Vehicle Speed Monitoring using Convolutional Neural Networks

V. Barth, R. de Oliveira, M. de Oliveira, and V. do Nascimento

***Abstract*—Recently, Computer Vision Techniques have been pushing the development of robust traffic monitoring systems. Such methods utilize images captured by video cameras to infer important traffic features, such as vehicle speed and traffic density. Frame Subtraction is currently the most used method to detect vehicles in a video stream, but there are scenarios where this method provides poor accuracy, given their struggle in handling disturbances caused by lighting changes, pedestrians in the scene, etc. In order to improve the accuracy of Traffic Monitoring Systems (TMS), this paper proposes a novel TMS design and implementation in which a Convolutional Neural Network is used to replace Frame Subtraction methods in the vehicles detection task. The results show up to 12% improvements on Vehicle Detection in comparison with Frame Subtraction-based systems, proving its effectiveness on challenging scenarios, while maintaining an error rate of 5% for speed detection.**

***Index Terms*—CNN, Traffic analysis, Urban traffic, Overspeed detection, Computer vision.**

I. Introdução

SEGUNDO a Organização Mundial da Saúde, são registradas anualmente, em todo o mundo, cerca de 1 (um) milhão de fatalidades em acidentes de trânsito, e metade desses incidentes vitimizam pedestres, ciclistas ou motociclistas. A redução de limites de velocidade conjuntamente com a aplicação mais rigorosa de leis de trânsito são apontados como soluções para a diminuição do número de acidentes de trânsito [1].

Ainda que possam surtir efeito positivo, equipamentos de medição de excesso de velocidade utilizados em autoestradas, como laços indutivos, cabos piezelétricos, *laser* ou *sonar* não se mostram adequados para vias urbanas, dados a área de cobertura e os custos de aquisição, instalação e manutenção destes sistemas [2]. Além do seu custo, estes sistemas podem ser facilmente burlados por veículos se deslocando no sentido contrário ao da via, o que aumenta o risco de acidentes [3].

V. B. O. Barth, Instituto Federal de Mato Grosso, Cuiabá, Mato Grosso, Brasil (E-mail: vitor.barth@gmail.com).

R. de Oliveira, Instituto Federal de Mato Grosso, Cuiabá, Mato Grosso, Brasil (E-mail: ruy@cba.ifmt.edu.br).

M. A. de Oliveira, Instituto Federal de Mato Grosso, Cuiabá, Mato Grosso, Brasil (E-mail: mario.oliveira@cba.ifmt.edu.br).

V. E. do Nascimento, Instituto Federal de Mato Grosso, Cuiabá, Mato Grosso, Brasil (E-mail: valtemir.nascimento@cba.ifmt.edu.br).

A área de Inteligência Artificial (IA) tem sido explorada como forma de simplificar o desenvolvimento de sistemas inteligentes multivariáveis, onde há a necessidade de monitoramento de vários sensores simultaneamente. Sistemas inteligentes são capazes de inferir padrões de comportamento de grandes bases de dados sem que haja necessidade de se implementar algoritmos estáticos, os quais normalmente são incapazes de lidar com as incertezas presentes na informação a ser processada [4]. A visão computacional é uma área da IA que busca extrair informações de alto nível, próximas àquelas inferidas por seres humanos, a partir de fontes de imagens digitais, como câmera de segurança e imagens de satélite [2].

Os avanços recentes no campo de visão computacional se devem em grande parte à popularização de câmeras digitais e ao grande poder computacional dos processadores atuais. Dessa forma, tornou-se viável o desenvolvimento de sistemas de detecção de velocidade não intrusivos, mais baratos e robustos.

Diversos trabalhos utilizam visão computacional para identificar infrações e acidentes, contar veículos e aferir velocidade média [5]–[8]. Contudo, tais sistemas, desenvolvidos para vias de alto fluxo de veículos, não são eficazes em locais onde a circulação de veículos ocorre em baixas velocidades ou de modo desorganizado, na medida em que pode haver pedestres, por exemplo, na cena.

Independentemente dos equipamentos ou métodos utilizados, o bom funcionamento de um Sistema de Monitoramento de Tráfego (SMT) depende de sua capacidade de rastrear veículos com precisão. O problema de rastreamento de veículos consiste em localizar veículos em movimento utilizando uma câmera digital. As abordagens encontradas recorrentemente na literatura utilizam a técnica de subtração de plano de fundo para detectar as Áreas de Interesse (comumente chamadas de *blobs* ou *Regions of Interest – RoI*) [5]–[8].

A técnica de subtração de plano de fundo se baseia na ideia de identificar porções da imagem que permanecem inalteradas em quadros sucessivos. Posteriormente, o modelo de plano de fundo é subtraído de cada quadro do vídeo, e as áreas que restam são *blobs* que indicam os veículos presentes na cena [9].

Entre as abordagens possíveis para a criação do modelo de plano de fundo, as mais difundidas utilizam diferenciação de quadros [10]–[12] ou recursos estatísticos [13]–[22]. Estes métodos, no entanto, não se adaptam instantaneamente ao

ambiente, e a precisão não é mantida em ambientes com presença de pedestres ou outros obstáculos, tanto por falhas de detecção quanto por falsos-positivos.

Sobral & Vacavant [9] e Choudhury et Al [23] realizaram testes em diversos cenários, verificando a influência da vegetação, sombras, condições climáticas e mudanças na iluminação utilizando diversos métodos de Subtração de Plano de Fundo. Foi comprovado experimentalmente que tais condições são limitadoras para o funcionamento das técnicas avaliadas.

Além das técnicas de segmentação de plano de fundo citadas anteriormente, recentemente as Redes Neurais Convolucionais (*Convolutional Neural Networks – CNNs*) também têm sido utilizadas para a detecção e classificação de objetos em imagens [24]–[28].

Ao contrário das abordagens anteriores, CNNs são capazes de detectar precisamente os veículos presentes em um único quadro de vídeo estático. Assim como outros métodos baseados em Aprendizagem de Máquina, as Redes Neurais Convolucionais classificam e detectam objetos por meio da busca de *features*, i.e., características comuns à uma classe de objetos, as quais são extraídas por processo de análise de uma base de dados.

Ao resolver problemas semelhantes, CNNs promoveram resultados mais precisos que o estado da arte [24]–[28]. Espera-se que o uso de uma CNN para detecção de veículos em um SMT possa aumentar sua eficácia, sobretudo em ambientes com oclusões, como a presença de pedestres, sombras, postes, entre outros.

Por isso, a fim de comprovar a eficácia da CNN em um SMT, esse trabalho propõe uma abordagem que utiliza CNN para fazer a detecção de veículos de forma eficiente num sistema de monitoramento de tráfego. A principal vantagem desta proposta é evitar os problemas de detecção de veículos devido às oclusões, comuns aos sistemas baseados em modelo de plano de fundo.

Dessa forma, o sistema proposto possui as seguintes características: i) permite a presença de pedestres e outros veículos estacionados na imagem, ii) monitora o deslocamento do veículo em ambos os sentidos da via, iii) tolera faixas de rolamento que não sejam delimitadas, iv) permite variações nas condições de iluminação da cena, v) pressupõe apenas um veículo em movimento na cena. Diferentemente dos trabalhos citados, tais características garantem a precisão do sistema em áreas onde a sinalização é precária e propensa a oclusões, comuns em áreas urbanas.

Por ser não-invasivo, o método proposto diminuirá os custos de implantação de sistemas de detecção de velocidade, o que pode contribuir com a disseminação desses sistemas em localidades de baixo poder aquisitivo, e o uso de Redes Neurais Convolucionais permitirá o desenvolvimento de sistemas de detecção de velocidade mais robustos e precisos, assim ampliando a área de monitoramento para locais cuja precisão não era garantida, como ruas com grande quantidade de pedestres ou estacionamentos.

As demais seções deste artigo estão estruturadas como segue: a seção 2 descreve os principais elementos e técnicas utilizados e sistemas de monitoramento de tráfego por imagem. Na seção 3 é apresentado o método proposto, na seção 4 é apresentada a avaliação de desempenho para o método proposto. Por fim, na seção 5, são apresentadas as conclusões.

II. Elementos Inerentes Aos Sistemas De Monitoramento De Tráfego Por Imagem

Nesta seção, são apresentados os principais elementos utilizados para se obter métricas de tráfego por meio de visão computacional. As técnicas foram agrupadas de acordo com sua funcionalidade: a) detecção de veículos, b) rastreamento de veículos, c) medição de velocidade.

A. Detecção de Veículos

Algoritmos de Detecção de Veículos tem por finalidade localizar todos os veículos presentes em uma imagem. A maneira mais simples e eficiente de realizar tal reconhecimento é através da Subtração de Planos de Fundo.

Para a visão computacional, plano de fundo é definido como o conjunto de *pixels* de uma cena que permanece fixo através do tempo. O processo de separação do plano de fundo das áreas de interesse, i. e., daquelas onde houve mudanças, é denominado subtração de plano de fundo.

Os métodos de detecção de veículos mais comuns na literatura utilizam técnicas de subtração de plano de fundo por meio da diferenciação de quadros [10]–[18], [22], a qual consiste em criar um mapa de diferenças entre o quadro atual e um quadro de referência, chamado de modelo de plano de fundo. Em um SMT é esperado que as áreas de interesse denotem veículos, como o apresentado na Fig. 1.

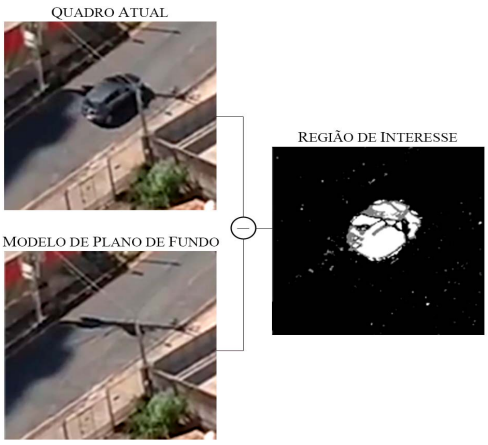


Fig. 1. Detecção de áreas de interesse (veículo) por meio da diferenciação de plano de fundo.

Um modelo de plano de fundo pode ser determinado de vários modos. Destacam-se os trabalhos de Lai & Yung [10], precursores no desenvolvimento de algoritmos para reconhecimento de objeto em vídeos, que utilizam subtração direta entre quadros sequenciais para indicar as Áreas de Interesse, sem necessidade de um modelo de plano de fundo.

A utilização de *Mixture of Gaussians* (MOG) trouxe grandes avanços para a criação de modelos de Plano de Fundo, sendo utilizada por Bouwmans [16] e Zivkovic [17] para produzir resultados mais eficientes que a subtração direta.

A análise probabilística realizada pela MOG permitiu a criação de métodos mais precisos de subtração de plano de fundo, capazes de analisar com precisão fontes de vídeo externas. Apesar disso, assim como na diferenciação de quadros, mudanças bruscas de iluminação, sombras ou a presença de oclusões, i.e., objetos sobrepostos às áreas de interesse, podem ocasionar detecções imprecisas [29], como mostra a Fig. 2 (a), em que o veículo em movimento ao passar pela sombra da árvore (oclusão) não foi detectado pelo MOG, mas sim pela CNN.

Além da mudança nas condições de iluminação, diversas situações afetam negativamente a precisão de técnicas de subtração de plano de fundo, como a presença de quaisquer objetos em movimento que não sejam veículos [9], como ilustrado na Fig. 2 (b), em que o MOG mostra-se muito sensível a outros objetos em movimento; a pouca adaptabilidade a condições climáticas adversas, tais como chuva ou neblina [9], como mostrado na Fig. 2(c). Tais fatores, tornam a subtração de plano de fundo inadequada para a implantação em um sistema de monitoramento de tráfego.

Uma alternativa ao MOG foi proposta recentemente por Bowmans [14] que utilizou Lógica *Fuzzy* para realizar a subtração de plano de fundo, criando modelos com cor, textura e formatos da borda dos objetos presentes nos quadros de vídeo, adaptando-se, assim, mais facilmente às mudanças no cenário.

Após a criação do modelo de plano de fundo, por qualquer dos métodos citados acima, é realizada a detecção das áreas de interesse, por meio da comparação do modelo de plano de fundo com o quadro atual. Por fim, o modelo deve ser atualizado, incorporando nele, por exemplo, a presença de objetos que estão estáticos há um longo período, ou mudanças na iluminação e sombras.

Dentre as técnicas para a diferenciação de quadros, destacam-se a diferenciação de quadro estática e a diferenciação de quadros sequencial.

A diferenciação de quadros estática utiliza uma imagem definida manualmente para representar o plano de fundo, e faz a diferenciação absoluta entre a o quadro atual e a imagem estática. Esse método não é adequado para locais com variação de iluminação ou câmeras que podem se deslocar por forças naturais, como vento.

Como alternativa, sugere-se a diferenciação de quadros sequencial, que gera um modelo de plano de fundo com base no último quadro ou da média dos últimos *n* quadros: dada uma sequência de vídeo **V** com comprimento ***n***, contendo uma sequência de quadros definida por $\mathbf{V} = \{\mathbf{Q_1}, \mathbf{Q_2}, ..., \mathbf{Q_n}\}$, um modelo de plano de fundo **P** pode ser definido por:

$$\mathbf{P} = \frac{\mathbf{1}}{\mathbf{n}} \sum_{\mathbf{t=1}}^{\mathbf{n}} \mathbf{Q_t} \tag{1}$$

Tipicamente, (1) é utilizada para inicializar o modelo de plano de fundo [9]. Após a inicialização do método, o modelo é atualizado através de chamadas recursivas, conforme segue:

$$\mathbf{P_t} = (\mathbf{1} - \propto)\mathbf{Q_{t-1}} + \propto \mathbf{Q_t} \tag{2}$$

onde, $\mathbf{P_t}$ é o modelo de plano de fundo no instante $\mathbf{t} \in \{\mathbf{1}, \mathbf{n}\} \subset \mathbb{N}$ e $\propto \in [\mathbf{0}, \mathbf{1}] \subset \mathbb{R}$ é a taxa de aprendizagem [9].

A maior vantagem da diferenciação de quadros sequencial é manter o modelo de plano de fundo sempre atualizado. No entanto, a inclusão de partes das áreas de interesse no modelo de plano de fundo é inevitável, e torna-se necessário o uso de filtros de adaptação seletiva para que sejam utilizadas somente as áreas onde não existem *blobs* [15]. Além disso, a detecção de objetos de interesse só é perfeita se estes estiverem em constante movimento [9].

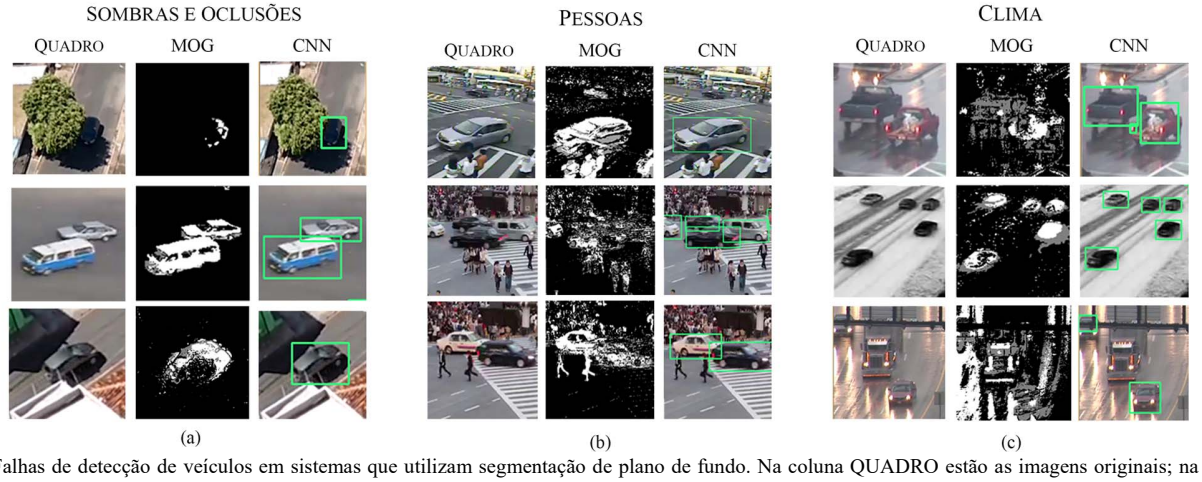


Fig. 2. Falhas de detecção de veículos em sistemas que utilizam segmentação de plano de fundo. Na coluna QUADRO estão as imagens originais; na coluna MOG está o resultado obtido pelo algoritmo de Subtração de Plano de Fundo desenvolvido por Bowmans [16], onde as áreas em cor branca são as Regiões de Interesse (veículos); na coluna CNN estão os resultados obtidos pelo método proposto, onde as Regiões de Interesse estão indicadas em verde.

Diferentemente do método de diferenciação de quadros, as Redes Neurais Convolucionais (*Convolutional Neural Networks – CNNs*) utilizam um mapa de características (*features*) para localizar objetos [24]–[28], neste caso veículos presentes na cena, sem a necessidade de uso de um modelo de plano de fundo, tornando-a robusta nos cenários com oclusões.

A localização de objetos utilizando CNN não busca gerar um contorno preciso do objeto, como os *blobs* em segmentação de plano de fundo, mas sim criar uma caixa delimitadora (*bounding box)* retangular em torno das partes visíveis de um objeto na imagem.

Li, Zhand & Xia [30] utilizaram CNNs a detecção de veículos utilizando LIDAR. Tang et. al [31] conseguiram detectar veículos em imagens aéreas, e Zhen et. Al [32] foram capazes de classificar veículos em tempo real através de CNNs.

B. Rastreamento de Veículos

Métodos de rastreamento são utilizados para estabelecer e mensurar o caminho percorrido por um veículo em um vídeo. Este processo segue os seguintes passos: i) reconhecimento de *features* e a marcação de pontos que serão seguidos, para promover correspondência entre quadros distintos, e ii) ligar as posições de um mesmo ponto do veículo na sequência de quadros, como ilustrado na Fig. 3.

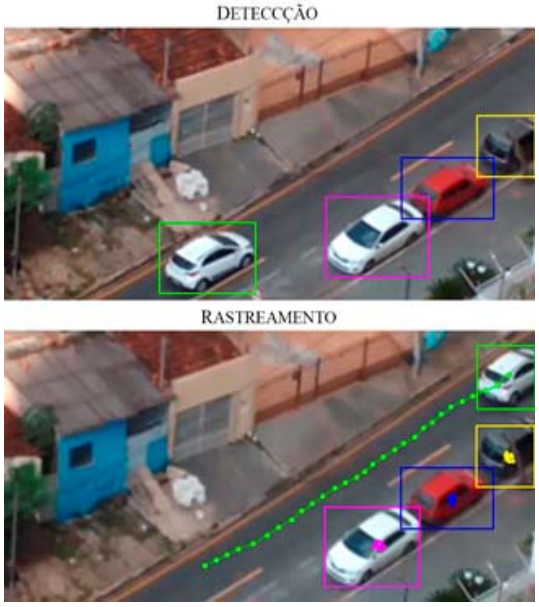


Fig. 3. Detecção e Rastreamento de um carro. De cima para baixo: veículos detectados pela rede neural; resultado esperado do rastreamento.

Para garantir a consistência temporal, é preferível que o intervalo entre os quadros seja constante e curto, e que não haja mudanças bruscas na direção do objeto [29], [33], [34]. A escolha dos pontos (*features*) que serão seguidos é essencial para a precisão do rastreamento.

Para simplesmente traçar o vetor de movimento, a análise por meio de correspondência de grafos é suficiente [34]. Por outro lado, estes modelos não são adequados para ambientes com oclusões, mudanças de cor e de clima, visto que eles necessitam estar continuamente expostos ao objeto, e a presença de oclusões, ainda que parciais, podem afetar o rastreamento.

Um algoritmo de rastreamento muito comum é o *MedianFlow* [35], que seleciona um conjunto de características de um objeto e estima o movimento deste objeto em quadros consecutivos. Entretanto, este algoritmo não é capaz de se recuperar de falhas, causadas por perda da visão do objeto.

Um método mais robusto é o TLD (*Tracking-Learning-Detection*) [36], que une o *MedianFlow* a um arcabouço de aprendizagem de máquina, sendo capaz de detectar o objeto mesmo que este desapareça do campo de visão momentaneamente. Por esta razão, o TLD foi empregado neste trabalho.

C. Medição de Velocidade

Para que seja possível medir precisamente a velocidade de um objeto em movimento, é necessário encontrar sua posição global (em metros) em termos da sua posição na imagem (em *pixels*). Assumindo a utilização de uma câmera *pinhole,* a qual não apresenta distorção à imagem, esta informação pode ser obtida através de uma matriz de homografia – relação que conecta duas imagens a um mesmo plano.

Dado uma matriz de homografia tridimensional H, um ponto da imagem em *pixels* $p_i = (x_i, y_i)$ pode ser mapeado ao ponto $p_w = (x_w, y_w)$ no plano global, por meio de (3) [37].

$$\begin{bmatrix} \boldsymbol{x_w} \\ \boldsymbol{y_w} \\ \boldsymbol{1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{zx_w} \\ \boldsymbol{zy_w} \\ \boldsymbol{z} \end{bmatrix} = \boldsymbol{H} \begin{bmatrix} \boldsymbol{x_i} \\ \boldsymbol{y_i} \\ \boldsymbol{1} \end{bmatrix} \tag{3}$$

A matriz de homografia H é obtida ao se associar quatro *pixels* da imagem a coordenadas globais conhecidas. Em nossos testes, foi utilizada a projeção ortogonal da marcação de divisão de faixas nas extremidades das faixas de rolamento.

A saída de cada par de quadros analisados pelo rastreamento é um conjunto de vetores de movimento, como mostrado em (4)

$$\overrightarrow{\boldsymbol{d_i}} = \boldsymbol{u_i(t)} - \boldsymbol{u_i(t - \Delta t)} \tag{4}$$

onde: $u_i(t)$ é a posição do objeto no quadro atual, $u_i(t - \Delta t)$ é a posição do objeto no quadro anterior, Δt é o intervalo de tempo entre os quadros e i = {1, 2, ..., n} é a sequência de rastreamentos.

A partir do vetor deslocamento e da matriz de homografia H, é possível obter o deslocamento no plano global, denotado por $\overrightarrow{s_i}$, mostrado em (5).

$$\overrightarrow{\boldsymbol{s_i}} = \boldsymbol{H u_i(t)} - \boldsymbol{H u_i(t - \Delta t)} \tag{5}$$

Julgando que o intervalo entre os quadros é constante, que a câmera consegue obter 30 quadros por segundo, $\Delta t = \frac{1}{30}\ \forall\ i$, e que o carro permaneceu em velocidade constante durante a área descrita na matriz homografia H, é possível estimar a

velocidade v_i do veículo no plano global através da equação de movimento retilíneo uniforme, conforme abaixo:

$$\overrightarrow{v_i} = \frac{\overrightarrow{s_i}}{\Delta t} \quad (6)$$

III. Método Proposto

O sistema proposto é composto por cinco elementos principais: uma câmera, que deverá estar fixa de modo que seja possível acompanhar o fluxo de veículos na via; a CNN, que é responsável por localizar veículos no quadro de vídeo; o Rastreador de Objetos, utilizado para estabelecer o percurso percorrido pelo veículo, e assim calcular a velocidade; e por fim um controlador, que gerenciará a execução do sistema. Resumidamente, a metodologia proposta está apresentada na Fig. 4. A seguir, cada um destes blocos é apresentado detalhadamente.

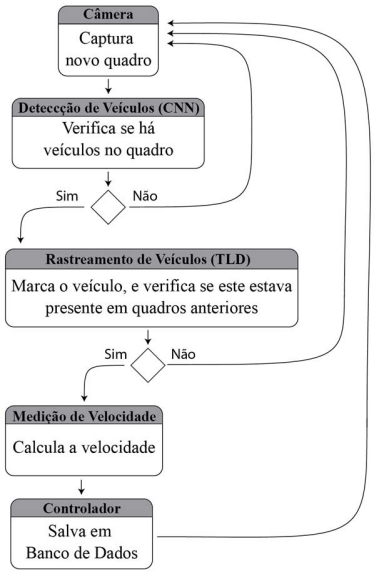


Fig. 4. Diagrama de blocos de funcionamento do sistema proposto.

A. Detecção de Veículos

Após cada novo *frame* ser capturado pela câmera, é iniciado o processo de detecção dos veículos presentes no quadro de vídeo. Tendo em vista melhorar a eficácia das abordagens tradicionais, baseadas no processo de Subtração de Plano de Fundo, este trabalho propõe uma abordagem alternativa, que utiliza uma CNN para otimizar o processo de detecção de objetos. Utilizou-se a CNN baseada na arquitetura *Faster R-CNN* [38].

Arquiteturas baseadas em regiões, como a *Faster R-CNN*, podem ser divididas em dois estágios: o primeiro demarca as Regiões de Interesse (RoI, do inglês *Region of Interest*), produzindo como resultado um par de *pixels* que demarcam as regiões com maior probabilidade de conter um veículo; e o segundo estágio gera uma taxa de confiança sobre cada uma das regiões, i.e., um valor percentual que caso esteja próximo a 0% indica que a RoI provavelmente não contém um veículo, e caso esteja próximo a 100% indica que é muito provável que na RoI esteja presente um veículo.

Cada um dos resultados retornados pela CNN é avaliado, observando-se a taxa de confiança. Caso a taxa de confiança esteja abaixo de um limiar, definido pelo usuário, a RoI é descartada. Caso a taxa de confiança esteja acima do limiar, esta é enviada para o rastreador indicando a posição de um dos veículos na cena. O pseudocódigo é mostrado na Fig. 5.

```
1: function locateVehicles(frame)
2:   out ← FastRCNN.evaluate(frame)
3:   foreach region in out:
4:     if region.score > threshold:
5:       tracker.initialize(region.vertices)
```

Fig. 5. Rotina para Detecção de Veículos.

B. Rastreamento de Veículos

Algoritmos de rastreamento de objetos permitem traçar o trajeto percorrido por um objeto em conjunto de quadros de vídeo. Este bloco, portanto, fornece como resultado a distância percorrida por um veículo assim como a quantidade de quadros que o veículo levou para percorrer tal distância.

Em vias urbanas o cenário é altamente dinâmico, e o algoritmo de rastreamento deve ser robusto o suficiente para calcular o traçado de modo contínuo e estável sob estas condições.

Para este trabalho foi escolhido o algoritmo de rastreamento TLD [39] dado a sua maior precisão em comparação a outros métodos como *MedianFlow, KLT* ou Filtros *Kalman* [39]. O melhor desempenho do algoritmo TLD é decorrente do uso de blocos de aprendizado para alterar as *features* rastreadas [39].

Após a CNN fornecer os vértices das Regiões de Interesse, o algoritmo de rastreamento é inicializado. Os algoritmos de rastreamento permitem um ou mais pontos do objeto. Neste trabalho optou-se por rastrear apenas um ponto, devido à complexidade de se administrar uma quantidade maior de rastreadores paralelamente.

A cada quadro, o rastreador recalcula a posição dos veículos na cena e retorna o trajeto percorrido por cada um deles nos últimos quadros. A execução do bloco de rastreamento é apresentada no formato de pseudocódigo na Fig. 6.

```
1: function trackVehicle(vertices)
2:   tracker ← TLD.initialize(vertices)
3:   vehicle_path ← []
4:   foreach frame in camStream:
5:     tracker.update(frame)
6:     if tracker.current_vertices ≠ null:
7:       vehicle_path.append(tracker.current_vertices)
```

Fig. 6. Rotina de rastreamento de veículos.

C. Medição de Velocidade

Julgando que a câmera esteja estática, é possível analisar a posição de pontos entre os quadros através da homografia. Para isto, ao iniciar o sistema, a área de medição deve ser demarcada pelo usuário. Esta área deve ser retangular, sendo as arestas superior e inferior perpendiculares a via, e as arestas esquerda e direita paralelas à margem da via. Deve ser também informado o comprimento real das arestas laterais

para que o cálculo da distância possa ser feito de maneira precisa. Um modelo da área de medição é mostrado na Fig. 7.

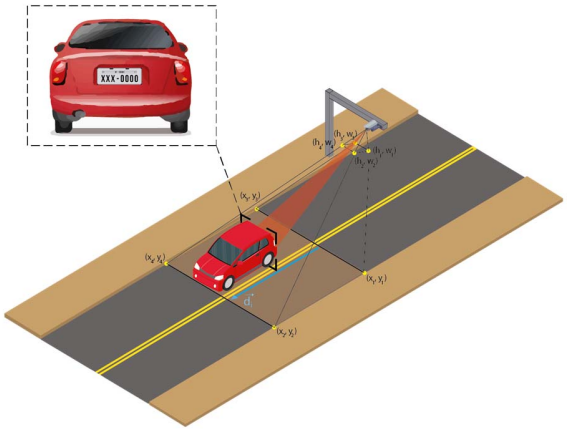


Fig. 7. Protótipo de sistema para reconhecimento de velocidade.

A rotina de cálculo de velocidade verifica, em cada quadro de vídeo, se o corpo do veículo está sobre a área de medição. Para a interseção do veículo com os limites da zona de interesse, utilizou-se o algoritmo *Liang-Barsky* [40], o qual utiliza as equações paramétricas das retas e os pontos de extremidade de um quadrilátero para verificar se ambos se intersectam.

Quando o veículo é primeiro detectado na área de medição, a contagem de quadros se inicia. Ao sair da área de medição, a contagem de quadros é utilizada para o cálculo do tempo utilizado para o veículo percorrer toda a extensão da área de medição. O processo completo de cálculo de velocidade é descrito na forma de pseudocódigo na Fig. 8.

```
1: function getSpeed()
2:  foreach frame in camStream:
3:   if vehicle in frame:
4:    intersect ← Liang-Barsky(vehicle, ground_marking)
5:    if intersect:
6:     frames_in_ground_marking++
7:    else if not intersect and frames_in_ground_marking > 0:
8:     time ← frames_in_ground_marking ÷ framerate
9:     speed ← ground_marking_distance ÷ time
10:    return speed
```

Fig. 8. Rotina para cálculo de velocidade.

IV. Avaliação de Desempenho

Apresenta-se nesta seção as avaliações realizadas com o propósito de aferir o desempenho do modelo proposto, apresentado na Fig. 4. Avalia-se inicialmente a capacidade de a Rede Neural detectar veículos nos quadros de vídeo utilizados, com eficiência aceitável. Em seguida avalia-se o tempo de resposta da CNN utilizada, pois isso pode impactar a viabilidade do método proposto. Por fim, avalia-se a precisão do sistema completo em medir a velocidade dos veículos. Neste segundo cenário de avaliação, o sistema proposto foi comparado com pesquisas já realizadas, de modo a validar a proposta deste trabalho. Ressalta-se que o foco dessas avaliações foram cenários com oclusões, os quais impõem baixo desempenho às propostas existentes.

A. Configuração dos Cenários de Avaliação

O conjunto de dados de validação utilizado para avaliação da localização de veículos é composto por vídeos de vias urbanas, contendo pessoas, oclusões ou mudanças na iluminação.

Além dos vídeos capturados pelos autores, foi utilizado o *dataset* de Luvizon, Nassu & Minetto [8] que amostraram a velocidade dos veículos presentes no quadro de vídeo por meio de sensores piezelétricos e *laser*, respectivamente.

Para os testes foi utilizado um computador com processador Intel Core i7 4770, placa de vídeo nVidia GeForce 1050 Ti e 24 GB de Memória RAM. Os algoritmos utilizados foram implementados na linguagem de programação *Python 3.6.4* [41]. A geração das Figuras foi feita por meio a biblioteca de Visão Computacional *OpenCV 3.4.0* [42]. A rede neural *Faster R-CNN* [38] foi treinada utilizando o *framework TensorFlow* v0.10.0 [43].

B. Precisão da Detecção dos Veículos com CNN

Ainda que não seja possível realizar o rastreamento de objetos utilizando somente a rede neural, a precisão desta é essencial para o ideal funcionamento do método proposto, uma vez que todos os veículos em um quadro de vídeo devem ser reconhecidos e demarcados corretamente, sem falsos positivos, caso contrário o bloco de rastreamento não será iniciado e a detecção de velocidade falhará. Por isso, é importante verificar a eficácia do bloco relativo à detecção de veículos.

Nesta avaliação foi utilizado um conjunto de nove vídeos gravados pelos autores, ou publicados sob licença de uso público, onde há a presença de apenas um veículo por quadro, o que está de acordo com a aplicação deste trabalho. A sequência de vídeos de 1 a 7, mostrados na Tabela I, foram gerados pelos autores. Todos os vídeos utilizados apresentam algum tipo de oclusão, como árvores e sinalizações de trânsito (denominadas oclusões estáticas), chuva ou neblina e/ou presença de pedestres. As características de cada vídeo estão descritas na Tabela I.

TABELA I
DESCRIÇÃO DOS VÍDEOS

Vídeo	Extensão (segundos)	Oclusões estáticas	Pessoas	Chuva ou Neblina
Sequência 1	10	SIM	NÃO	NÃO
Sequência 2	11	SIM	NÃO	NÃO
Sequência 3	10	NÃO	NÃO	SIM
Sequência 4	7	SIM	SIM	NÃO
Sequência 5	4	SIM	NÃO	SIM
Sequência 6	7	SIM	NÃO	SIM
Sequência 7	8	SIM	SIM	NÃO
Sequência 8*	277	SIM	NÃO	NÃO
Sequência 9*	920	SIM	NÃO	SIM

* Sequências com informações da velocidade real do veículo [8]

Para avaliar a precisão média da CNN, foram contabilizados os quadros onde veículos foram corretamente detectados. Quanto maior a precisão, menor é a probabilidade de que um veículo passe pela cena sem ser rastreado. Os resultados da avaliação de precisão da CNN são apresentados na Tabela II.

TABELA II
PRECISÃO DA CNN

Vídeo	Quadros Totais	Quadros Corretamente Detectados	Precisão
Sequência 1	245	222	**91%**
Sequência 2	302	297	**98%**
Sequência 3	183	183	**100%**
Sequência 4	148	122	**82%**
Sequência 5	83	83	**100%**
Sequência 6	161	143	**89%**
Sequência 7	156	156	**100%**
Sequência 8	6918	6700	**97%**
Sequência 9	22979	22447	**98%**
Precisão Média	31175	30353	**98%**

Foi observada uma precisão média de 98% durante a detecção de veículos, mesmo sob condições adversas, o que comprova que o uso de CNN para esta tarefa é adequado e eficiente. Em vídeos curtos e com muitas oclusões, como as Sequências 4 e 6, a precisão obtida ficou abaixo de 90%. Em sequências maiores, como as 8 e 9, o sistema proposto foi capaz de reconhecer os veículos presentes e demarcar sua posição em 97% dos quadros, sob situações que outros sistemas não garantem tal precisão.

C. Tempo de Resposta da CNN

Outra métrica relevante para Sistemas de Monitoramento de Tráfego é o Tempo de Resposta. Tempos de Resposta muito altos tornam o sistema inviável, porque quadros seriam perdidos durante a análise e, assim, comprometeriam a precisão do sistema. Dessa forma, avaliou-se também a capacidade da CNN processar o fluxo de vídeo em tempo hábil no escopo desta proposta, e os resultados são apresentados na Tabela III. Esta Tabela mostra o total de quadros avaliados em cada vídeo, o tempo gasto pela CNN para processar o vídeo e a razão entre esses dois dados.

Os experimentos iniciais apresentaram tempos de resposta muito longos, porque a CNN tinha que processar todas as informações contidas nos vídeos, embora apenas as áreas referentes às vias de trânsito fossem relevantes para o sistema proposto. Por isso, utilizou-se máscaras de recorte, a fim de restringir a avalição a tais áreas de interesse, o que melhorou significativamente os resultados obtidos, os quais são mostrados na Tabela III.

TABELA III
TEMPO DE RESPOSTA DA CNN

Vídeo	Quadros Totais	Tempo Total (s)	Tempo Médio (ms/quadro)
Sequência 1	245	33,2	**136**
Sequência 2	302	40,71	**135**
Sequência 3	183	25,36	**139**
Sequência 4	148	20,11	**136**
Sequência 5	83	11,5	**139**
Sequência 6	161	22,1	**137**
Sequência 7	156	21,98	**141**
Sequência 8	6918	933,25	**135**
Sequência 9	22979	3230,61	**141**
Tempo Médio de Detecção	3464	482	**138**

Os resultados da Tabela III mostram que o sistema proposto requer aproximadamente 140ms para o processamento de cada um dos quadros.

Embora esse tempo possa ser reduzido usando-se *hardware* com maior capacidade de processamento, esses resultados são satisfatórios para a aplicação alvo deste trabalho em que as velocidades dos veículos devem estar próximas às velocidades permitidas nas áreas urbanas, de cerca de 100 km/h.

D. Comparação com a Diferenciação de Quadros

A fim de comprovar a eficácia do sistema proposto, ele foi comparado com o trabalho em [8] que utiliza o método de Diferenciação de Quadros. A Tabela IV mostra a quantidade total de veículos que transitaram durante toda a duração do vídeo, e o percentual de veículos efetivamente detectados por ambos os métodos. Na Tabela V são apresentadas as médias de erro nas velocidades computadas por cada método.

TABELA IV
COMPARATIVO ENTRE O MÉTODO PROPOSTO, BASEADO EM CNN, E O MÉTODO DE DIFERENCIAÇÃO DE QUADROS, PARA DETECÇÃO DE VEÍCULOS

Vídeo	Qtd. de Veículos	Veículos Corretamente Detectados	
		Diferenciação de Quadros [8]	Proposto
Vídeo 8	39	86%	98%
Vídeo 9	189	86%	92%

TABELA V
COMPARATIVO ENTRE O MÉTODO PROPOSTO, BASEADO EM CNN, E O MÉTODO DE DIFERENCIAÇÃO DE QUADROS, PARA CÁLCULO DE VELOCIDADE

Vídeo	Qtd. de Veículos	Taxa Média de Erro das Velocidades	
		Diferenciação de Quadros [8]	Proposto
Vídeo 8	39	**3%**	**5%**
Vídeo 9	189	**3%**	**4%**

Considerando-se a quantidade de veículos detectados e rastreados e um cenário apenas com oclusões estáticas, referente ao vídeo 8, explicado na Tabela I, o método proposto foi superior em 12% em relação ao resultado em [8]. No caso do vídeo 9, referente a um cenário com oclusões estáticas e de baixa visibilidade devido a condições climáticas, conforme Tabela I, o método proposto obteve 6% de vantagem sobre o trabalho em [8].

Esses resultados mostraram que as Redes Neurais Convolucionais são capazes de reconhecer veículos com maior precisão que métodos baseados em Subtração de Plano de Fundo. No entanto, o sistema proposto deve ainda ser capaz de detectar de maneira precisa a velocidade dos veículos localizados na cena. A avaliação do método proposto para cálculo de velocidade está disposta na Tabela V.

As taxas de erro de velocidade encontradas para o método proposto são levemente superiores às obtidas por [8], o que pode ser explicado pela maior quantidade de veículos efetivamente detectados pela proposta deste trabalho. Neste caso, enquanto o método em [8] detectou apenas 86% dos veículos na cena, o método proposto obteve 98%. Se o método de Diferenciação de Quadros conseguir melhorar sua taxa de

detecção de veículos, muito provavelmente aumentará também sua taxa de erro de velocidade computada. Ainda assim, o nível de erro do método proposto encontra-se dentro do limiar aceitável para aplicação em sistemas comerciais.

V. CONCLUSÕES

Este trabalho apresentou uma proposta de sistema de monitoramento de tráfego baseado em Redes Neurais Convolucionais como mecanismo de detecção de veículos em sequências de vídeos.

O método proposto foi eficiente para as situações avaliadas, conseguindo substituir completamente a Subtração de Plano de Fundo para reconhecimento de veículos. Os resultados obtidos mostraram que o método proposto é capaz de medir velocidade de veículos com precisão de até 95%, no pior caso, o que está próximo ao obtido por métodos existentes baseados em Subtração de Plano de Fundo; e ainda apresenta a grande vantagem de ser até 12% mais eficaz na detecção de veículos em cenários com oclusões.

Portanto, o método proposto mostrou-se adequado para aplicações em áreas urbanas de baixa circulação, onde pode haver muitas oclusões e os veículos transitam em velocidades moderadas.

Como trabalho futuro, sugere-se que: i) o tempo de resposta da CNN seja aprimorado, visando cenários de maiores velocidades; ii) o algoritmo de rastreamento seja otimizado, de modo a permitir que mais de um veículo seja rastreado na cena, o que pode ser necessário em ambientes de velocidades altas; iii) a técnica de demarcação de diversos pontos de rastreamento no veículo, em [8], seja aplicada no sistema aqui proposto.

AGRADECIMENTOS

Este projeto foi financiado parcialmente pelo Instituto Federal de Mato Grosso (Editais PROPES) e pelas agências de fomento CNPq e FAPEMAT.

REFERÊNCIAS

[1] Organiação Mundial da Saúde (OMS), "Relatório Global Sobre O Estado Da Segurança Viária em 2015," *Relatório Glob. Sobre O Estado Da Segurança Viária 2015*, vol. 1, p. 16, 2015.

[2] B. Coifman, D. Beymer, P. Mclauchlan, and J. Malik, "A real-time computer vision system for vehicle tracking and surveillance," vol. 6, pp. 271–288, 1998.

[3] Jian Xing, "Evaluation of Roadside Wrong-Way Warning Systems with Different Types of Sensors," *J. Traffic Transp. Eng.*, vol. 4, no. 3, pp. 155–166, 2016.

[4] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Computer*, vol. 29, no. 3. pp. 31–44, 1996.

[5] R. Waregaonkar and A. R. P. M. B, "Development of Prototype for Vehicle Speed Measurement," 2017.

[6] A. Yabo, S. Arroyo, F. Safar, and D. Oliva, "Vehicle Classification and Speed Estimation using Computer Vision Techniques," 2016.

[7] J. Lan, J. Li, G. Hu, B. Ran, and L. Wang, "Vehicle speed measurement based on gray constraint optical flow algorithm," *Optik (Stuttg).*, vol. 125, no. 1, pp. 289–295, 2014.

[8] D. C. Luvizon, B. T. Nassu, and R. Minetto, "A Video-Based System for Vehicle Speed Measurement in Urban Roadways," vol. X, no. X, pp. 1–12, 2016.

[9] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Underst.*, vol. 122, pp. 4–21, 2014.

[10] A. H. S. Lai and N. H. C. Yung, "Fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 4. pp. 241–244, 1998.

[11] P. Kaewtrakulpong and R. Bowden, "An Improved Adaptive Background Mixture Model for Real- time Tracking with Shadow Detection," *Adv. Video Based Surveill. Syst.*, pp. 1–5, 2001.

[12] F. Kristensen, P. Nilsson, and V. Öwall, "Background segmentation beyond RGB," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3852 LNCS, pp. 602–612, 2006.

[13] M. M. Azab, H. A. Shedeed, and A. S. Hussein, "A new technique for background modeling and subtraction for motion detection in real-time videos," *Proc. - Int. Conf. Image Process. ICIP*, pp. 3453–3456, 2010.

[14] T. Bouwmans, "Background Subtraction for Visual Surveillance: A Fuzzy Approach," *Handb. Soft Comput. Video Surveill.*, pp. 103–138, 2012.

[15] M. H. Sigari, N. Mozayani, and H. R. Pourreza, "Fuzzy Running Average and Fuzzy Background Subtraction: Concepts and Application," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 8, no. 2, pp. 138–143, 2008.

[16] T. Bouwmans, "Recent Advanced Statistical Background Modeling for Foreground Detection - A Systematic Survey," 2015.

[17] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," *Proc. 17th Int. Conf. Pattern Recognit.*, vol. 2, no. 2, pp. 28-31 Vol.2, 2004.

[18] Z. Zivkovic and F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, 2006.

[19] S. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," p. 881, 2004.

[20] Z. Tang, Z. Miao, and Y. Wan, "Background Subtraction Using Running Gaussian Average and Frame Difference," *Ifip Int. Fed. Inf. Process.*, pp. 411–414, 2007.

[21] O. Tuzel, F. Porikli, and P. Meer, "A Bayesian Approach to Background Modeling," *2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. - Work.*, vol. 3, pp. 58–58, 2005.

[22] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, "Neural Network Approach to Background Modeling for Video Object Segmentation," *IEEE Trans. Neural Networks*, vol. 18, no. 6, pp. 1614–1627, 2007.

[23] S. K. Choudhury, P. K. Sa, S. Bakshi, and B. Majhi, "An Evaluation of Background Subtraction for Object Detection Vis-a-Vis Mitigating Challenging Scenarios," *IEEE Access*, vol. 4, no. c, pp. 6133–6150, 2016.

[24] J. Kim, "Deep Learning for Computer Vision," *Comput. Vis. Pattern Recognit. Work.*, pp. 1–85, 2014.

[25] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Comput. Intell. Neurosci.*, vol. 2018, p. 13, 2018.

[26] J. Gu *et al.*, "Recent Advances in Convolutional Neural Networks," pp. 1–38, 2015.

[27] Goodfellow, Ian, Y. Bengio, and A. Courville, "Deep Learning," *MIT Press*, 2016.

[28] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object Recognition with Gradient-Based Learning," no. 0, pp. 319–345, 1999.

[29] N. K. Kanhere, R. Hall, S. T. Birchfield, R. Hall, and W. A. Sarasua, "Vehicle Segmentation and Tracking in the Presence of Occlusions," no. 864, pp. 1–18.

[30] B. Li, T. Zhang, and T. Xia, "Vehicle Detection from 3D Lidar Using Fully Convolutional Network," 2016.

[31] T. Tang, S. Zhou, Z. Deng, H. Zou, and L. Lei, "Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining," *Sensors (Switzerland)*, vol. 17, no. 2, 2017.

[32] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle Type Classification Using a Semisupervised Convolutional Neural Network," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2247–2256, 2015.

[33] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *Proc. 1999 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Cat No PR00149*, vol. 2, no. c, pp. 246–252, 1999.

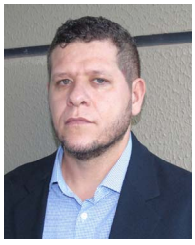
[34] M. Fiaz, A. Mahmood, and S. K. Jung, "Tracking Noisy Targets: A Review of Recent Object Tracking Approaches," 2018.

[35] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," *Proc. - Int. Conf. Pattern Recognit.*, pp. 2756–2759, 2010.

[36] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," vol. 6, no. 1, pp. 1–14, 2010.
[37] G. Wang, Z. Hu, F. Wu, and H. T. Tsui, "Single view metrology from scene constraints," *Image Vis. Comput.*, vol. 23, no. 9, pp. 831–840, 2005.
[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.
[39] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, 2012.
[40] Y.-D. Liang and B. a. Barsky, "An analysis and algorithm for polygon clipping," *Commun. ACM*, vol. 26, no. 11, pp. 868–877, 1983.
[41] Python Software Foundation, "Python 3.6.4." 2018.
[42] OpenCV Team, "OpenCV 3.4.0." 2018.
[43] Google Inc., "TensorFlow." 2018.

Vitor Bruno de Oliveira Barth é aluno do curso de Engenharia da Computação do Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso (IFMT). Realiza pesquisas nas áreas de Inteligência Artifical, Visão Computacional e *Smart Grids*.

Ruy de Oliveira possui graduação em Engenharia Elétrica pela Universidade Federal de Itajubá (1992), mestrado em Engenharia Elétrica pela Universidade Federal de Uberlândia (2001) e doutorado em Redes de Computadores e Sistemas Distribuídos pela Universidade de Berne, Suiça (2005). Atualmente é professor titular do Instituto Federal de Mato Grosso (IFMT). Tem experiência na área de Engenharia Elétrica, com ênfase em sistemas de comunicação de dados e automação de processos. Desenvolve pesquisas nos seguintes temas: redes de comunicação de dados, redes elétricas inteligentes, segurança da informação e aprendizagem de máquinas.



Mário Anderson de Oliveira possui graduação em Engenharia Elétrica pela Universidade Federal de Goiás (2004). Mestre em Engenharia Elétrica pela Universidade Federal de Santa Catarina (2007). Doutor em Engenharia Elétrica pela UNESP de Ilha Solteira (2013). Em 2014-2015 desenvolveu estágio de Pós Doutorado na *University of Michigan* (*Aeroespace Department*) focando na Análise da Integridade Estrutural (*Structural Health Monitoring* - SHM). Desde 2009 é professor efetivo do Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso (IFMT) e possui experiência na área de Engenharia Elétrica, com ênfase em eletrônica e instrumentação, atuando principalmente nos seguintes temas: processamento de sinais, aquisição de dados, instrumentação, aplicações de transdutores PZT, sistemas SHM, sistemas inteligentes e redes neurais artificiais.

Valtemir Emerêncio do Nascimento possui graduação em Engenharia Elétrica pela Universidade Federal de Mato Grosso do Sul (1999), mestrado pela EESC - Universidade de São Paulo (2002) e doutorado pela EESC - Universidade de São Paulo (2007). Atualmente é professor no Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso. Possui experiência em redes de sensores, métodos numéricos aplicados a propagação de onda em comunicações e dispositivos, possui interesse em computação de alto desempenho e inteligência artificial.