

Traffic Sign Detection and Recognition using the AdaBoost and transition between pixels

João Paulo Masieiro, Francisco Assis da Silva,
Danillo Roberto Pereira

Universidade do Oeste Paulista – Unoeste/FIPP
Presidente Prudente/SP, Brasil
joao.p.masiero@gmail.com, {chico, danilopereira}@unoeste.br

Almir Olivette Artero, Marco Antonio Piteri
Universidade Estadual Paulista – Unesp/FCT
Presidente Prudente/SP, Brasil
{almir, piteri}@fct.unesp.br

Abstract—This paper presents the algorithm TSRATP (Traffic Sign Recognition using AdaBoost and Transition between Pixels) proposed for traffic sign detection and recognition using images. Initially, the AdaBoost algorithm is used to detect the plates from a specific training for traffic signals, without considering all the information printed on the plates, but only the basic information, such as red contours, the presence or not of diagonal lines represent prohibition, yellow or red background. Different from several studies in the literature, this strategy reduces the amount of different plates needed for AdaBoost training. In addition, the HSV color space was used to eliminate regions incorrectly detected by AdaBoost. The method proposed in this paper to perform the recognition of traffic signs is based on transitions which occur between adjacent pixels which form the signals.

Keywords—*traffic sign detection; traffic sign recognition; AdaBoost; transition between pixels*

I. INTRODUÇÃO

Os sinais de trânsito são muito importantes porque estabelecem as regras de navegação pelas vias. Suas propriedades visuais são muito fortes e facilmente identificáveis pelos seres humanos [9]. O reconhecimento automático de placas de trânsito está se tornando uma área com muitas aplicações na indústria automotiva [3], por exemplo, possibilitando que um motorista seja avisado sobre ações impróprias e situações potencialmente perigosas, o que consiste em uma tarefa essencial para um sistema inteligente de auxílio a motoristas [4]. Os textos e símbolos inseridos em um sinal de trânsito, geralmente, apresentam informação útil, como limite de velocidade, direção da via, e situações de trânsito, que ajudam os motoristas a dirigir com maior segurança [5].

Este trabalho apresenta uma metodologia para realizar a detecção e o reconhecimento de placas de trânsito, em que foi utilizada a metodologia de Silva et al. [19] para realizar a detecção dos sinais de trânsito, na qual os autores fazem uso do algoritmo AdaBoost [16], que tem sido muito usado para o reconhecimento de faces, para selecionar áreas candidatas contendo imagens de placas de sinalização de trânsito. Para o reconhecimento do sinal de trânsito, propõe-se neste trabalho, uma estratégia simples e rápida para modelar o comportamento dos sinais de trânsito, usando apenas as transições que ocorrem entre os níveis de pixels adjacentes que formam os sinais. Essa estratégia foi inspirada no trabalho

de reconhecimento de caracteres de Silva et al. [20]. Os sinais de trânsito tratados nesta proposta são as placas de sinalização afixadas em calçadas de ruas e avenidas.

As demais seções deste trabalho estão organizadas da seguinte maneira: Na Seção II são apresentados alguns trabalhos relacionados ao reconhecimento de sinais de trânsito; Na Seção III é apresentada uma breve revisão do algoritmo usado na detecção; Na Seção IV é apresentado o algoritmo TSRATP, proposto neste trabalho, que realiza a detecção e o reconhecimento das placas; Na Seção V são apresentados os resultados obtidos com a aplicação da metodologia proposta em conjunto de imagens de alta resolução, além de apresentar uma comparação com trabalhos anteriores; Por fim, na Seção VI são apresentadas as conclusões e trabalhos futuros.

II. TRABALHOS RELACIONADOS

Segundo Chen e Hsieh [5], a detecção de sinais de trânsito apresenta muitas dificuldades, devido a diferentes alterações de condições ambientais (nebulosidade, iluminação, sombras, etc.), o que ocasiona sinais de trânsito com diferentes mudanças na sua aparência. Outros fatores, ocasionados pela captura das imagens ou vídeos devem ser considerados, isto porque os efeitos de perspectiva fazem o sinal de trânsito ter diferentes tamanhos, formas, contrastes e borrachentos. Em alguns casos, a placa com o sinal de trânsito pode ficar totalmente ou parcialmente encoberta por outros objetos como árvores, veículos, etc.

Ao longo dos últimos anos foram propostas diversas abordagens para resolver o problema do reconhecimento de sinais de trânsito, como por exemplo: utilizando propriedades de cor dos sinais [6]; a partir da geometria das placas em imagens em tons de cinza [7]; baseada nas propriedades de forma dos sinais [8]; baseada em propriedades de cor e forma [9]; utilizando classificadores baseados em redes neurais [3]; analisando a curvatura da representação do sinal com contornos ativos (Snakes) [10]; usando *Support Vector Machine* (SVM) [11]; usando uma segmentação de sinais de trânsito baseado em conjuntos fuzzy [12].

O trabalho de Silva et al. [2] apresenta como proposta para o reconhecimento de placas de sinalização de trânsito, o uso do algoritmo SIFT, para detectar pontos chave nas imagens e, em seguida, compara os descritores destes pontos com os descritores obtidos em *templates* contendo os sinais de

trânsito. Por fim, aplica um algoritmo de reconhecimento de caracteres para reconhecer textos, tais como os limites de velocidade. Este algoritmo apresenta bons resultados, entretanto, como precisa aplicar o algoritmo SIFT na imagem inteira, a sua aplicação em tempo real se torna impraticável.

O trabalho de Silva et al. [19] melhora o processo de detecção proposto por Silva et al. [2] utilizando o algoritmo AdaBoost na imagem inteira e, então, usa o SIFT apenas nas regiões candidatas, detectadas pelo AdaBoost. A vantagem desta proposta é que o AdaBoost possui um tempo de processamento muito pequeno, sendo indicado para aplicações em tempo real.

III. REFERENCIAL TEÓRICO

O detector de Viola e Jones [1] baseia-se em uma cascata de classificadores boosted Haar-like, combinando dois conceitos: (i) classificadores Haar-like e (ii) AdaBoost [16]. Os classificadores Haar-like são construídos com características retangulares simples, que representam as diferenças de somas de pixels específicos em uma imagem. Cada característica é emparelhada com um limiar, e a decisão do classificador é determinada por uma comparação do valor da característica com o limiar [13]. Esta simplicidade constitui um diferencial importante para minimizar o tempo de processamento. O algoritmo AdaBoost usa pesos em uma combinação linear de classificadores fracos, formando um classificador forte [5].

Viola e Jones seguem nesta linha, agrupando vários classificadores fortes, construídos por AdaBoost, em uma cascata que apresenta um processamento muito rápido. O classificador forte na primeira fase da cascata é escolhido de modo a descartar falsos positivos e, ao mesmo tempo, preservar quase todos os verdadeiros positivos do conjunto de treino. O processo de detecção de faces é realizado por uma janela deslizante de detecção através da imagem. Dentro da janela, a resposta da cascata é calculada. Depois de completar uma passagem sobre a imagem, o tamanho da janela de detecção é aumentado por algum fator (1.2 no OpenCV, significando que a escala da janela será aumentada em 20%). O tamanho da janela é aumentado até um tamanho pré-definido ser alcançado. Aumentando a janela de detecção por uma percentagem menor produz as melhores taxas de detecção, mas aumenta o número de falsos positivos e também o tempo total de processamento [13].

O sistema de detecção de faces proposto por Viola e Jones [1] utiliza características simples para classificar as imagens, baseadas nas funções de Haar. As características correspondem basicamente a retângulos que possuem duas regiões, sendo uma clara e outra escura. O cálculo do valor correspondente à característica é feito subtraindo o somatório da intensidade dos pixels da região escura do total acumulado pela intensidade dos pixels da região clara. Este cálculo pode ser feito rapidamente utilizando a representação de imagem integral. Na Fig. 1 são apresentadas as características básicas de Haar que foram utilizadas por Viola e Jones [1].



Fig. 1. Características Básicas de Haar.

Com o intuito de melhorar ainda mais a etapa de detecção de objetos, Lienhart e Maydt [17] estenderam a quantidade de tipos possíveis de características retangulares, acrescentando características às previamente sugeridas, mas desta vez rotacionadas em 45 graus. Com estas novas adições ao conjunto, consegue-se reduzir o número de falsos positivos em torno de 10%, aumentando ainda mais a eficiência do processo de detecção. Na Fig. 2 é apresentado o conjunto completo com as novas características retangulares sugeridas.

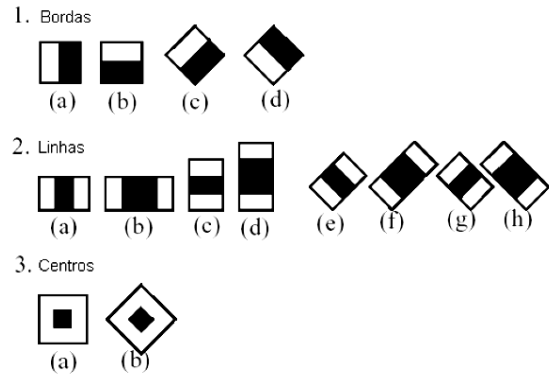


Fig. 2. Características Estendidas. Fonte: [17].

Devido ao fato do algoritmo AdaBoost apresentar alta velocidade de processamento e eficácia, ele é indicado neste trabalho para selecionar as áreas candidatas na imagem a terem uma placa de sinalização de trânsito.

IV. TSRATP - RECONHECIMENTO DE SINAIS DE TRÂNSITO USANDO ADABOOST E TRANSIÇÃO ENTRE PIXELS

O processo de reconhecimento de placas, geralmente, é dividido em duas etapas, que são a detecção e o reconhecimento. Neste trabalho, a detecção utiliza o algoritmo AdaBoost [1]. Apesar do AdaBoost ter sido usado originalmente na detecção de rostos, segundo Brkić [13], outras pesquisas têm aplicado com sucesso o detector em uma grande quantidade de outras classes de objetos. De fato, muitos autores já utilizam essa técnica para realizar o reconhecimento de placas de trânsito [3][4][5][14][15][19]. Neste trabalho foi adotado o treinamento de imagens de placas de sinalização realizada em [19], em que são usadas imagens de treinamento simplificadas, que omitem a maior parte das informações impressas nas placas, ou seja, com apenas as informações de bordas, como círculos vermelhos com ou sem as linhas diagonais que representam proibição, além de placas com fundo amarelo, ou placas com fundo vermelho. Segundo Silva et al. [19], o objetivo desta simplificação é abranger a maior quantidade possível de diferentes tipos de sinais de trânsito, sem a necessidade de se usar exemplares de cada um deles durante o treinamento. Essa limitação a uma pequena quantidade de placas diferentes é para se ter um tempo baixo de processamento.

No processo em [19], após a aplicação da etapa de detecção, que deveria resultar apenas regiões das imagens que possuem uma placa de sinalização de trânsito, ainda é possível existir algumas regiões que não possuem tais elementos. Para que essas regiões sejam rapidamente eliminadas do processo,

evitando o seu processamento desnecessário para realizar o reconhecimento, foi utilizado o espaço de cor HSV (*Hue Saturation Value*), para se verificar as cores predominantes nessas regiões e, assim, descartar regiões detectadas incorretamente.

Em seguida, as regiões candidatas passam pela etapa de reconhecimento, utilizando um algoritmo que analisa o comportamento sequencial dos pixels.

A. Modelagem do Comportamento das Sequências de Pixels

A estratégia proposta neste trabalho sugere classificar os sinais de trânsito, enquadrando-os em uma malha com dimensões definidas previamente e, em seguida, observar as transições entre os níveis de cinza (0 e 1 – imagens binárias) dos pixels adjacentes. Deste modo, uma imagem com dimensões $m \times n$ gera um vetor de transições (atributos) com $m \times n$ posições, sendo os pixels percorridos na sequência indicada na Fig. 3. Assim, determina-se o comportamento das sequências de pixels em cada classe e, então, este conhecimento pode ser usado para classificar as poligonais de outros registros, para os quais não se conhece a classe.

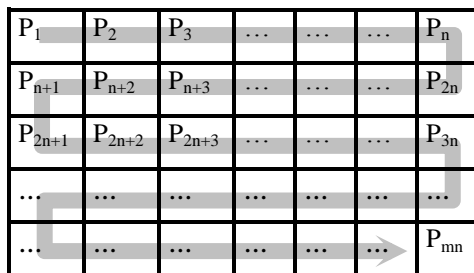


Fig. 3. Sequência de pixels na imagem usada.

Nesta estratégia, é construída uma lista com as transições permitidas para cada placa, anotando as transições que ocorrem entre os atributos adjacentes em cada uma das classes. No entanto para que se chegue a uma classe com as transições, faz-se necessário aplicar filtros para remover os detalhes que não compõem os sinais de trânsito da placa causados pelas condições de iluminação e sombras, bem como o fundo da cena a qual placa foi capturada. A Fig. 4 mostra algumas imagens de placas antes do tratamento.



Fig. 4. Imagens de placas para treinamento antes do tratamento.

Observando os tipos de placas, basicamente elas podem ser divididas em dois grupos básicos: 1 – redondas com fundo branco, 2 – losangos com fundo amarelo e a placa de pare com fundo vermelho.

A necessidade de dividir as placas em dois grupos foi pelo fato de terem formas e cores diferentes, assim, um filtro aplicado sobre uma placa do grupo 1 não apresentaria o mesmo efeito no grupo 2.

Todo o processo se inicia com a abertura da imagem em tons de cinza. Para o grupo 1 dois *thresholds* com limiares

diferentes foram aplicados. Inicialmente foi aplicado um *threshold* que define como 0 qualquer pixel fora da faixa determinada e mantém a intensidade dos outros pixels. A faixa de intensidade utilizada foi de 50 a 255, em que todos os pixels abaixo de 50 se tornaram 0 (Fig. 5).



Fig. 5. a) Imagem antes do *threshold*, b) após aplicação do *threshold*.

Posteriormente, foi aplicado um *threshold* de Otsu [18]. O resultado são imagens binárias, como as mostradas na Fig. 6.



Fig. 6. Imagens após a aplicação do *threshold* de Otsu.

Após a aplicação dos *thresholds*, em algumas imagens podem restar ruídos causados pelas falhas na impressão dos sinais na placa. Para eliminação destes ruídos foi aplicado uma erosão morfológica e uma dilatação morfológica respectivamente na imagem com um elemento estruturante 2x2 na forma de um quadrado com o ponto de origem na coordenada 0,0. O resultado são imagens em que pequenas falhas na impressão dos sinais nas placas são preenchidas. Ao final de todo o processo, a imagem é redimensionada para uma malha de tamanho de 64x64 pixels e as falhas restantes são eliminadas (Fig. 7).

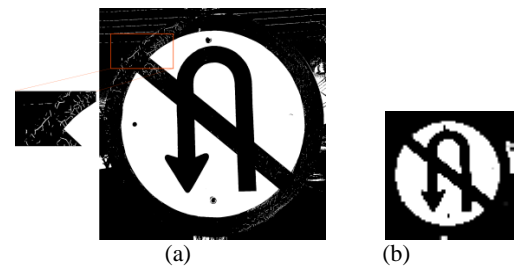


Fig. 7. Imagens a) antes e b) após as operações morfológicas e redimensionamento.

Para se ter uma padronização das placas, elas foram divididas em três grupos: A – placas redondas, B – placas em formato de losango e C – placa octogonal (placa de pare). Este agrupamento tem como finalidade eliminar todo o fundo da placa e ter uma máscara padronizada que é aplicada sobre a imagem inicial. A Fig 8. Mostra três máscaras que correspondem a cada grupo, e que são adicionadas nas imagens das placas para padronização por meio de sobreposição. A sobreposição da imagem do sinal de trânsito com a sua máscara correspondente resulta em uma imagem, a qual os pixels pretos da máscara irão sobrepor as regiões que correspondem ao fundo da cena a qual elas foram capturadas.



Fig. 8. Máscaras usadas na padronização das placas.

A área branca da máscara corresponde à área central da imagem onde se localiza o sinal de trânsito impresso na placa a ser reconhecido, assim o centro da imagem não é sobreposto. O resultado são imagens como mostradas na Fig. 9.



Fig. 9. Imagens finais com a aplicação das máscaras.

O processo de filtragem aplicado no grupo 1 é semelhante ao do grupo 2, porém, os filtros iniciais (*threshold*) são distintos para esses dois grupos. A Fig. 10 mostra alguns exemplos de imagens do grupo 2 (placa no formato de losango com fundo amarelo e a placa de pare com fundo vermelho).



Fig. 10. Imagens originais do grupo 2.

Assim como no primeiro caso elas são carregadas em tons de cinza. Um *threshold* com limiar 120 é aplicado nas imagens. Esse *threshold* já consegue eliminar sombras e detalhes desnecessários nas placas, restando apenas o sinal de trânsito impresso a ser reconhecido. A Fig. 11 mostra o resultado de algumas imagens de placas após a aplicação do *threshold*.



Fig. 11. Imagens após a aplicação do *threshold*.

Após aplicação do *threshold*, os demais passos para tratamento das imagens são os mesmos do grupo 1, em que são aplicados os filtros morfológicos e são adicionadas as máscaras por sobreposição. O resultado são imagens como a Fig. 12.



Fig. 12. Imagens do grupo 2 ao final do tratamento.

Como as imagens usadas são binárias, as transições possíveis entre dois pixels que formam uma figura são: 00, 01, 10 e 11. A Figura 13 apresenta um exemplo em que as poligonais da classe 1 possuem apenas a transição 10, não ocorrendo as transições 01, 11 e 00 entre os atributos a_1 e a_2 . Quanto à classe 2, nota-se que não ocorrem as transições 10 e 00. Assim, para cada classe são determinadas todas as transições que não ocorrem. Em seguida, a partir desta informação, os sinais de trânsito a serem reconhecidos são incluídos na classe que apresenta o menor número de inconsistências em relação às transições características anotadas em cada classe.

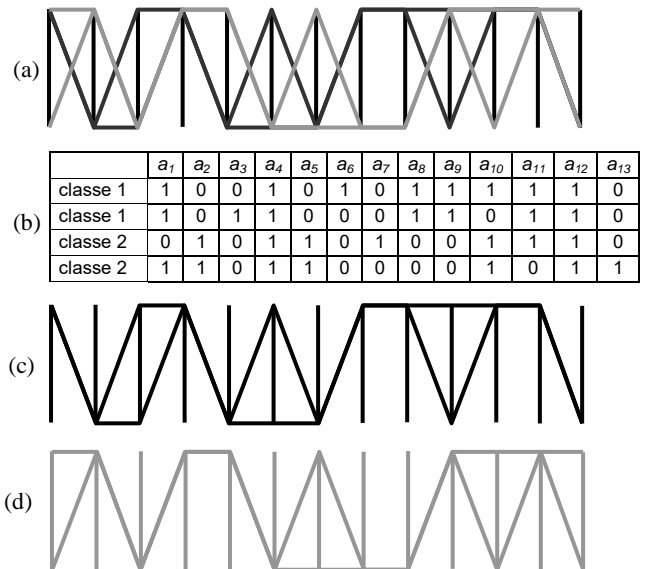


Fig. 13. a) Exibição dos registros da classe 1 em preto e da classe 2 em cinza; b) conjunto de dados contendo 4 registros com treze atributos em duas classes; c) Exibição dos registros da classe 1 e d) classe 2.

Em seguida, a classificação dos sinais de trânsito pode então ser conduzida usando as três regras de classificação propostas na Fig. 14. A última regra (R_3) prevê um limiar definido pelo usuário para determinar quando o registro deve ser classificado como ruído. Por exemplo, classificando-o como ruído quando ele não atende ao menos 30% das restrições de regra alguma.

- R_1 – Reconhecer o sinal de trânsito na classe cujas transições não são violadas pelas transições do registro;
- R_2 – Havendo mais de uma classe que satisfaz esta condição, a classe que possuir mais restrições (classe mais restritiva) deverá ser a escolhida;
- R_3 – Quando o registro não atende todas as restrições de nenhuma classe, deverá ser inserido na classe menos violada, ou ser classificado como ruído em casos extremos.

Fig. 14. Regras para reconhecimento dos sinais de trânsito.

As restrições das duas classes do conjunto de dados ilustrado na Fig. 13 (b) são apresentadas na Tabela I. No caso, as duas classes possuem quantidades iguais de restrições, ou

seja, não ocorrem as trinta transições na classe 1 e as trinta transições na classe 2, indicadas na Tabela I.

TABELA I. TRANSIÇÕES QUE NÃO OCORREM ENTRE ATRIBUTOS ADJACENTES NO CONJUNTO DE DADOS ILUSTRADO NA FIG. 13 (B).

	Atributos adjacentes											
	a_1-a_2	a_2-a_3	a_3-a_4	a_4-a_5	a_5-a_6	a_6-a_7	a_7-a_8	a_8-a_9	a_9-a_{10}	$a_{10}-a_{11}$	$a_{11}-a_{12}$	$a_{12}-a_{13}$
classe 1	$\frac{00}{01}$ $\frac{11}{11}$	$\frac{10}{11}$ $\frac{11}{11}$	$\frac{00}{10}$ $\frac{10}{11}$	$\frac{00}{01}$ $\frac{11}{11}$	$\frac{10}{11}$ $\frac{11}{11}$	$\frac{01}{11}$ $\frac{11}{11}$	$\frac{00}{10}$ $\frac{10}{11}$	$\frac{00}{01}$ $\frac{10}{10}$	$\frac{00}{01}$ $\frac{10}{10}$	$\frac{00}{10}$ $\frac{10}{10}$	$\frac{00}{01}$ $\frac{10}{10}$	$\frac{00}{01}$ $\frac{11}{11}$
classe 2	$\frac{00}{10}$ $\frac{11}{11}$	$\frac{00}{01}$ $\frac{11}{11}$	$\frac{00}{10}$ $\frac{11}{11}$	$\frac{00}{01}$ $\frac{10}{10}$	$\frac{00}{01}$ $\frac{11}{11}$	$\frac{10}{11}$ $\frac{11}{11}$	$\frac{01}{11}$ $\frac{10}{11}$	$\frac{01}{10}$ $\frac{11}{11}$	$\frac{00}{10}$ $\frac{10}{11}$	$\frac{00}{01}$ $\frac{10}{11}$	$\frac{00}{10}$ $\frac{10}{11}$	$\frac{00}{01}$ $\frac{10}{11}$

V. EXPERIMENTOS REALIZADOS

Esta seção apresenta o experimento realizado com a proposta apresentada neste trabalho. O experimento consistiu em comparar os resultados obtidos com a abordagem proposta por Silva et al. [2], com a abordagem por Silva et al. [19] e com os resultados obtidos com o uso do algoritmo TSRATP.

Toda a parte de detecção foi obtida em Silva et al. [19] Para a realização do treinamento AdaBoost, os autores utilizaram 5.300 imagens positivas, ou seja, imagens que contém placas de trânsito, distribuídas em quatro classes, conforme mostra a Tabela II.

TABELA II. CLASSES DE PLACAS USADAS NO TREINAMENTO.

	Positivas	Negativas	Estágios	Tamanho
Circular	2.892	6.000	25	24 x 24
Alerta	608	1200	20	24 x 24
Pare	900	2.000	20	24 x 24
Preferência	950	2.000	20	24 x 24

A Fig. 15(a) mostra alguns exemplos de imagens positivas, enquanto que em (b), tem-se as mesmas placas sem a informação referente ao sinal de trânsito, com apenas as bordas circulares, sem ou com as linhas diagonais de proibição, e uma placa apenas com seu fundo amarelo. Essas placas sem a informação do sinal, representam exemplos de imagens que foram usadas no treinamento AdaBoost, adotado na etapa de detecção, cujo objetivo é identificar as regiões da imagem que podem ter uma placa de trânsito.



Fig. 15. Exemplos de imagens positivas: a) originais; b) usadas no treinamento, apenas com as informações básicas das placas; c) usadas diretamente no treinamento.

As 60 imagens utilizadas possuem resolução de 2144 x 1424 pixels, obtidas a partir de uma câmara Nikon D300S instalada sobre um veículo (Sistema de Mapeamento Móvel). As imagens foram obtidas com o veículo em movimento em condições normais de tráfego.



Fig. 16. Placas de sinais de trânsito usadas como *template*.

As 73 placas utilizadas neste trabalho são placas dos mesmos tipos da Fig. 16, e para treinamento do algoritmo de reconhecimento foram utilizadas mais de uma imagem de sinal por placa. Conforme apresentado na Tabela III a abordagem sugerida por Silva et al. [2] foram detectadas 62 placas, obtendo-se uma taxa de acerto de 84.93%. Não ocorreram erros de identificação e 11 placas não foram reconhecidas (15.07%). Utilizando a abordagem sugerida por Silva et al. [19], foram reconhecidas corretamente 58 placas, obtendo-se uma taxa de acerto de 79.45%. Ocorreram apenas 2 erros de identificação (2.74% de erros) e 13 placas não foram reconhecidas (17.81%). Adotando a nova proposta apresentada neste trabalho, foram detectadas 66 placas obtendo uma taxa de acerto de 90.41%. Ocorreram 6 erros de identificação (8.22%) e 1 placa não foi reconhecida (1.37%).

Usando a abordagem sugerida por Silva et al. [2], o tempo médio de processamento destas 60 imagens em uma máquina com processador Intel Core i3 M330 2.13 GHz (RAM de 4 Gigabytes) é de 90 segundos. No trabalho sugerido por Silva et al. [19], o tempo médio de processamento destas 60 imagens na mesma máquina é de apenas 7.4 segundos. Adotando a nova proposta apresentada neste trabalho o tempo de processamento é de em média 7.0 segundos.

TABELA III. RESULTADOS DO RECONHECIMENTO DAS PLACAS.

Template	Metodologia de Silva et al. [2]			Metodologia de Silva et al. [19]			Metodologia proposta neste trabalho		
	Acertos	Erros	Não rec.	Acertos	Erros	Não rec.	Acertos	Erros	Não rec.
1	4 (100%)	0 (0%)	0 (0%)	3 (75%)	0 (0%)	1 (25%)	4 (100%)	0 (0%)	0 (0%)
2	3 (100%)	0 (0%)	0 (0%)	2 (67%)	0 (0%)	1 (33%)	3 (100%)	0 (0%)	0 (0%)
3	14 (73,68%)	0 (0%)	5 (26,32%)	16 (84,21%)	0 (0%)	3 (15,79%)	19 (100%)	0 (0%)	0 (0%)
4	0 (0%)	0 (0%)	1 (100%)	0 (0%)	0 (0%)	1 (100%)	0 (0%)	0 (0%)	1 (100%)
5	4 (100%)	0 (0%)	0 (0%)	3 (75%)	1 (25%)	0 (0%)	4 (100%)	0 (0%)	0 (0%)
6	5 (100%)	0 (0%)	0 (0%)	3 (60%)	1 (20%)	1 (20%)	4 (80%)	1 (20%)	0 (0%)
7	7 (77,78%)	0 (0%)	2 (22,22%)	5 (55,56%)	0 (0%)	4 (44,44%)	7 (77,78%)	2 (22,22%)	0 (0%)
8	8 (100%)	0 (0%)	0 (0%)	8 (100%)	0 (0%)	0 (0%)	8 (100%)	0 (0%)	0 (0%)
9	4 (66,67%)	0 (0%)	2 (33,33%)	6 (100%)	0 (0%)	0 (0%)	5 (83,33%)	1 (16,67%)	0 (0%)
10	1 (100%)	0 (0%)	0 (0%)	1 (100%)	0 (0%)	0 (0%)	1 (100%)	0 (0%)	0 (0%)
11	2 (100%)	0 (0%)	0 (0%)	2 (100%)	0 (0%)	0 (0%)	2 (100%)	0 (0%)	0 (0%)
12	6 (100%)	0 (0%)	0 (0%)	5 (83,33%)	0 (0%)	1 (16,67%)	6 (100%)	0 (0%)	0 (0%)
13	1 (100%)	0 (0%)	0 (0%)	1 (100%)	0 (0%)	0 (0%)	1 (100%)	0 (0%)	0 (0%)
14	3 (75%)	0 (0%)	1 (25%)	3 (75%)	0 (0%)	1 (25%)	2 (50%)	2 (50%)	0 (0%)

VI. CONCLUSÕES

O reconhecimento de placas de sinalização de trânsito é uma tarefa importante na construção de sistemas de apoio ao tráfego de veículos e, fundamental na implementação de veículos autônomos, porém, a implementação de algoritmos que apresentam tempos de processamento razoáveis continua sendo um grande desafio. O trabalho de Silva et al. [2] apresenta taxas de acerto razoáveis, entretanto, a aplicação do algoritmo SIFT em todas as imagens torna o tempo de processamento inadequado para aplicações em tempo real. Assim, a adaptação do algoritmo AdaBoost para selecionar as áreas com maior potencial para conter placas de sinalização, e a aplicação do algoritmo SIFT, apenas nestas áreas, conseguiu manter a qualidade dos resultados e, ainda, diminuiu consideravelmente o tempo de processamento. Esta última abordagem consegue uma taxa de acerto ainda maior do que as últimas duas propostas e ainda consegue diminuir sensivelmente o tempo de processamento das imagens.

REFERÊNCIAS

- [1] P. Viola and M.J. Jones, "Robust real-time object detection", In: Second Int. Workshop on Statistical and Computational Theories of Vision – Modeling, Learning, Computing and Sampling, 2001.
- [2] F.A. Silva, A.O. Artero, M.S.V. Paiva e R.L. Barbosa, "Uma Metodologia para Detectar e Reconhecer Placas de Sinalização de Trânsito", In: VIII Workshop de Visão Computacional (WVC 2012). Anais... Goiás: UFG, 2012.
- [3] A. Martinović, G. Glavaš, M. Juribašić, D. Sutić and Z. Kalafatić, "Real-time Detection and Recognition of Traffic Signs", Proc. 33rd Inter. Conv. on Information and Communic. Technology, Electronics and Microelectronics – MIPRO 2010, v. III, 2010, p. 247-252.
- [4] X. Baró, S. Escalera, J. Vitrià, O. Pujol and P. Radeva, "Traffic Sign Recognition using Evolutionary Adaboost Detection and Forest-ECOC classification", In: IEEE Trans. on Intelligent Transportation Systems. 2009. p. 113-126.
- [5] S.-Y. Chen and J.-W. Hsieh, "Boosted road sign detection and recognition", In: International Conference on Machine Learning and Cybernetics, Proceedings... v. 7, pp. 3823–3826, July 2008.
- [6] I.M. Creusen, R.G.J. Wijnhoven, E. Herbschleb and P.H.N. de With, "Color Exploitation in Hog-Based Traffic Sign Detection", Proc. IEEE 17th Int. Conf. on Image Processing. Hong Kong, 2010, pp. 2669-2672.
- [7] G. Loy and A. Zelinsky, "Fast Radial Symmetry for Detecting Points of Interest", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 8, 2003, pp. 959-973.
- [8] R. Belaroussi and J.-P. Tarel, "A real-time road sign detection using Bilateral Chinese Transform", Proc. International Symposium on Visual Computing, ISVC, 2009, pp. 1161-1170.
- [9] A. Arlicot, B. Soheilian and N. Paparoditis, "Circular Road Sign Extraction from Street Level Images Using Colour, Shape and Texture Database Maps". In: Stilla U, Rottensteiner F, Paparoditis N (Eds) CMRT09. IAPRS, Vol. 38, France, 2009.
- [10] C.F. Paulo and P.L. Correia, "Traffic Sign Recognition Based on Pictogram Contours", Ninth Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'2008), Klagenfurt, Austria, 2008, pp. 67-70.
- [11] N. Barnes, A. Zelinsky, "Real-Time Radial Symmetry for Speed Sign Detection", Proc. Intelligent Vehicles Symposium, 2004, pp. 566–571.
- [12] H. Fleyeh, "Road and Traffic Sign Color Detection and Segmentation - A Fuzzy Approach", MVA2005 IAPR Conf. on Machine Vision Applic., Tsukuba Science City, Japan, May 16-18, 2005, pp. 124-127
- [13] K. Brkić, "An overview of traffic sign detection methods", Department of Electronics, Microelectronics, Computer and Intelligent Systems Faculty of Electrical Engineering and Computing Unska 3, 10000 Zagreb, Croatia, Tech. Rep., 2010.
- [14] X. Baró and J. Vitria, "Fast traffic sign detection on greyscale images", Recent Advances in Artificial Intelligence Research and Development, p. 69-76, October 2004.
- [15] K. Brkić, A. Pinz, and S. Šegvić, "Traffic sign detection as a component of an automated traffic infrastructure inventory system", Stainz, Austria, 2009.
- [16] Y. Freund and R.E. Schapire, "A Short Introduction to Boosting", Journal of Japanese Society for Artificial Intelligence, 1999, p. 771-780.
- [17] R. Lienhart, and J. Maydt, "An extended set of Haar-like features for rapid object detection", In IEEE ICIP 2002, 2002, pp. 900–903.
- [18] N. Otsu, "A threshold selection method from gray-level histograms". IEEE Transactions on System, Man and Cybernetics. v. 9, n. 1, pp. 62-66, 1979.
- [19] F.A. Silva, A.C.P. Nascimento, M.S.V. Paiva, A.O. Artero, M.A. Piteri, R.L. Barbosa. "Detecção e Reconhecimento de Placas de Sinalização de Trânsito usando os algoritmos AdaBoost e SIFT". In: X Workshop de Visão Computacional (WVC 2014). Anais... Uberlândia: UFU, pp. 179-184, 2014.
- [20] F.A. Silva, A.O. Artero, M.S.V. Paiva, R.L. Barbosa. "Um algoritmo rápido para o reconhecimento de caracteres". In: VII Workshop de Visão Computacional (WVC 2011). UFPR, Curitiba, 2011, p. 149–154.