

Sistema de reconhecimento de sinais de trânsito de tempo real no Raspberry Pi

Esdras Vitor Silva Pinto
Engenharia Elétrica - UFMG - esdrasvitorsi@gmail.com

1. Introdução

Muitos veículos atuais, contam com sistemas sofisticados que visam auxiliar a dirigibilidade do motorista, aumentando a segurança e o conforto do veículo. Tais tecnologias são conhecidas como sistemas de assistência avançada de condução (ADAS - Advanced Driver Assistance Systems). Os ADAS fornecem informações relevantes aos motoristas sobre o ambiente ao redor do veículo, aumentando a sua segurança e dirigibilidade. Dentre os vários sistemas ADAS, como detectores de faixa [2] e estacionamento auxiliado [3], destaca-se os sistemas capazes de detectar e reconhecer sinais de trânsito[1]. Este último, é referido na literatura como sistemas de reconhecimento de sinais de trânsito, (TSR - Traffic Sign Recognition).

Sistemas TSR captam imagens através de uma câmera focalizada na visão frontal do veículo, ou seja, a câmera possui a mesma visão que o motorista. O algoritmo dos sistemas TSR é composto basicamente por duas partes. A primeira parte consiste na identificação de potenciais regiões que podem conter sinais de trânsito em um frame [1]. A segunda envolve a análise das regiões de possíveis sinais de trânsito, classificando e reconhecendo o potencial sinal. Portanto, o algoritmo é composto por uma fase de detecção e outra de reconhecimento.

Na etapa de detecção (ou identificação), são empregadas informações de espaço de cores para a seleção de áreas de um frame que podem ser sinais de trânsito. Como os sinais de trânsito possuem formas e cores definidas, o uso de espaço de cores se apresenta conveniente em suas identificações. Por exemplo, a maioria dos sinais de trânsito brasileiros possuem as cores branca, azul, amarelo, verde e vermelho. Adicionalmente, as formas mais comumente utilizadas são circulares e retangulares. Com as

informações de cores aliadas às informações de geometria, as regiões mais propensas de ser um sinal de trânsito são identificadas [4].

Na parte de reconhecimento, existem duas sub etapas. A primeira é a extração de características das regiões provenientes da fase de identificação. Essas características são pontos na imagem que são suficientemente estáveis e característicos da imagem em questão. Vários algoritmos têm sido utilizados para extrair características de imagens, como histograma de gradientes orientados HOG (histogram of oriented gradients) [5,6], extração robusta acelerada (SURF - speeded-up robust feature), característica com invariância de escala (SIFT) [7], dentre outros. Uma vez que conhecemos os pontos característicos de um objeto, pode-se buscar por essas características em outra imagem. É exatamente isso que é feito na segunda etapa do reconhecimento, o qual envolve aprendizado de máquina.

Para que o algoritmo do TSR seja capaz de reconhecer sinais de trânsito, ele precisa ser treinado. O algoritmo de extração de características é aplicado em um conjunto de imagens que representam genuinamente um sinal de trânsito. Essas características são tratadas e armazenadas em um banco de dados. Uma vez que se conhecem as características dos sinais de trânsito, o reconhecimento ocorre por meio de comparação das características salvas no banco de dados com aquelas que foram extraídas da região que potencialmente representa um sinal de trânsito. Se houver uma correspondência entre essas características acima de um valor limiar estabelecido, o sinal de trânsito pôde ser reconhecido. Por outro lado, se não houver correspondência suficiente, nenhum sinal será reconhecido na área de interesse.

Uma característica importante para um sistema TSR é o grau de confiabilidade. Para que esse sistema seja parte dos ADAS, é crucial o reconhecimento correto de sinais de trânsito nas mais variadas situações do ambiente. Ou seja, os sistemas TSR devem ser invariantes à iluminação, onde serão capazes de reconhecer sinais tanto durante o dia quanto à noite. Além disso, esses sistemas precisam ser capazes de detectar sinais de trânsito parcialmente ocultado, desgastado com o tempo, e inclinado. Outro desafio que os sistemas TSR enfrentam é a operação em tempo real. Para que o reconhecimento seja possível até

mesmo em velocidades elevadas, existe um tempo máximo na ordem de milissegundos para o TSR identificar e reconhecer os sinais de trânsito em um frame. Como o algoritmo envolve varias etapas de processamento de imagem, operar em tempo real é um desafio.

1.1 Motivação

Os sinais de trânsito são importantes mecanismos para garantir a dinâmica e segurança no trânsito. Sempre que um sinal de trânsito é ignorado, como por exemplo, parada obrigatória e limite máximo de velocidade, aumenta-se o risco de um potencial acidente. A simples observância aos sinais de trânsito contribui significativamente para o aumento da segurança nas vias e rodovias do país. Neste contexto de segurança e auxilio ao motorista, sistemas TSR apresentam grande relevância. Tais sistemas matem o motorista ciente das informações visuais dos sinais de trânsito, emitindo alertas no painel e auxiliando o motorista a observá-los. Por exemplo, o sistema TSR pode alertar que há uma parada obrigatória à frente ou se a velocidade do veículo está acima da permitida no local.

Além de ser um importante sistema de assistência avançada de condução, sistemas TSR são também parte chave dos carros autônomos. Também aliado à segurança, carros autônomos estão se tornando uma realidade. Empresas como Google, já estão em fase avançadas de testes, e a confiabilidade tem sido surpreendente. Portanto, o reconhecimento de sinais de trânsito também é utilizado em carros autônomos.

1.2 Objetivos

Nesse trabalho de final de curso, será implementado a etapa de reconhecimento do algoritmo TSR no Raspberry Pi. A motivação pela escolha do Raspberry Pi se deve ao fato de ser uma plataforma com grande desempenho computacional, baixo custo e compacto, podendo ser facilmente embarcado em um veículo. Adicionalmente, com o uso do Raspberry Pi é possível desenvolver uma solução completa do sistema TSR, incluindo captura de imagem por câmera instalada no veículo, processamento para identificar e reconhecer sinais de trânsito, e mostrar os resultados em um display LCD. No vasto arsenal de periféricos para Raspberry Pi, existem disponíveis no mercado, câmeras e displays LCD acessíveis e

de baixo custo. Para a extração de características, será utilizado o algoritmo SURF. Conforme [8], SURF possui invariância à escala, invariância a rotação e pode ser utilizado para reconhecimento de objetos parcialmente ocultos. Essas características do SURF, permitem a sua utilização na implementação de sistemas TSR robustos, que sejam capazes de lidar com condições variadas de ambiente, bem como diferentes estados de conservação e posicionamento dos sinais de trânsito.

2. Revisão bibliográfica

Sistemas de identificação e reconhecimento de sinais de trânsito têm resultado em várias pesquisas na área de visão computacional. Várias plataformas de implementação foram propostas até o momento da escrita desse trabalho, incluindo computadores de uso geral, FPGA, chips customizados para processamento de imagens e processamento digital de sinal, sistemas embarcados dedicados, como Raspberry Pi, e também processadores gráficos (GPU - Graphic Processing Unit).

As implementações dos sistemas TSR em computadores de uso geral, têm se mostrado rápidas o suficiente para permitir detecção em tempo real. Conforme apresentado em [9], o processo de reconhecimento, o qual envolve extração de características da região de interesse por meio do algoritmo SIFT e o tempo de classificação, foi realizado com apenas 81ms em um computador de uso geral operando a 2.4GHz com dois núcleos e sistema operacional Linux. Além da rápida execução da fase de reconhecimento, a metodologia proposta neste trabalho atingiu uma confiabilidade de 95%. Embora tenham sido obtidos resultados promissores, tanto de desempenho quando de confiabilidade, a solução apresentada é difícil de ser embarcada em um veículo.

Ainda no contexto dos computadores de uso geral, algumas implementações utilizam processadores gráficos (GPU) para paralelizar maciçamente os algoritmos de detecção e reconhecimento de sinais de trânsito. Essa paralelização aumenta consideravelmente o desempenho. No trabalho [10], um sistema TSR foi implementado na plataforma Tesla K20 GPU da NVIDIA. A execução da detecção e reconhecimento levou entre 13 a 17 ms, apresentando uma taxa média de 21 frames por segundo. Entretanto, o uso de GPU só se justifica para imagens de alta resolução, conforme mencionado pelos

autores. Isto ocorre devido o overhead da linguagem CUDA necessário para instanciar de núcleos na GPU, o que impacta no tempo de processamento. Para compensar esse overhead, a imagem precisa ser de alta resolução, como 1236 x 1628 utilizado no trabalho. Por fim, os resultados obtidos nesse trabalho do ponto de vista de confiabilidade são bastante promissores, onde atingiu-se uma taxa de sucesso de reconhecimento de 93,77%.

Soluções alternativas aos computadores de uso geral e GPU, empregam hardware específico e possuem a vantagem de serem mais viáveis para uma situação real de uso do TSR em um veículo. Muitas dessas soluções se baseiam em hardware reconfigurável, como FPGA (Field-Programmable Gate Array). O trabalho [11] apresenta um sistema TSR implementado em FPGA, em específico, na FPGA Spartan-6. A fase de identificação de áreas de interesse é baseada no espaço de cores, e a parte de reconhecimento é baseada na extração e classificação de características. Embora o sistema seja passível de ser embarcado em um veículo, ele é capaz de reconhecer apenas sinais de limite de velocidade.

O grande benefício da FPGA é a flexibilidade. Caso haja alguma melhoria ou modificação nos algoritmos de detecção ou de reconhecimento, é possível reconfigurar a FPGA para executar o novo algoritmo. Adicionalmente, as FPGAs modernas compartilham o interior do chip com um ou mais núcleos de processadores físicos. Isso permite obter um equilíbrio entre desempenho e flexibilidade, onde partes mais lentas do algoritmo podem ser realizadas em hardware, e partes que necessita de flexibilidade podem ser implementada em software. De fato, uma abordagem envolvendo hardware e software embarcado é apresentado em [1].

Alguns sistemas TSR foram implementados em chips dedicados de processamento digital de sinal (DSP - Digital Sign Processing). Conforme apresentado em [2], o chip de processamento digital de sinal OMAP L138 da Texas Instruments (TI) foi empregado em um sistema TSR, onde foi obtido um tempo total de processamento de 300ms por frame. Ainda que o tempo seja promissor para aplicação de tempo

real, o algoritmo de reconhecimento utilizado, baseado em comparação de modelos, não é robusto o suficiente para detectar sinais de trânsito parcialmente ocultos ou rotacionados [1].

Duas qualidades importantes para sistemas TSR é a operação em tempo real e robustez na detecção e reconhecimento. A maioria das soluções para sistemas TSR, seja em software ou hardware, operam em tempo real. Assim, é possível garantir que a detecção de sinais de trânsito será possível mesmo quando o veículo estiver em alta velocidade. Por alto lado, a confiabilidade ou robustez de detecção ainda está abaixo do desejado. O grau de confiabilidade dos sistemas TSR analisados variam entre 90 e 96%. Por se tratar de uma aplicação crítica, onde uma classificação errada ou simplesmente o não reconhecimento do sinal pode causar graves consequências, como acidentes ou comportamento inesperado de um veículo autônomo, espera-se que a confiabilidade seja muito próxima de 100%.

Complementar os requisitos de operar em tempo real e ter confiabilidade de tecnicamente 100%, os sistemas TSR precisam ser embarcados em um veículo para terem utilidade no mundo real. As soluções discutidas envolvendo computadores de uso geral, embora possam ser embarcadas no veículo, não são interessantes do ponto de vista de produção, devido ao alto custo e espaço ocupado. Por outro lado, soluções envolvendo sistemas embarcados dedicados, como FPGAs, são mais atraentes do ponto de vista de serem integradas aos veículos, pois são compactos e dependendo do tipo de FPGA usado, o custo de produção é menor do que computadores de uso geral. Essa é uma das principais razões na qual grande parte das pesquisas destinada a sistemas TSR serem desenvolvidas em sistemas embarcados.

Esse presente trabalho é inspirado em um trabalho similar desenvolvido durante meu intercâmbio acadêmico no Instituto de Tecnologia de Illinois [1], no qual foi implementado o algoritmo de reconhecimento do TSR em um sistema embarcado composto por uma FPGA no kit de desenvolvimento ZedBoard da Xilinx. O algoritmo de reconhecimento usa o SURF para extrair características das imagens e a classificação é feita pelo algoritmo k-means. Todo algoritmo de reconhecimento é implementado em software, embora se tenha a possibilidade de uso da FPGA para desenvolver aceleradores de hardware.

Os resultados obtidos não são suficientes para operação em tempo real. Em algumas situações, o tempo total para o reconhecimento é maior do que 2s. Devido ao elevado custo e à dificuldade de se encontrar uma câmera destinada à Zedboard, optou-se por implementar toda fase de reconhecimento em uma outra plataforma mais promissora no presente momento: Raspberry Pi.

Raspberry Pi são computadores portáteis e apresentam alto desempenho. Adicionalmente, possui uma crescente e engajada comunidade que disponibilizam uma vasta documentação e suporte ao desenvolvimento de uma aplicação para o Raspberry Pi. Além desse apoio ao desenvolvedor, o Raspberry Pi é de baixo custo, compacto e possui uma grande variedade de periféricos que podem ser adicionados ao sistema, como, por exemplo, uma câmera. Por ser atualmente uma plataforma difundida entre os profissionais e entusiastas da área de computação e sistemas embarcados, pode-se encontrar facilmente na internet, lojas que vendem tanto o Raspberry Pi quanto câmeras e displays LCD para serem utilizadas em conjunto com o Raspberry. A solução utilizando Raspberry Pi possui a grande vantagem de ser possível construir todo sistema embarcado no veículo, incluindo a câmera, o processamento, e um display LCD para mostrar os resultados, já que todos esses componentes são acessíveis do ponto de vista de serem encontrados no mercado e são de baixo custo.

3. Metodologia

Conforme definido na introdução, o objetivo do presente trabalho é implementar o algoritmo de reconhecimento de um sistema TSR usando o SURF como extrator de características na plataforma Raspberry Pi. Nessa seção, será apresentada a ideia geral do funcionamento do algoritmo de reconhecimento, e, nas seções subsequentes, maior detalhamento será fornecido. Embora não será implementado a fase de identificação ou detecção, a qual procede à fase de reconhecimento, uma breve introdução é feita a seguir.

3.1 Detecção ou identificação

A identificação de sinais de trânsito em uma imagem é feita usando-se segmentação de cores. É comum as câmeras disponibilizarem frames no formato RGB. Este formato baseia-se na combinação das

cores vermelha (R-Red), verde (G-Green) e azul (B-Blue) para produzir todas demais cores conhecidas. Entretanto, o espaço de cores RGB não é apropriado para segmentar uma imagem baseada em uma determinada cor, já que a informação de cor está misturada em três componentes.

O espaço de cor HSI (H - Hue, S - Saturation, I - Intensity) é mais apropriado para a segmentação de imagens por meio de cores. A componente hue especifica a cor propriamente dita, enquanto que a saturação indica o grau de luz branca presente na cor. Os sinais de trânsito possuem formas e cores bem definidas. O processo de segmentação por cores para identificar sinais de trânsito, usa o fato da maioria dos sinais possuírem as cores vermelho, amarelo, branco, verde e azul para definir regiões de interesse. Assim, conforme apresentado em [12], pode-se estabelecer limiares para a componente hue de modo a detectar apenas as cores que se deseja obter. Trabalhando com a componente hue, torna-se o sistema mais robusto aos efeitos de iluminação do ambiente, pois o sistema HSI separa o nível de saturação da cor propriamente dita.

Uma vez estabelecido os limiares para as cores mais comuns de sinais de trânsito, a imagem em HSI é transformada em uma imagem binária, de modo que todos os pixels que possuem valores de hue dentro dos limiares estabelecidos seja '0' e os demais pixels sejam '1'. A figura a seguir mostra a conversão de um frame contendo dois sinais de trânsito para uma imagem binária. Nesta imagem, apenas a cor vermelha está sendo levada em consideração. Para detectar outras regiões de interesse, basta adicionar limiares para a componente hue para as cores azul, verde, amarelo e branco.

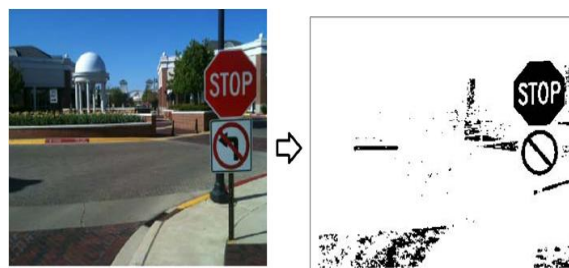


Figura 1: Conversão de um frame para imagem binária, onde apenas pixels com as cores desejadas na imagem de entrada possuem a cor preta na imagem binária, e as demais regiões possuem cor branca¹

¹ Imagem obtida durante o desenvolvimento do trabalho [1].

Note que o processo de conversão de um frame para uma imagem binária pode conter vários ruídos, já que existem vários outros objetos que também são das mesmas cores dos sinais de trânsito. Para eliminar parte do ruído, são aplicados filtros morfológicos, conforme apresentado em [1]. A redução de ruído é importante para evitar falsos positivos e economizar tempo de processamento na fase de reconhecimento. A figura a seguir mostra o resultado da aplicação de filtros morfológicos para redução de ruídos.

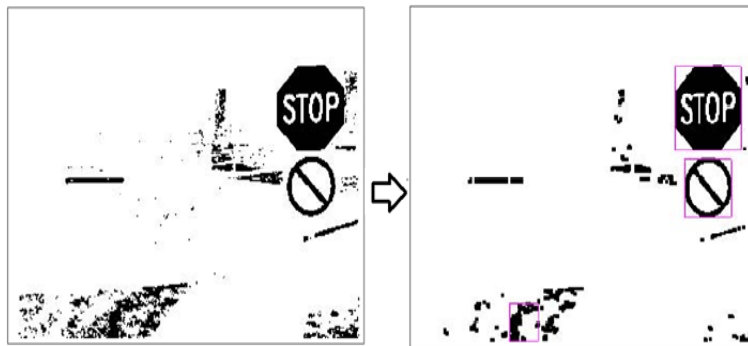


Figura 2: Redução de ruídos na imagem binária através da aplicação de filtros morfológicos

Após a eliminação de parte do ruído, a última etapa é varrer a imagem binária à procura de agrupamento de pixels. Caso o agrupamento seja menor que um valor previamente determinando, certamente não será possível reconhecer sinais de trânsito nesta região, e, portanto, ela será tratada como ruído. Por outro lado, caso o agrupamento seja denso o suficiente para realizar a detecção, definimos esse agrupamento como potencialmente composta por um sinal de trânsito, ou seja, esse agrupamento se torna uma área de interesse. Todas as áreas de interesse são enviadas para a etapa de reconhecimento. A figura a seguir, também fruto de trabalhos anteriores [1], mostra sucintamente o processo de identificação de potenciais áreas que podem conter sinais de trânsito.

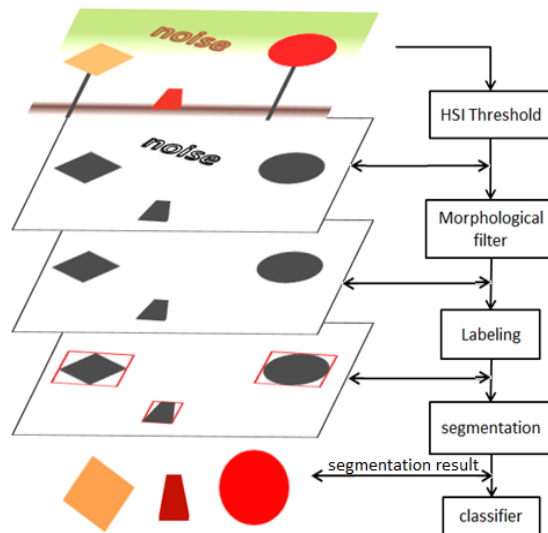


Figura 3: Sequência de operações na fase de detecção de sinais de trânsito

3.2 Reconhecimento

Na etapa de reconhecimento, é utilizada a metodologia de extração de características. Essas características podem ser vistas como pontos matemáticos (pixels) estáveis na imagem que são possíveis de serem encontrados em uma imagem similar. Neste caso, as características das regiões de interesse provenientes da etapa de detecção são extraídas e posteriormente comparadas com características dos modelos de sinais de trânsito no qual o sistema foi treinado. Ou seja, características de imagens genuínas de sinais de trânsito são extraídas na fase de treinamento do algoritmo e salvas no banco de dados. Na etapa de classificação, as características extraídas das áreas de interesse são comparadas com as características salvas no banco de dados.

Como o reconhecimento se baseia na extração de características, o tempo de execução na parte de reconhecimento é fortemente afetado pelo desempenho do algoritmo de extração. Nesse projeto, é utilizado o algoritmo SURF. Conforme [8], SURF é dito invariante à escala, rotação e oclusão parcial. Portanto, SURF é promissor para sistemas TSR, já que tais sistemas precisam ser capazes de reconhecer sinais em diferentes escalas e também saber lidar com sinais parcialmente ocultos ou não alinhados corretamente. A determinação de características através do SURF envolvem várias etapas e será destinada

uma seção específica para detalhar o seu funcionamento. No final do algoritmo, cada característica será representada por um descritor que consiste em um vetor de número real de 64 posições.

Adicionalmente, o algoritmo SURF é consideravelmente mais rápido que algoritmos similares de extração de características, como o SIFT. Portanto, SURF possibilita reconhecimento em tempo real mesmo em sistemas embarcados, o que é crucial para sistemas TSR. A figura a seguir ilustra as características obtidas através da aplicação do SURF no sinal de trânsito PARE dos Estados Unidos (STOP).



Figura 4: Círculos verdes indicam as características extraídas pelo SURF para o sinal PARE dos Estados Unidos

Uma vez que as características dos sinais de trânsito foram extraídas, elas precisam ser classificadas, etapa esta comumente referida como classificação. Na parte de classificação, tem-se um banco de dados composto por características extraídas de imagens modelos de sinais de trânsito que são utilizadas para treinamento do algoritmo. Mais especificamente, as características extraídas das regiões que potencialmente possuem sinais de trânsito, são comparadas com as características das imagens usadas no treinamento do algoritmo para obter correspondência entre essas características. Essa correspondência consiste em achar a menor distância entre os descritores das características encontradas na região de interesse daquelas salvas no banco de dados. Se a distância entre essas características é menor do que um valor liminar determinado, valida-se a correspondência entre as características. A figura a seguir ilustra a

correspondência entre o modelo do sinal proibido virar à esquerda no banco de dados e um candidato a sinal de trânsito.

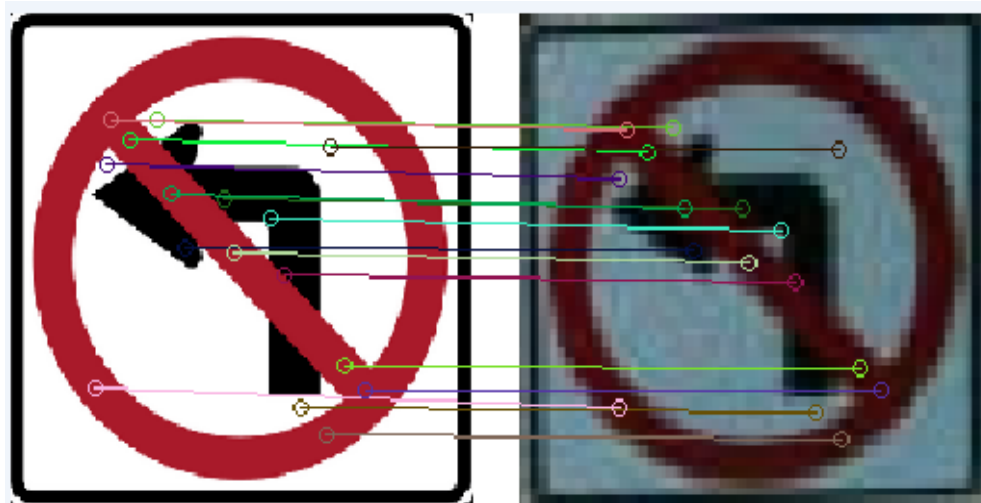


Figura 5: Correspondência entre as características do modelo no banco de dados para o sinal proibido virar à esquerda (esquerda) e as características do candidato a ser classificado como sinal de trânsito (direita).

Adicionalmente, o número de correspondências válidas são também levadas em consideração. Particularmente, a razão entre as o número de correspondências validadas para um dado sinal de trânsito e o número total de características no banco de dados para o sinal em questão é considerado. Se esta razão for acima de certo limiar estabelecido, a área de interesse pôde ser reconhecida e é designada ao sinal de trânsito correspondente. Para aperfeiçoar o tempo de execução da etapa de classificação, o banco de dados é estruturado usando o algoritmo K-means.

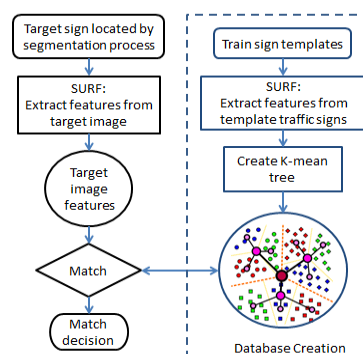


Figura 6: Etapas realizadas no processo de reconhecimento de um sinal de trânsito²

² Imagem obtida durante o desenvolvimento do trabalho [1].

Neste trabalho, toda etapa de reconhecimento, incluindo extração de características e classificação, será implementada no sistema embarcado Raspberry Pi.

4. Referências

- [1] Yan, H.; Virupakshappa, K.; Pinto, E. V. S.; Oruklu, E. Hardware/Software Co-Design of a Traffic Sign Recognition System Using Zynq FPGAs. *Electronics* (Basel), 01 December 2015, Vol.4(4), pp.1062-1089.
- [2] Al Smadi, T. Real-Time Lane Detection for Driver Assistance System. *Circuits and Systems*, 2014, 5, 201-207.
- [3] Kaur, R.; Singh, B. Design and implementation of car parking system on FPGA. *International Journal of VLSI design & Communication Systems (VLSICS)* Vol.4, No.3, June 2013
- [4] Waite, S.; Oruklu, E. FPGA-Based Traffic Sign Recognition for Advanced Driver Assistance Systems. *Transp. Technol.* 2012, 3, 1–16.
- [5] Liang, M.; Yuan, M.; Hu, X.; Li, J.; Liu, H. Traffic sign detection by ROI extraction and histogram features-based recognition. In *Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN)*, Dallas, TX, USA, 4–9 August 2013; pp. 1–8.
- [6] Wang, G.; Ren, G.; Wu, Z.; Zhao, Y.; Jiang, L. A robust, coarse-to-fine traffic sign detection method. In *Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN)*, Dallas, TX, USA, 4–9 August 2013; pp. 1–5.
- [7] Lowe, D.G. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Corfu, Greece, 30 September–2 October 1999; Volume 2, pp. 1150–1157.

- [8] Bay, H.; Tuytelaars, T.; Gool, L.V. SURF: Speeded up robust features. *Comput. Vis. Image Underst.* 2008, 110, 346–359.
- [9] Ren, F.; Huang, J.; Jiang, R.; Klette, R. General traffic sign recognition by feature matching. *Image and Vision Computing New Zealand, 2009. IVCNZ'09.* In *Proceedings of the 24th International Conference, Willington, New Zealand, 23–25 November 2009*; pp. 409–414.
- [10] Chen, Z.; Huang, X.; Ni, Z.; He, H. A GPU-based real-time traffic sign detection and recognition system. In *Proceedings of the 2014 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS), Orlando, FL, USA, 9–12 December 2014*; pp. 1–5.
- [11] Schwiegelshohn, F.; Gierke, L.; Hubner, M. FPGA based traffic sign detection for automotive camera systems. In *Proceedings of the 2015 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), Bremen, Germany, 29 June–1 July 2015*; pp. 1–6.
- [12] Gómez-Moreno, H.; Maldonado-Bascón, S.; Gil-Jiménez, P.; Lafuente-Arroyo, S. Goal Evaluation of Segmentation Algorithms for Traffic Sign Recognition. *IEEE Trans. Intell. Transp. Syst.* 2010, 11, 917–930.