

# Automatic Detection and Recognition of Text-Based Traffic Signs from Images

G. Oliveira, F. Silva, D. Pereira, L. Almeida, A. Artero, A. Bonora, and V. de Albuquerque

**Abstract**—Detection and recognition of texts in traffic signs has been widely studied and with the advance in image capture technology has helped to improve or to create new methods to achieve this issue. In this work, we presented a method for detection, segmentation and recognition of text-based traffic signs from images analyzing and processing techniques. The results show that the computational cost and accuracy rate considering the proposed approach are acceptable to real time applications, with an execution time under 0.5 seconds, with a hit rate of 94.38% in the plate detection, 83.42% in the character segmentation and 89.23 in the digit classification.

**Index Terms**—Traffic signs detection, Traffic signs recognition, Characters segmentation, OCR.

## I. INTRODUÇÃO

EXISTEM muitos desafios a serem considerados ao realizar a detecção e o reconhecimento de textos em placas indicativas de trânsito, tais como: variação de iluminação, desgaste da placa, sujeira, dígitos com grafia (fontes) diferentes, localização da placa na cena, ângulo da câmera, distância da placa em relação à câmera, poluição visual no cenário que contém a placa, qualidade da resolução, entre outros [1] [2]. Para resolver esses problemas, diversas técnicas foram criadas para detecção de placas indicativas de trânsito, segmentação e classificação dos caracteres como, por exemplo, detecção de bordas em múltiplas escalas para obter candidatos a texto [3], utilização de intervalo de cores para encontrar regiões textuais [3] [4], algoritmo MSER (*Maximally Stable Extremal Regions*) [5] e HSV (*Hue, Saturation, Value*) aplicados para detectar placas de trânsito [1] [6], utilização de filtros de formas geométricas e tamanhos do dígito para buscar candidatos a caracteres [6].

Os sistemas que visam localizar regiões textuais em imagens e reconhecer os caracteres, geralmente, realizam as etapas mostradas na Fig. 1 [6] [7].

G. H. Oliveira, Universidade do Oeste Paulista (Unoeste), Presidente Prudente, São Paulo, Brasil (gtvoliveira@hotmail.com).

F. A. Silva, Universidade do Oeste Paulista (Unoeste), Presidente Prudente, São Paulo, Brasil (chico@unoeste.br).

D. R. Pereira, Universidade do Oeste Paulista (Unoeste), Presidente Prudente, São Paulo, Brasil (danilopereira@unoeste.br).

L. L. Almeida, Universidade do Oeste Paulista (Unoeste), Presidente Prudente, São Paulo, Brasil (lalmeida@unoeste.br).

A. O. Artero, Universidade Estadual Paulista (Unesp), Presidente Prudente, São Paulo, Brasil (almir@fct.unesp.br).

A. F. Bonora, Universidade do Oeste Paulista (Unoeste), Presidente Prudente, São Paulo, Brasil (alexbonora@gmail.com).

V. H. C. de Albuquerque, Universidade de Fortaleza (Unifor), Fortaleza, Ceará, Brasil (victor.albuquerque@unifor.br).

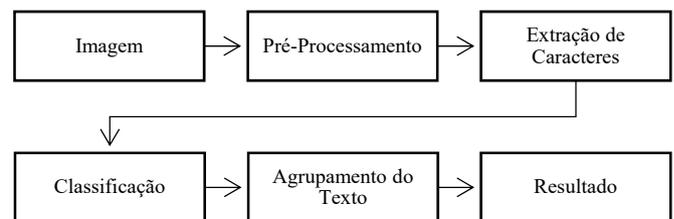


Fig. 1. Etapas de detecção de textos a partir de imagem.

A partir da captura da imagem é realizado um pré-processamento com a utilização de filtros, como o filtro gaussiano [8] e de intervalo de cores [3] [5], assim, é obtida a região da placa. Na sequência, são realizadas a segmentação e extração dos caracteres, sendo então feita a classificação e posteriormente o reconhecimento dos dígitos, que são agrupados em textos, formando o resultado do processo.

No que se refere ao reconhecimento ótico de caracteres, diversas técnicas já foram experimentadas na literatura [10], como, por exemplo, redes neurais artificiais [11], KNN (*K-Nearest Neighbor*) [12], OPF (*Optimum-Path Forest*) [13], transição entre pixels [14] [15], entre outras.

A detecção de placas indicativas em imagens capturadas no trânsito deve ser capaz de ter um bom comportamento em ambientes não controlados, em que as placas não possuem posições e lugares pré-definidos. Para a detecção de placas é comum que a busca seja realizada apenas em regiões de interesse na imagem, como mostrado na Fig. 2 [1] [4]. Neste trabalho, as buscas pelas placas ocorrem em toda a imagem, pois não há uma posição fixa da câmera ao capturar a imagem.

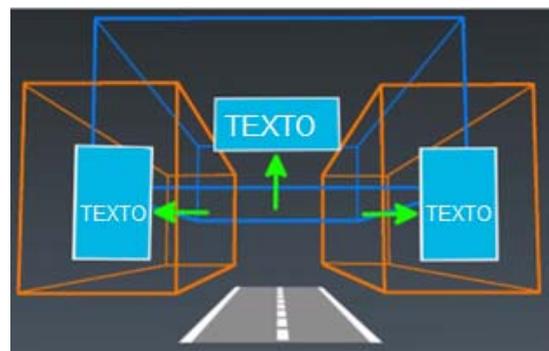


Fig. 2. Regiões de interesse para busca por placas indicativas. Adaptado de [1].

O primeiro passo no processo de detecção de placas indicativas de trânsito e o reconhecimento dos textos é a identificação da placa na cena. Neste trabalho, foi proposta

uma metodologia para trabalhar com placas indicativas de sentido, indicativas de distância e indicativas de serviços, que de acordo com Detran [19] possuem coloração verde ou azul de fundo.

Após essa seção introdutória, o trabalho está organizado da seguinte maneira. Na Seção II é descrito o método proposto para realizar a detecção e reconhecimento de placas com sinais de trânsito baseados em textos. A Seção III apresenta os experimentos realizados e resultados obtidos a partir da metodologia desenvolvida. Por fim, na Seção IV encontram-se as conclusões e propostas de trabalhos futuros.

## II. MÉTODO PROPOSTO

Nesta seção são detalhadas as etapas do método proposto, sendo a detecção da placa utilizando intervalo de cores e algoritmo de detecção de contornos [9] [16], segmentação da placa, correção de perspectiva utilizando matriz de transformação [17], segmentação dos caracteres por meio de limiarização [18] e da detecção de contornos [9]. E, por fim, para o reconhecimento dos caracteres das placas, foi utilizada a técnica proposta por Silva *et al.* [14], que analisa as transições entre os pixels da imagem do caractere segmentado, e faz a comparação dessas transições com todas as classes de dígitos, buscando a menor diferença. O fluxograma mostrado na Fig. 3 apresenta a sequência das etapas do método proposto.

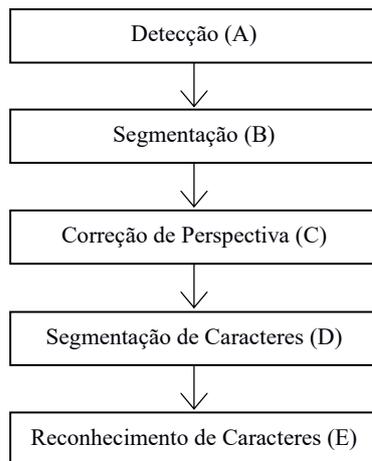


Fig. 3. Sequência das etapas do método proposto.

Os algoritmos utilizados neste trabalho para todas as etapas do método proposto foram implementados utilizando a linguagem C++ com auxílio da biblioteca de processamento de imagens e visão computacional OpenCV.

### A. Detecção

As placas escolhidas neste trabalho possuem cores de fundo pré-definidas, sendo azul ou verde. Para realizar a detecção, foram utilizadas duas imagens, a imagem original colorida RGB (*Red, Green, Blue*) e uma segunda imagem resultante da conversão desta para o espaço de cor HSV (*Hue, Saturation, Value*). Os canais RGB da imagem original e os canais da imagem HSV são separados, obtendo seis canais. É feita uma varredura em toda imagem, e, em cada pixel é verificado os

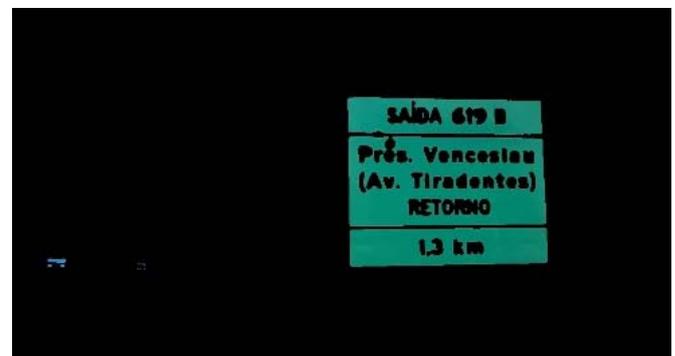
valores de  $R$  (do espaço de cores RGB) que representa o vermelho e  $S$  (do espaço de cores HSV) que é a saturação, se atendem a seguinte condição, determinada empiricamente:

$$R \leq 50 \text{ and } S \geq 180 \quad (1)$$

Placas indicativas de trânsito sempre têm cores intensas para chamar atenção dos condutores de automóveis no trânsito, apresentando uma alta saturação (canal  $S$  do espaço de cores HSV). Todos os pixels que atendem a condição de (1) permanecem na imagem convertida para o espaço de cor HSV, aos demais pixels dessa imagem são atribuídos o valor 0 (a região descartada, que não pertence à placa, fica com a cor preta). Na Fig. 4 é mostrado um exemplo de resultado da detecção da placa após a aplicação de (1).



a)



b)

Fig. 4. Exemplo de resultado após realizar a detecção da placa aplicando (1). a) Imagem original no espaço de cor RGB. b) Placa detectada no espaço de cor HSV.

Como a busca é realizada em toda a imagem, elementos com cores parecidas das placas (vegetação, céu) podem ainda permanecer no resultado, o que ainda precisam ser eliminados. Foi utilizado o algoritmo de Suzuki *et al.* [9] para detecção de contornos com a finalidade de extrair a região da placa, bem como eliminar os elementos de cores parecidas. A partir de um tamanho mínimo de contorno de 500 pixels, determinado empiricamente, são eliminados todos os contornos que possuem um valor inferior. As placas que aparecem nas imagens com tamanhos pequenos dificultam e provocam muitos erros na detecção e reconhecimento dos caracteres.

**B. Segmentação**

Com os contornos das placas definidos, faz-se a segmentação das mesmas. São utilizados os menores e maiores valores de  $u$  e  $v$  do contorno da placa, ou seja,  $u_{min}$ ,  $v_{min}$ ,  $u_{max}$  e  $v_{max}$  como mostrado na Fig. 5. Dessa forma, obtêm-se as regiões de interesse das placas.



Fig. 5. Placa Segmentada.

**C. Correção de Perspectiva**

Cada placa imageada, normalmente, está representada na imagem com um ponto de vista que pode ter deformações por causa do ângulo da câmera, no momento da captura. Essas deformações dificultam a detecção, segmentação e reconhecimento dos caracteres na placa.

Para corrigir essas deformações foi utilizada a correção de perspectiva da placa utilizando uma matriz de transformação [17]. A matriz possui os valores dos quatro pontos de coordenada  $(x_0, y_0)$ ,  $(x_1, y_1)$ ,  $(x_2, y_2)$  e  $(x_3, y_3)$  do contorno da placa na imagem, e também os quatro pontos da região de interesse  $(u_0, v_0)$ ,  $(u_1, v_1)$ ,  $(u_2, v_2)$  e  $(u_3, v_3)$ .

As coordenadas  $(u, v)$  são obtidas a partir dos valores  $(u_{min}, v_{min}, u_{max}, v_{max})$  da região de interesse que contém a placa segmentada, e as coordenadas  $(x, y)$  foram determinadas a partir dessa região nas linhas que compõem os contornos da placa na imagem. Na Fig. 6 são mostradas as coordenadas  $(u, v)$  da região de interesse da placa e as coordenadas  $(x, y)$  que representa o contorno da placa na cor vermelha.



Fig. 6. Coordenadas  $(x, y)$  e  $(u, v)$  da placa.

Para realizar a distorção da imagem e transformar as coordenadas  $(x, y)$  ( $A$ ) para as coordenadas  $(u, v)$  ( $B$ ) é utilizada (3):

$$A \cdot X = B \tag{3}$$

em que  $X$  representa os coeficientes desconhecidos, que são necessários para completar a matriz de transformação [17]:

$$\begin{bmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0u_0 & -y_0v_0 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -y_1v_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -y_2v_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3u_3 & -y_3v_3 \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -x_0v_0 & -y_0v_0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1v_1 & -y_1v_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2v_2 & -y_2v_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3v_3 & -y_3v_3 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{4}$$

Como é necessário encontrar os valores de  $X$ , é formado um sistema linear, o qual é resolvido pela decomposição  $LU$  (*Lower* e *Upper*) [20]. A decomposição  $LU$  separa a matriz em duas:

$$A = L \cdot U \tag{5}$$

onde  $L$  é a matriz triangular inferior e  $U$  é a matriz triangular superior. Para obter as matrizes  $L$  e  $U$  é utilizado o método de eliminação de Gauss [21]. A eliminação de Gauss tem como objetivo zerar todos os elementos abaixo da diagonal principal da matriz, assim obtêm-se a matriz  $U$ . Após esse processo, a matriz  $U$  é formada pela matriz com os elementos abaixo da diagonal principal zerados, e a matriz  $L$  é formada com o valor dos multiplicadores utilizados para zerar os respectivos elementos da matriz  $U$ , e com os valores acima da diagonal principal zerados.

Após obter ambas as matrizes *Lower* e *Upper* (3), pode-se substituir a matriz  $A$  pelas matrizes  $L$  e  $U$  formando assim  $(L \cdot U) \cdot X = B$  ou  $L \cdot (U \cdot X) = B$ . Com isso é possível obter:

$$L \cdot Y = B \tag{6}$$

$$U \cdot X = Y \tag{7}$$

Após resolver (6) e (7) são encontrados os valores dos coeficientes desconhecidos. Com todos os coeficientes já encontrados, são utilizadas as seguintes equações para descobrir os valores de  $U, V$ :

$$u_i = \frac{(a_0 \cdot x_i + a_1 \cdot y_i + a_2)}{c_0 \cdot x_i + c_1 \cdot y_i + 1} \tag{8}$$

$$v_i = \frac{(b_0 \cdot x_i + b_1 \cdot y_i + b_2)}{c_0 \cdot x_i + c_1 \cdot y_i + 1} \tag{9}$$

Então, para cada coordenada de pixel é aplicada (8) e (9) obtendo-se a nova coordenada. Após calcular as novas coordenadas de todos os pixels, a imagem da placa tem a sua perspectiva corrigida, como mostrado na Fig. 7.



Fig. 7. Correção de perspectiva. a) Placa deformada. b) Placa com a perspectiva corrigida.

D. Segmentação de Caracteres

Para realizar a segmentação dos caracteres de uma placa com a perspectiva corrigida, faz-se necessário realizar a binarização, ou seja, obter uma imagem apenas com pixels pretos e brancos. Foi utilizado o algoritmo de *threshold* (limiarização) de Otsu [18] para obter o limiar mais adequado. Após esse processo foi aplicada a detecção de contornos [9] para obter as regiões de todos os caracteres de texto contidos na placa.

A partir dos contornos obtidos dos caracteres foi realizada a segmentação de cada um deles, utilizando a mesma técnica da etapa de segmentação da placa, com os valores mínimos e máximos de altura e largura de cada caractere:  $x_{min}$ ,  $y_{min}$ ,  $x_{max}$  e  $y_{max}$ . Um exemplo de resultado dessa etapa de segmentação de caracteres pode ser visualizado na Fig. 8.



Fig. 8. Caracteres segmentados.

Algumas falhas na segmentação de caracteres da placa podem ocorrer como, por exemplo, dois ou mais caracteres ficarem unidos. Pode-se observar na Fig. 8 que o caractere “L” da palavra “AUTOMOVEL” está unido ao caractere “/” (barra), formando a junção de dois caracteres “L/”. Para corrigir essa falha, foi utilizado o tamanho (largura e altura) predominante da região dos outros caracteres vizinhos.

E. Reconhecimento de Caracteres

O método utilizado, neste trabalho, para o reconhecimento de caracteres foi implementado por Silva *et al.* [14]. A estratégia do algoritmo sugere descrever os caracteres, enquadrando-os em uma malha com dimensões definidas

previamente (neste trabalho, foi utilizada uma malha de dimensão 40x32) e, em seguida, observar as transições entre os valores 0 e 1 dos pixels adjacentes. Os atributos são definidos percorrendo os pixels na sequência indicada na Fig. 9.



Fig. 9. Caminho de varredura dos pixels pelo algoritmo de reconhecimento [14].

É construída uma lista com as transições permitidas para cada classe de caractere utilizando uma imagem (Fig. 10) contendo todos os caracteres alfanuméricos e alguns caracteres especiais ('\$', '(' e ')'), anotando as transições que não ocorrem em cada classe para formar a base do classificador. Cada caractere da Fig. 10 é segmentado e escalado para se enquadrar na malha com as mesmas dimensões utilizadas no processo de reconhecimento (40x32). Em seguida, são utilizadas as transições entre os valores 0 e 1 dos pixels adjacentes anotadas para formar o classificador.



Fig. 10. Imagem com caracteres alfanuméricos e os caracteres especiais utilizados.

Como as imagens dos caracteres usadas são binárias, as transições possíveis entre dois pixels que formam um caractere são: 00, 01, 10 e 11. Na Fig. 11 a) são representadas as transições de pixels de duas classes de letra ‘A’ e duas classes de letra ‘C’. Na Fig. 11 b) tem-se um gráfico de coordenadas paralelas destacando as transições das diferentes classes em a), e, finalmente, na Fig. 11 c), as não-ocorrências de transições nas classes.

Com as não-ocorrências anotadas no classificador, o reconhecimento de um caractere é realizado criando uma lista das transições entre pixels desse caractere. Essa sequência de transições entre pixels é comparada com a lista de não-ocorrências das classes do classificador. Três regras são estipuladas para determinar o reconhecimento do caractere: (i) reconhecer o caractere da classe que não for violada pelo caractere a ser reconhecido, (ii) caso mais de uma classe se encaixar na primeira regra, a classe que for mais restritiva entre elas é escolhida, e (iii) se as transições não se encaixarem nas restrições de nenhuma classe, a classe menos violada é escolhida.

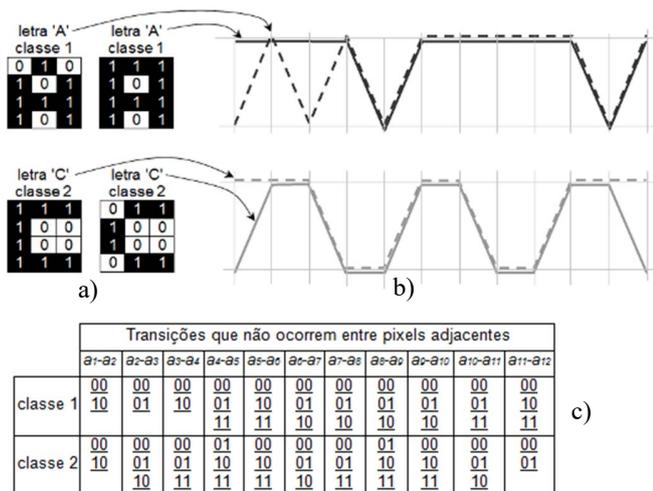


Fig. 11. Transições de pixels [14].

III. EXPERIMENTOS

Foram realizados dois experimentos neste trabalho, executados em dois computadores com hardware diferentes, sendo o primeiro utilizando um computador com processador i3 3240 3,4GHz com 4 GB de memória RAM e o segundo em um computador com processador i7 7740X 4,3GHz com 16 GB de memória RAM.

O banco de imagens foi formado capturando imagens de placas indicativas de trânsito ao longo da rodovia Raposo Tavares no trajeto da cidade de Presidente Prudente até a cidade de Presidente Epitácio, ambas as cidades no interior do estado de São Paulo, Brasil. O processo de captura das imagens foi realizado durante o dia com boa condição de iluminação devido à claridade do sol. Foi utilizada uma câmera Canon T3i EOS configurada para uma resolução de 1296x864 pixels, ISO 100, focal 55 mm, modo de exposição normal, balanço de cor e balanço de branco automático, e formato de imagem JPEG. Foram imageadas um total de 237 imagens, sendo que 41 imagens foram removidas por motivo de brilho excessivo provocado pela luz do sol, e 196 utilizadas nos dois experimentos. A Fig. 12 mostra um exemplo de imagem capturada na rodovia.



Fig. 12. Imagens de placas indicativas de trânsito presentes na rodovia Raposo Tavares entre as cidades de Presidente Prudente e Presidente Epitácio.

A. Resultados

O método proposto neste trabalho obteve resultados satisfatórios de detecção e segmentação. Na Tabela I são apresentados os resultados de acertos e tempos médios de execução de três partes dos algoritmos implementados do método proposto, obtidos a partir dos dois experimentos realizados. Os tempos de execução foram obtidos marcando-se o início e término de execução de cada uma das três partes dos algoritmos. A porcentagem de acertos da detecção, segmentação e correção da perspectiva foi obtida a partir do número de 185 imagens que tiveram sucesso na execução desses três algoritmos em relação às 196 imagens utilizadas. Dessas 185 imagens foram segmentados os dígitos, resultando em 83,42% de acerto. E a partir desses caracteres segmentados foi realizado o reconhecimento apresentando 89,23% de acerto utilizando o método de Silva *et al.* [14].

TABELA I  
RESULTADOS PARA CADA UM DOS EXPERIMENTOS E O TEMPO DE EXECUÇÃO DE CADA UM

	Primeiro Experimento		Segundo Experimento	
	Acertos [%]	Tempos [s]	Acertos [%]	Tempos [s]
Detecção, segmentação da placa e correção de perspectiva	94,38	1,80	94,38	0,36
Segmentação dos dígitos	83,42	0,72	83,42	0,01
Classificação dos dígitos	89,23	1,96	89,23	0,01

O tempo total médio de execução de uma imagem (somatório obtido pela Tabela I) de todas as etapas do primeiro experimento foi de 4,48 segundos para o computador com processador i3 de 3,4GHz, enquanto que no segundo experimento, utilizando um computador com processador i7 de 4,3GHz, o tempo de processamento foi de 0,38 segundos.

B. Falhas

Na Fig. 13 são apresentadas algumas falhas obtidas com a metodologia desenvolvida. Na Fig. 13 a), tem-se uma falha na etapa de segmentação de caracteres, em que a mancha (região demarcada) na imagem foi segmentada como um caractere. Quando aparece uma mancha na imagem da placa, após o processo de detecção, a mancha fica com a mesma cor dos caracteres impressos, e se tiver um tamanho semelhante ao tamanho dos caracteres, também é segmentado como um caractere. A Fig. 4 b) mostra a imagem completa de onde ocorreu a falha. Na Fig. 13 b), a detecção da placa é prejudicada pela presença de vegetação e variação da iluminação, isto porque o lado direito da placa está mais escuro. O escurecimento de parte da placa provocado por sombra de alguma vegetação, como foi esse caso, faz com que a condição determinada empiricamente em (1) não apresente bons resultados em toda a imagem da placa. Na Fig. 13 c), os caracteres unidos provocaram uma falha de reconhecimento de caractere. Essa falha foi corrigida utilizando o tamanho da região dos outros caracteres vizinhos (Seção II D). Na Fig.13 d), o resultado do reconhecimento dos caracteres,

mostrado na região inferior da imagem, apresenta um texto diferente do escrito na placa, em que a letra “I” foi confundida com o número “1” na fase de reconhecimento do caractere. Essa falha não foi corrigida neste trabalho, mas uma consulta a um dicionário de palavras poderia resolver esse problema.



Fig. 13. Exemplos de falhas ocorridas.

#### IV. CONCLUSÕES

Os resultados apresentados com a utilização da metodologia desenvolvida neste trabalho mostram que esta pode ser aplicada em situações que necessitam de um tempo de resposta rápido, isto porque os tempos de processamento foram abaixo de 0,5 segundos utilizando um computador com processador i7 de 4,3GHz, com taxa de acerto de 94,38% na etapa de detecção de placas, 83,42% na etapa de segmentação dos caracteres e 89,23% na classificação de dígitos. Esses resultados comprovam que a metodologia é promissora para aplicações que necessitam ser executadas em tempo real. Como neste trabalho é realizada apenas a detecção e reconhecimento dos textos escritos nas placas indicativas de sentido, de distância e de serviços, em trabalhos futuros seria interessante aumentar a abrangência dos tipos de placas de sinalização de trânsito. Seria de grande valia incluir a detecção e o reconhecimento de placas contendo símbolos. Novos experimentos poderiam ser realizados a partir de *datasets* de imagens de placas com resoluções variadas, o que necessita do desenvolvimento de novos algoritmos para tratar essa questão. Aplicações poderiam ser desenvolvidas com necessidade de execução em tempo real e fazer uso da metodologia desenvolvida. Um bom caminho para se conseguir acelerar o processamento em tempo real seria utilizar a programação em GPU (*Graphics Processing Unit*). Outro trabalho de grande valia seria desenvolver um aplicativo para smartphone para testar o método proposto, já que esses aparelhos possuem excelentes câmeras, e processadores e GPU's com capazes até de rodar jogos com gráficos complexos.

#### REFERÊNCIAS

[1] J. Greenhalgh, M. Mirmehdi, “Recognizing text-based traffic signs”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1360-1369, 2015.  
 [2] L. Kodwani, S. Meher, “Automatic license plate recognition in real time videos using visual surveillance techniques”, *ITSI Transactions on*

*Electrical and Electronics Engineering (ITSI-TEEE)*, vol. 1, no. 6, pp. 60-66, 2013.  
 [3] J. Gao, J. Yang, “An adaptive algorithm for text detection from natural scenes”, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, Kauai, HI, USA, vol. 2, pp. 84-89, 2001.  
 [4] A. González, L.M. Bergasa, J. Yebes, “Text detection and recognition on traffic panels from street-level imagery using visual appearance”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 228-238, 2014.  
 [5] J. Matas, O. Chum, M. Urban, T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions”, *Image and Vision Computing*, vol. 22, no. 10, pp. 761-767, 2004.  
 [6] H. Chen, S.S. Tsai, G. Schroth, D.M. Chen, R. Grzeszczuk, B. Girod, “Robust text detection in natural images with edge-enhanced maximally stable extremal regions”, in *18th IEEE International Conference on Image Processing*, Brussels, Belgium, pp. 2609-2612, 2011.  
 [7] B.L.S. Silva, P.M. Ciarelli, “Edge detection and confidence map applied to identify textual elements in images”, in *Proceedings of XII Workshop De Visão Computacional (WVC 2016)*, Campo Grande, MS, 2016.  
 [8] M. Basu, “Gaussian-based edge-detection methods-a survey”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 3, pp. 252-260, 2002.  
 [9] S. Suzuki, K. Abe, “Topological structural analysis of digitized binary images by border following”, *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32-46, 1985.  
 [10] R. Smith, “An overview of the tesseract OCR engine”, in *Proceedings of Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2007.  
 [11] A.R. Alexandria, P.C. Cortez, J.H.S. Felix, A.M. Girão, J.B.B. Frota, J.A. Bessa, “An OCR system for numerals applied to energy meters”, *IEEE Latin America Transactions*, vol. 12, no. 6, pp. 957-964, September 2014.  
 [12] T.K. Hazra, D.P. Singh, N. Daga, “Optical character recognition using KNN on custom image dataset”, in *Proceedings of 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON)*, pp. 110-114, 2017.  
 [13] G.S. Lopes, D.C.S. Vieira, A.W.O. Rodrigues, P.P. Rebouças-Filho, “Recognition of handwritten digits using the signature features and optimum-path forest”, *IEEE Latin America Transactions*, vol. 14, no. 5, pp. 2455-2460, September 2016.  
 [14] F.A. Silva, A.O. Artero, M.S.V. Paiva, R.L. Barbosa, “Um algoritmo rápido para o reconhecimento de caracteres”, in *Proceedings of VII Workshop de Visão Computacional (WVC 2011)*, UFPR, Curitiba, pp. 149-154, 2011.  
 [15] G.L. Corneto, F.A. Silva, D.R. Pereira, L.L. Almeida, A.O. Artero, J.P. Papa, V.H.C. Albuquerque, H.M. Sapia, “A New Method for Automatic Vehicle License Plate Detection”, *IEEE Latin America Transactions*, vol. 15, no. 1, pp. 75-80, jan. 2017.  
 [16] S. Xavier, R. Reshmi, “Automatic detection and recognition of tet in traffic sign boards based on word recognizer”, *International Journal for Innovative Research in Science & Technology - IJIRST*, vol. 3, no. 4, pp. 223-231, September 2016.  
 [17] B.F. Reis, J.M.X.N. Teixeira, V. Teichrieb J. Kelner, “Perspective correction implementation for embedded (marker-based) augmented reality”. In *Proceedings of Workshop de Realidade Virtual e Aumentada*, UFPE, Recife, PE, 2008.  
 [18] N. Otsu, “A threshold selection method from gray-level histograms”, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no.1, pp. 62-66, 1979.  
 [19] Detran. Departamento Estadual de Trânsito. (2017) [ONLINE]. Available: <http://portal.detran.ce.gov.br/index.php/educacao/742-placas-de-indicacao>.  
 [20] J.R. Bunch, J.E. Hopcroft, “Triangular factorization and inversion by fast matrix multiplication”, *Mathematic of Computation*, no. 28, pp. 231-236, 1974.  
 [21] A.K. Kaw, E.E. Kalu, D. Nguyen, “Numerical methods with applications: chapter 04.06 Gaussian Elimination”. University of South Florida. (2018). [ONLINE] Available: [http://mathforcollege.com/nm/mws/gen/04sle/mws\\_gen\\_sle\\_txt\\_gaussian.pdf](http://mathforcollege.com/nm/mws/gen/04sle/mws_gen_sle_txt_gaussian.pdf).



**Gustavo Henrique de Oliveira** graduado em Ciência da Computação na Universidade do Oeste Paulista (Unoeste), Brasil (2018).



**Alex Fernando Bonora** graduado em Ciência da Computação na Universidade do Oeste Paulista (Unoeste), Brasil (1997), especialista em Banco de Dados no Centro Universitário Eurípedes de Marília (Univem), Brasil (2004) e Mestrando em Informática na Universidade Tecnológica Federal do Paraná (UTFPR), Brasil (2018). Atualmente é professor na Universidade do Oeste Paulista (Unoeste), Brasil.



**Francisco Assis da Silva** graduado em Ciência da Computação na Universidade do Oeste Paulista (Unoeste), Brasil (1998), Mestre em Ciência da Computação na Universidade Federal do Rio Grande do Sul (UFRGS), Brasil (2002), Doutor em Ciências, Programa de Engenharia Elétrica da Universidade de São Paulo (USP), Brasil (2012) e Pós-Doutor na Universidade Estadual Paulista (Unesp), Brasil (2017). Atualmente é professor na Universidade do Oeste Paulista (Unoeste), Brasil.



**Victor Hugo C. de Albuquerque** graduado em Tecnologia Mecatrônica no Centro Tecnológico Federal de Educação do Ceará, Brasil (2006), Mestre em Engenharia Teleinformática na Universidade Federal do Ceará, Brasil (2007) e Doutor em Engenharia Mecânica com ênfase em Materiais na Universidade Federal da Paraíba, Brasil (2010). Atualmente é professor do Programa de Graduação em Informática da Universidade de Fortaleza (Unifor), Brasil.



**Danilo Roberto Pereira** graduado em Ciência da Computação na Universidade Estadual de São Paulo (Unesp), SP, Brasil em 2006, Mestre em Ciência da Computação na Universidade de Campinas (Unicamp), Brasil (2009), Doutor em Ciência da Computação (Unicamp), Brasil (2013), Pós-Doutor na Universidade Estadual de São Paulo, Brasil (2016) e Pós-Doutor na Universidade Federal de São Carlos (UFSCar), Brasil (2017). Atualmente é professor na Universidade do Oeste Paulista (Unoeste), Brasil.



**Leandro Luiz de Almeida** graduado em Ciência da Computação na Universidade do Oeste Paulista (Unoeste), Brasil (1997), Mestre em Ciências Cartográficas na Universidade Estadual de São Paulo (Unesp), Brasil (2001) e Doutor em Ciências, Programa de Engenharia Elétrica da Universidade de São Paulo (USP), Brasil (2013). Atualmente é professor na Universidade do Oeste Paulista (Unoeste), Brasil.



**Almir Olivette Artero** graduado em Matemática na Universidade Estadual de São Paulo (Unesp), Brasil (1990), Mestre em Ciências Cartográficas na Universidade Estadual de São Paulo (Unesp), Brasil (1999) e Doutor em Ciência da Computação na Universidade de São Paulo (USP) (2005). Atualmente é professor na Universidade do Estado de São Paulo (Unesp), Brasil.