

# Segmentation of Characters on Car License Plates

Heitor Rapela Medeiros {hrm}

Diogo Rodrigues da Silva {drs3}

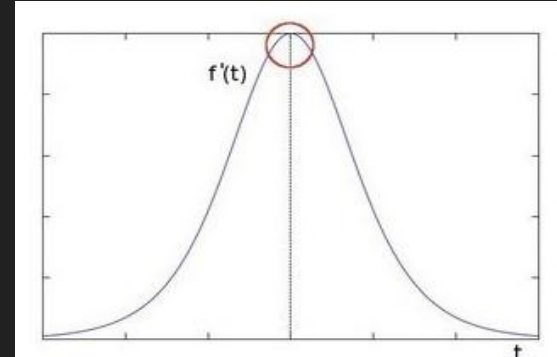
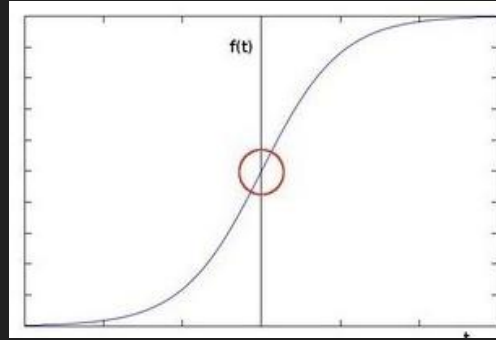
# Roteiro

- Horizontal Projection
  - Máscara de Sobel
  - Método de Otsu
  - Cálculo do histograma de zeros horizontal
- Vertical Projection
  - Binarização através de threshold
  - Cálculo do histograma de zeros vertical
  - Crop da região detectada com zeros
- Support Vector Machine (SVM)
  - Aquisição da base para treino
  - Método para treino

# Horizontal Projection

# Horizontal Projection

- Máscara de Sobel



# Horizontal Projection

- Máscara de Sobel
- Método de Otsu
- Abertura

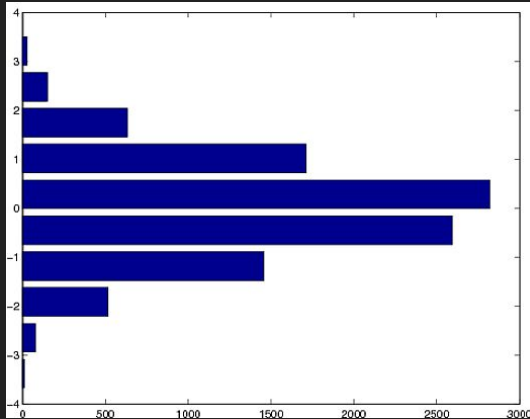
$$G_x = \begin{bmatrix} -3 & 0 & +3 \\ -10 & 0 & +10 \\ -3 & 0 & +3 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ +3 & +10 & +3 \end{bmatrix}$$



# Horizontal Projection

- Cálculo do histograma de zeros horizontal



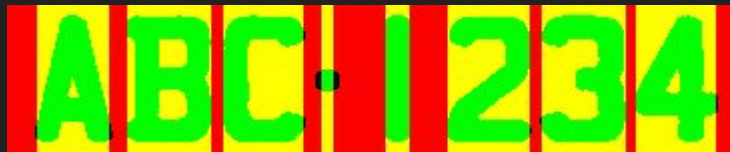
# Vertical Projection

# Vertical Projection

Imagem de entrada



Regiões para serem detectadas





# Threshold

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) \geq k \\ 0, & \text{caso contrario} \end{cases}$$

# Threshold ( $T = 40$ )

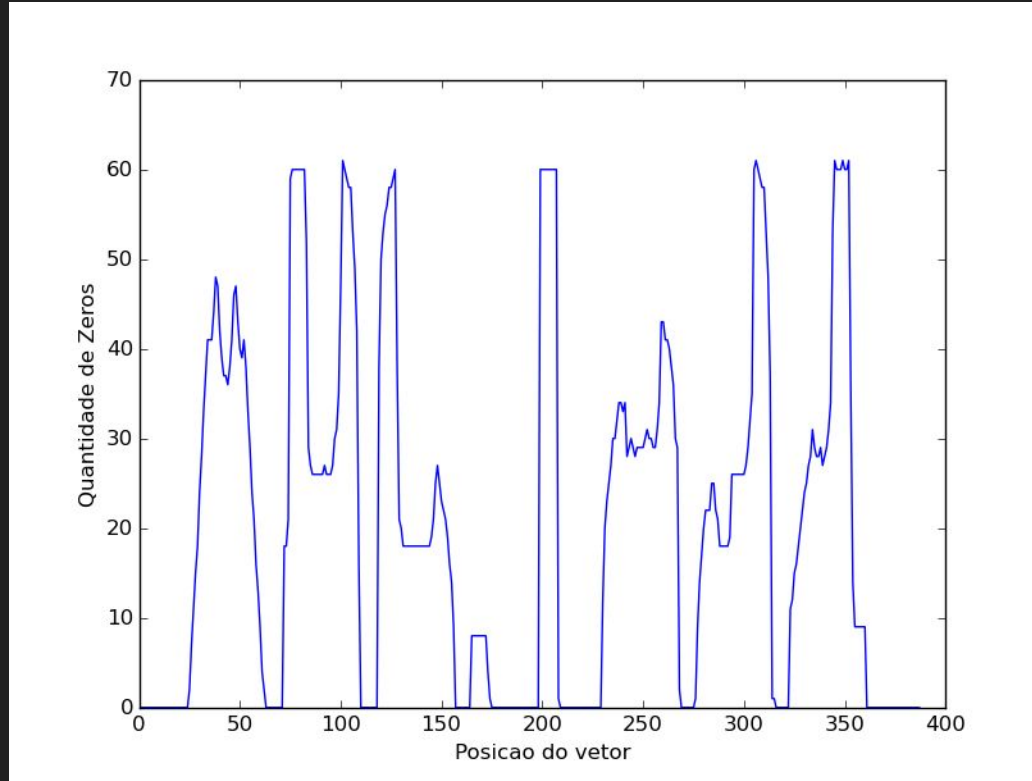
Entrada



Saída

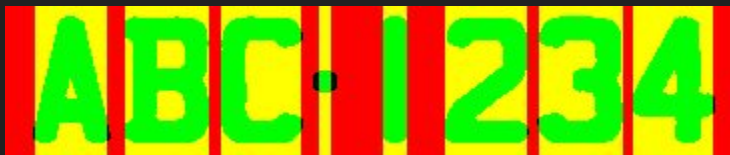


# Histograma de zeros vertical



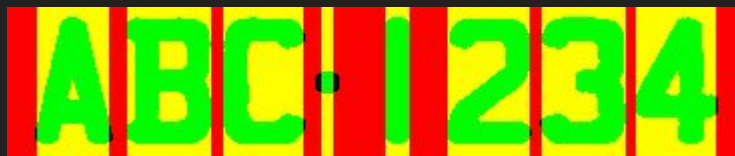
# Crop da região com zeros

- Nesta etapa, se determina a quantidade de zeros que se deve agrupar por colunas, e então calcula-se a posição inicial e final de cada segmento a ser cortado



# Crop da região com zeros

Entrada teórica (na verdade se entra com os valores de início e fim de cada corte)



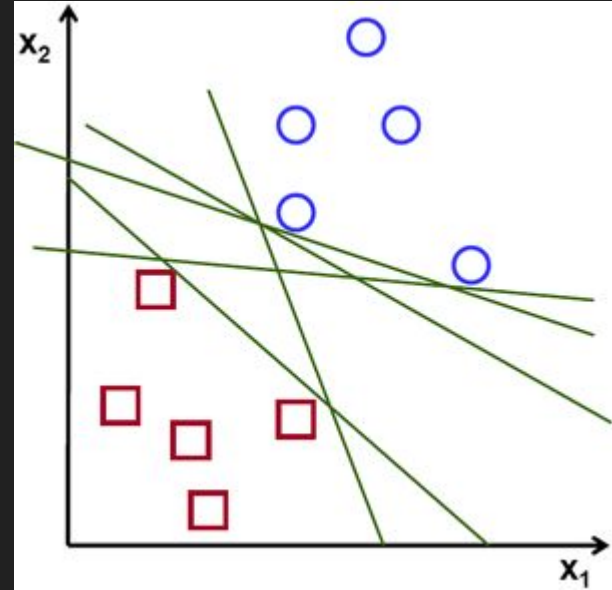
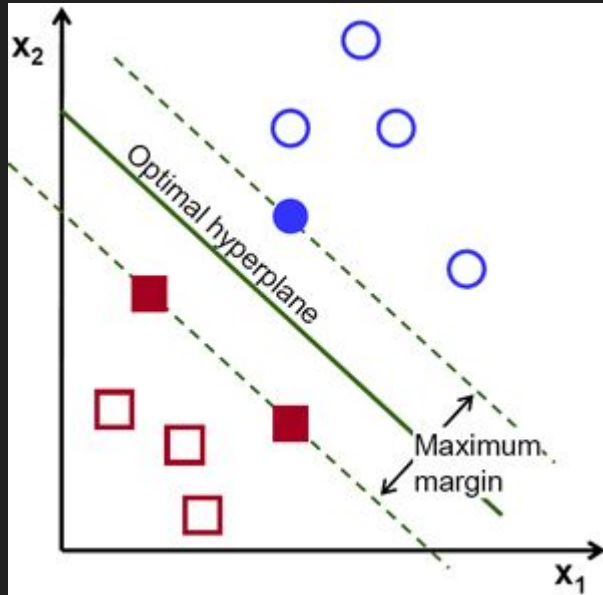
Saída



# Support Vector Machine (SVM)

- O algoritmo SVM tem como base encontrar o hiperplano que divide as classes de treino pela menor distância, tornando a margem entre elas maior possível. O hiperplano considerado ótimo, maximiza a margem de treino das classes de dados.

# Support Vector Machine (SVM)



# Support Vector Machine (SVM)

- Utilização dos coeficientes da DCT gerados a partir de cada conjunto de imagem de treinamento para treinar a SVM multiclasse.

```
C_SVC;
```

```
LINEAR;
```

```
degree = 3;
```

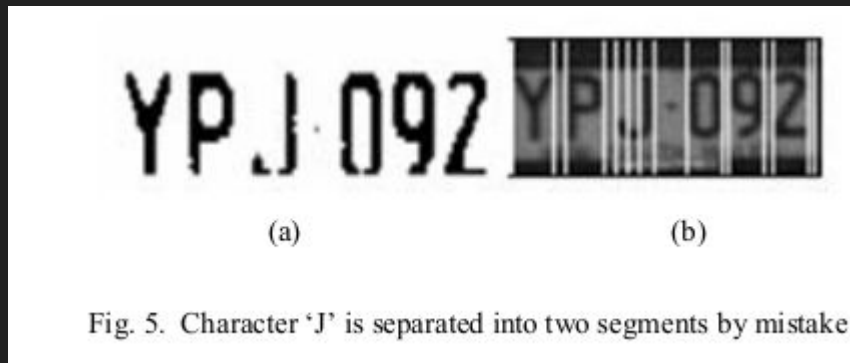
```
gamma = 3;
```

```
term_crit = cvTermCriteria(CV_TERMCRIT_ITER, 100, 1e-6);
```



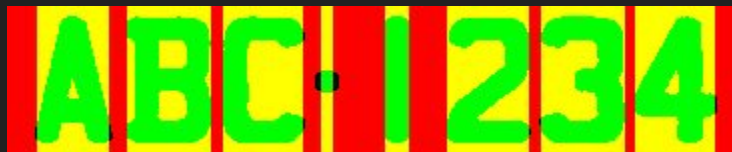
# Testes

- Falhas possíveis na binarização devido a variação na luminosidade



# Testes

- O artigo diz somente a proporção de cada letra e número para o modelo de placa utilizado (americano), por isso fica bem abstrato. Para o modelo brasileiro, colunas de menos de 10 pixels pretos foram desconsideradas, tornando uma aproximação razoável para o crop.
- Na parte de reconhecimento foram treinados dois svm's. Uma para letra e outra para números, para evitar o caso em que letras como O sejam confundidas com o número 0 por exemplo.



# Resultados

- Tabela de acertividade de reconhecimento dos caracter

Carácter	taxa	Carácter	taxa	Carácter	taxa	Carácter	taxa
0	67.5%	A	60.2%	K	70.0%	U	33.3%
1	90.9%	B	46.0%	L	66.6%	V	30.2%
2	45.0%	C	78.0%	M	34.5%	W	50.0%
3	50.2%	D	10.0%	N	52.3%	X	91.5%
4	60.5%	E	39.0%	O	30.2%	Y	92.0%
5	70.7%	F	53.0%	P	43.4%	Z	87.3%
6	45.0%	G	47.0%	Q	52.0%	.	.
7	47.0%	H	45.0%	R	45.3%	.	.
8	87.0%	I	50.0%	S	70.1%	.	.
9	68.0%	J	72.0%	T	40.2%	.	.

# Resultados

- Tabela de acurácia da segmentação, mostrando nossos resultados e o proposto pelo artigo estudado.

Referências	Acurácia de segmentação
Proposto em [3]	98.82%
Nossa proposta	90.9%

# Conclusão

- Método eficaz quando se consegue binarizar bem.
- A etapa de reconhecimento não obteve um resultado satisfatório devido ao tamanho da base de treinamento reduzida.

# Referências

- He, Xiangjian, et al. "Segmentation of characters on car license plates." Multimedia Signal Processing, 2008 IEEE 10th Workshop on. IEEE, 2008.
- C++, OpenCV. SVM em OpenCV. Disponível em: [http://docs.opencv.org/2.4/doc/tutorials/ml/introduction\\_to\\_svm/introduction\\_to\\_svm.html](http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html). Acesso em: 30 nov. 2016.