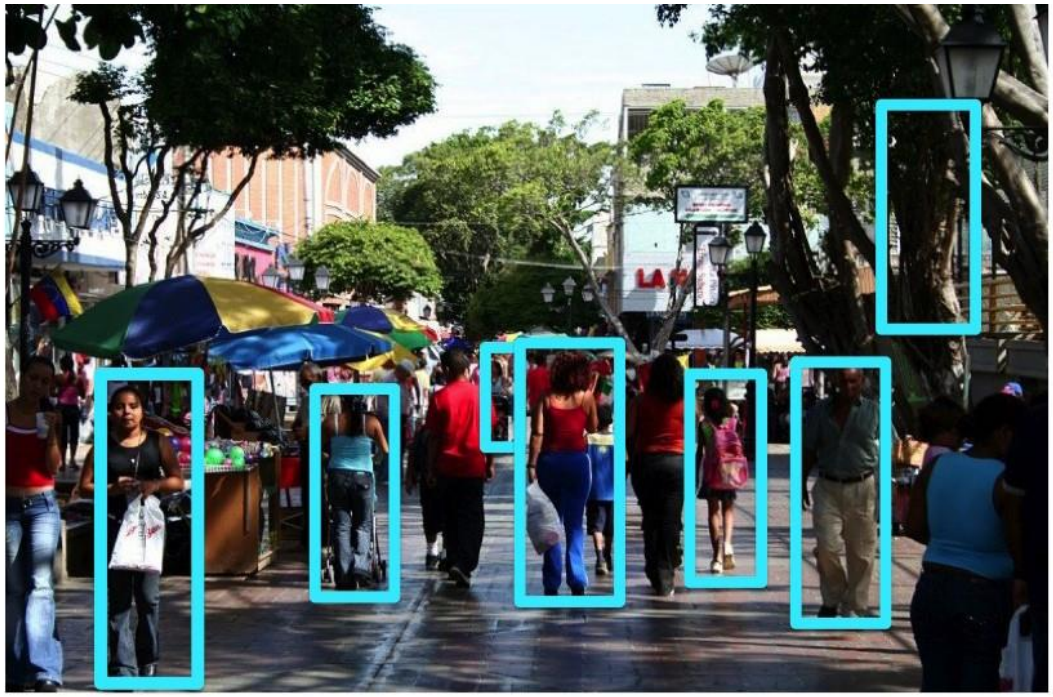
Detecção Utilizando Janela Deslizante

- Podemos utilizar diferentes critérios para eliminar janelas redundantes
- Um critério muito comum:
  1. Obtenha a janela J com maior probabilidade de conter uma pessoa
  2. Elimine as janelas possuindo intersecção com J maior do que um limiar
  3. Armazene J numa lista final de janelas e remova-a da imagem
  4. Repita 1 a 3 até que não haja mais janelas a serem analisadas



Detecção Utilizando Janela Deslizante

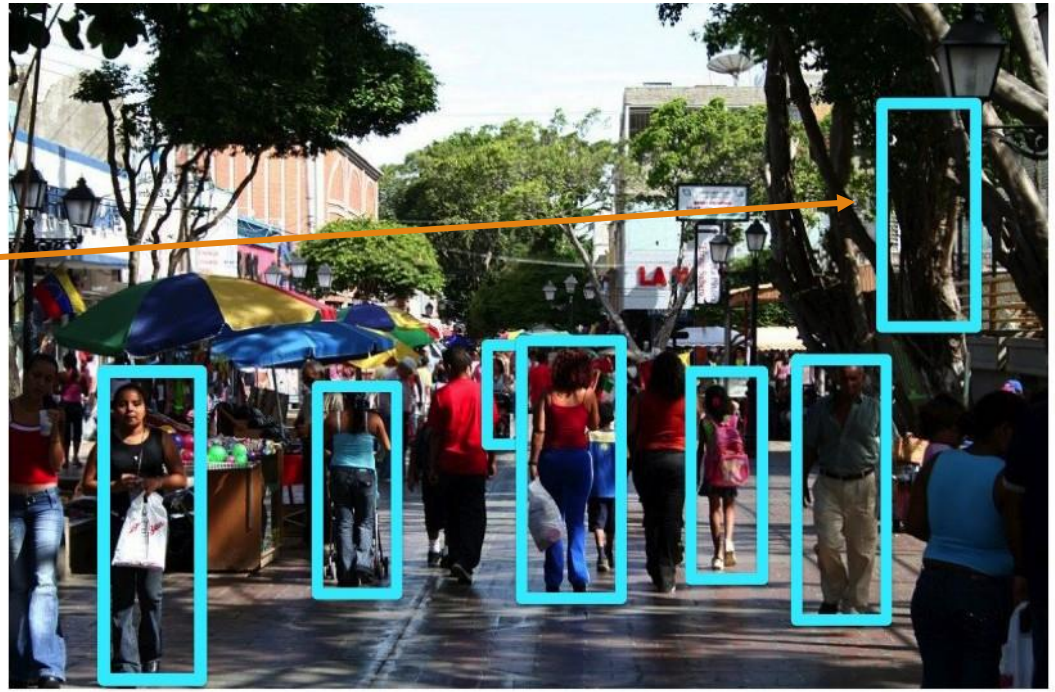
- Podemos utilizar diferentes critérios para eliminar janelas redundantes
- Um critério muito comum:
  1. Obtenha a janela J com maior probabilidade de conter uma pessoa
  2. Elimine as janelas possuindo intersecção com J maior do que um limiar
  3. Armazene J numa lista final de janelas e remova-a da imagem
  4. Repita 1 a 3 até que não haja mais janelas a serem analisadas



Detecção Utilizando Janela Deslizante

- Podemos utilizar diferentes critérios para eliminar janelas redundantes
- Um critério muito comum:
  1. Obtenha a janela J com maior probabilidade de conter uma pessoa
  2. Elimine as janelas possuindo intersecção com J maior do que um limiar
  3. Armazene J numa lista final de janelas e remova-a da imagem
  4. Repita 1 a 3 até que não haja mais janelas a serem analisadas

Problema! O que fazer?



Detecção Utilizando Janela Deslizante

- Podemos utilizar diferentes critérios para eliminar janelas redundantes
- Um critério muito comum:
  1. Obtenha a janela J com maior probabilidade de conter uma pessoa
  2. Elimine as janelas possuindo intersecção com J maior do que um limiar
  3. Armazene J numa lista final de janelas e remova-a da imagem
  4. Repita 1 a 3 até que não haja mais janelas a serem analisadas

Uma estratégia é adicionar a imagem no conjunto de imagens de treinamento negativas, e treinar novamente o classificador