Anomalies Identification in Images from Security Video Cameras Using Mask R-CNN

G. Minari, F. Silva, D. Pereira, L. Almeida, M. Pazoti, A. Artero, and V. de Albuquerque

***Abstract*—In this work we developed a system to identify anomalies in images from video security cameras in an urban environment. Initially people are detected in the images using Mask R-CNN. From the binary mask are extracted characteristics of the people so that the anomalies can be detected. In order to facial recognition we used Facial Landmarks so that the system knows the residents and authorized people avoiding the false anomalies. We considered four anomalies in this work: the act of jumping a wall, standing for a long time in front of the residence, walking thru the sidewalk several times and entering a place without permission.**

***Index Terms*—Mask R-CNN, CNN, HOG, People characteristics extraction, Intrusion detection, Facial recognition.**

I. Introdução

NOS últimos anos, o índice de delitos em locais urbanos teve um crescimento considerável, no período de 2017 foi obtido o maior número de furtos registrado (515.595), aumento de 1% em relação a 2016 e 4% em relação a 2015. Um aumento de 60% no número de presos pelo crime de roubos e furtos a residências foi registrado 2018, tendo 6.452 ocorrências de furtos e roubos somente no primeiro semestre de 2018 [1].

O investimento em segurança residencial vem aumentando, os sistemas de segurança eletrônica estão sendo cada vez mais utilizados. Empresas de monitoramento são dependentes do trabalho manual e de sistemas de segurança secundários. Normalmente quando um sistema de segurança secundário é acionado, a empresa necessita que um funcionário verifique e analise as imagens para identificar se não é uma falsa anomalia [2]. São consideradas anomalias o ato de pular um muro, ficar muito tempo parado na frente da residência, passar várias vezes em frente à residência e adentrar a um local sem permissão. Vale ressaltar que equipamentos de segurança eletrônica estão propícios a falhas.

G. H. Minari, Universidade do Oeste Paulista (Unoeste), Presidente Prudente, São Paulo, Brasil (gustavo_minari@hotmail.com.br).

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

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

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

M. A. Pazoti, Universidade do Oeste Paulista (Unoeste), Presidente Prudente, São Paulo, Brasil (mario@uoneste.br).

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

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

Poucas câmeras são utilizadas para segurança, grande parte delas é utilizada para monitoramento, sendo assim necessária uma pessoa para analisar as imagens e encontrar anomalias. Todo esse processo depende da observação de um ser humano, no qual possui certo grau de subjetividade, podendo influenciar na ineficiência do resultado [2].

Existem muitas tecnologias utilizadas na área de segurança, mas não tem seu aproveitamento total devido à falta de recursos computacionais, como sistemas inteligentes que identificam anomalias e de fato vigiam a residência. Projetos como um sistema de vigilância com detecção de intrusão utilizando inteligência artificial [3] e um sistema de detecção de fumaça e fogo [4] demonstram a utilização de detecção através de processamento digital de imagem. Uma grande quantidade de pesquisas nessa área vem sendo realizadas nos últimos anos, mas ainda existe uma grande lacuna para estudos e pesquisas científicas.

A tarefa de detectar pessoas e a análise de seu movimento objetiva além da segurança, o rastreamento visual, a contagem automática de pessoas entre outros. A detecção e o rastreamento de pessoas em sequências de imagens são de grande utilidade para várias tarefas desempenhadas pela sociedade, como monitoramento de espaços públicos, estações de ônibus, estádios de futebol e até mesmo para análise de comportamento humano [5].

Este trabalho vem contribuir com uma solução computacional que, a partir das imagens obtidas por meio de câmeras de monitoramento residencial, realize a detecção de anomalias e evite identificar falsas anomalias com os próprios moradores da residência, fazendo identificação por reconhecimento facial.

II. Conceitos Fundamentais

Nesta seção é apresentada a fundamentação teórica sobre os métodos utilizados para o desenvolvimento deste trabalho.

A. Redes Neurais Convolucionais

Uma rede neural convolucional (*Convolutional Neural Network* – CNN) consiste em uma rede neural de multicamadas, sem a necessidade de um processamento inicial, reduzindo o pré-processamento ao mínimo, assim sua velocidade de execução se torna maior. Dito isto, esta diferença na arquitetura da CNN causa uma facilidade para reconhecimento em situações de extrema variação como exemplificado por LeCun *et al*. (2015) [6]. As redes neurais convolucionais conseguem reconhecer padrões extremamente complexos e são adaptáveis a distorções e

transformações geométricas. É exatamente essa independência de um conhecimento específico e do esforço humano no desenvolvimento de suas funcionalidades básicas a maior vantagem de sua aplicação [7][8][9][10][11].

B. *Mask R-CNN*

O método Mask R-CNN proposto por He *et al*. em 2018 [12] apresenta um conceito simples e flexível baseada no método Faster R-CNN [13] para detectar objetos e interpolação bilinear [14] para criar a máscara dos objetos.

Mask R-CNN trabalha em dois estágios, sendo o primeiro de detectar redes de proposta de região (RPN – *Region Proposal Network*). O segundo estágio realiza três atividades em paralelo: previsão de possível classe; definição da região de interesse do objeto (RoI – *Region of Interest*) e criação de uma máscara binária para o objeto (Fig. 1). Com o uso da máscara binária e RoI pode ser determinada a classe do objeto. A saída da CNN é a probabilidade de a imagem de entrada pertencer a uma das classes para qual a rede foi treinada [15].

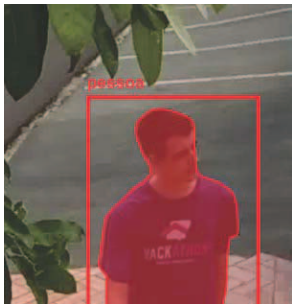
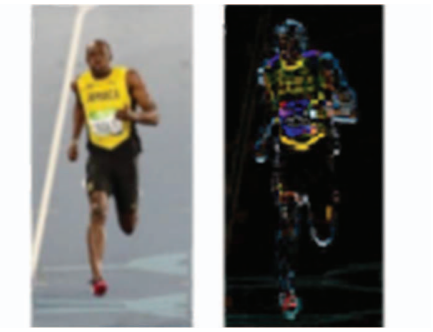


Fig. 1. Demonstração resultado Mask R-CNN.

C. *HOG* (*Histogram of Oriented Gradient*)

HOG é um descritor de características que analisa a forma do objeto (forma e textura) [16][17], em que são criadas duas matrizes, a de direção e a de intensidade. A matriz de intensidade é preenchida de acordo com a variação das cores entre os pixels. Quando não se tem variação de cor é definida a cor preta e onde se tem variação de cor é definida a cor do pixel, como é mostrado na Fig. 2. Um histograma (Fig. 3) é criado na busca da detecção do rosto de uma pessoa.



(a) (b)

Fig. 2. (a) Representação da imagem original. (b) Representação da imagem gerada utilizando HOG [16].

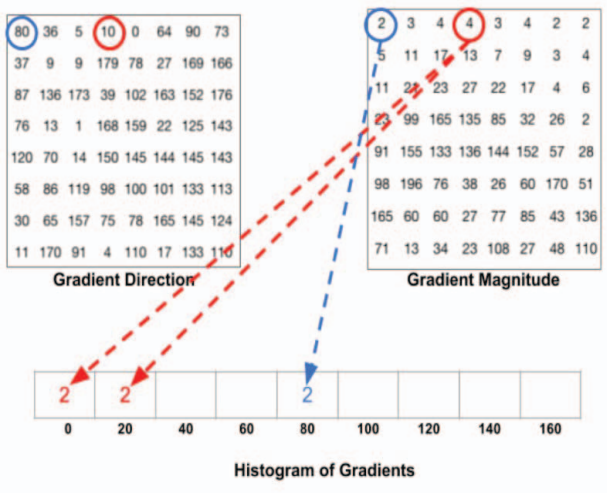


Fig. 3. Representação da matriz de intensidade de direção, e o histograma gerado [16].

D. *Detector de Pontos Faciais para Reconhecimento Facial*

O algoritmo de detecção dos pontos faciais [18] utiliza uma SVM para identificar pontos do rosto de uma pessoa, necessários no reconhecimento facial. SVM (*Support Vector Machine*) é um algoritmo de classificação e regressão, as suas classificações são definidas de acordo com os valores treinados [19].

Seu funcionamento pode ser entendido da seguinte maneira: dada *N* classes e um conjunto de pontos que pertencem a essas *N* classes, o SVM determina o hiperplano que separa os pontos [20]. Os pontos são separados de maneira que a mesma classe tenha seus pontos o mais próximo possível e que tenha a maior distância possível entre as classes e o hiperplano.

No reconhecimento de face são utilizados 68 pontos do rosto de uma pessoa, entre eles: queixo, lábio externo, lábio interno, sobrancelhas, olhos e nariz (Fig. 4).

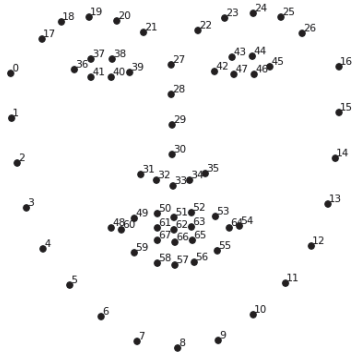


Fig. 4. Representação de pontos detectados em uma face [18].

E. *Definição de Características*

O Algoritmo de definição de características utiliza de uma imagem no espaço de cor HSV [21], desconsiderando a iluminação existente no local (Fig. 5). Somente o valor de H (matiz) é utilizado para essa definição, em que ele representa a tonalidade da cor (Fig. 6).

(a) (b)

Fig. 5. Mesma imagem nos espaços de cor RGB e HSV. (a) RGB. (b) HSV.

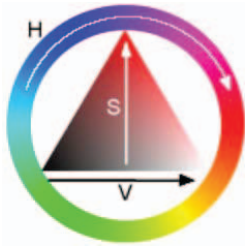


Fig. 6. Representação do padrão HSV [21].

Os valores de H são normalizados por (1), sendo aplicado o valor que mais se repete na região da camiseta. É considerada a área maior que 21% e menor que 50% da atura total para definir como característica para rastreio de uma pessoa a cor predominante da camiseta.

$$H_{xy} = \frac{H_{xy}}{35} \tag{1}$$

F. Captura do Vídeo da Câmera de Segurança

Foi utilizada a biblioteca FFmpeg [22] para fazer a obtenção dos quadros (*frames*) de uma câmera de monitoramento. Com esse recurso foi possível converter o formato de vídeo da câmera para quadros de imagem no formato desejado como saída (RGB 24). Também foi possível realizar a limitação da quantidade de quadros de maneira que não tenha atraso, o que poderia dificultar a detecção de anomalias em tempo real. A limitação de quadros por segundo foi realizada por (2), em que *ms* representa o tempo máximo no qual o processamento leva para ser efetuado.

$$QPS_{max} = \frac{1000}{ms} \tag{2}$$

G. Detecção de Anomalias

As anomalias são detectadas através da utilização de técnicas de processamento digital de imagem, dente as técnicas está a análise de posição da pessoa no ambiente, comparação entre quadros atual e anteriores e a contagem de quadros.

Na detecção da anomalia, na qual uma pessoa fica um grande tempo em frente a residência, é calculada a diferença DQ da contagem de quadros CQ atual (CQ_{final}) e a CQ em que a pessoa apareceu pela primeira vez ($CQ_{inicial}$) usando (3). Quando a diferença DQ maior que X é considerada uma anomalia. X é definido por (4) que calcula o número de quadros por minuto.

$$DQ = CQ_{final} - CQ_{inicial} \tag{3}$$

$$X(m) = QPS_{max} * 60 * m \tag{4}$$

Quando uma pessoa é detectava pelas câmeras, é analisada a diferença entre a contagem de quadros CQ atual e a CQ em que a pessoa apareceu pela última vez (CQ_{final}) usando (5).

$$DQ = CQ_{atual} - CQ_{final} \tag{5}$$

Quando essa diferença é menor que X, a contagem de quadros visto pela última vez (CQ_{final}) recebe o valor da contagem de quadros atual (CQ_{atual}), e caso a diferença seja maior que X, a contagem de quadros inicial ($CQ_{inicial}$) também recebe o valor da contagem de quadros atual (CQ_{atual}), além de acrescentar um ao contador de vezes em que uma pessoa passou em frente a uma residência.

Entretanto, existe uma checagem de X quadros em (3) para considerar que a pessoa está passando depois em um longo período em frente à residência, sendo assim o contador tem o seu valor zerado.

A detecção de anomalias, que considera quantas vezes a pessoa passou em frente à residência, analisa o contador que é atribuído pela verificação de tempo em frente à residência. Quando esse contador é maior que o desejado, é considerado então uma anomalia.

A tentativa de pular o muro é detectada por meio de duas análises distintas. Para isso é realizada uma comparação entre o quadro anterior e o quadro atual, sendo analisada a posição em que a pessoa se encontra em (6) e o tamanho da área dessa pessoa em (7).

$$P(x,y) = \left(\frac{X_{max} + X_{min}}{2}, Y_{\max}\right) \tag{6}$$

$$A = (X_{max} - X_{min}) * (Y_{max} - Y_{min}) \tag{7}$$

em que a área da pessoa é obtida por meio dos valores mínimo (X_{min}, Y_{min}) e máximo (X_{max}, Y_{max}) de cada eixo.

Quando a posição da pessoa no eixo Y se altera, mas a sua área não sofre alteração, é considerado uma anomalia representando que a pessoa está pulando. A outra análise é feita através do tamanho da área da pessoa, que quando os valores do eixo X e eixo Y não sofrem alteração, mas sua área é maior, é considerado que a pessoa está mais próxima da câmera, então ela está pulando, sendo assim ocorre uma anomalia.

A última anomalia a ser analisada é a de acesso a uma área sem permissão. Quando uma pessoa adentra uma área que foi definida como restrita previamente (Fig. 7), é considerado uma anomalia. Para isso é comparada se a posição em que a pessoa se encontra pertence a uma dessas áreas calculadas em (8), (9) e (10). É levado em consideração o maior valor do eixo Y representando o pé da pessoa, pois como a imagem é um

ambiente 2D não existe uma noção precisa da localização da pessoa em profundidade.

$$F1(Y_{max}, A) = A_{yi} \leq Y_{max} \leq A_{yf} \quad (8)$$

$$F2(X, A) = A_{xi} \leq X_{min} \leq A_{xf} \quad (9)$$

$$F3(X, A) = A_{xi} \leq X_{max} \leq A_{xf} \quad (10)$$

Quando o resultado da função F1 em (8) é verdadeiro e umas das duas funções F2 em (9) ou F3 em (10) também é, considera-se que a pessoa adentrou a uma área sem permissão, representando uma anomalia.

Fig. 7. Representação de uma demarcação de área restrita.

III. Método Proposto

Nesta seção é descrito o funcionamento do método proposto, sendo dividido em cinco etapas: criação do conjunto de dados de teste, modelo da CNN utilizado, funcionamento do algoritmo e avaliação dos resultados (Fig. 8).

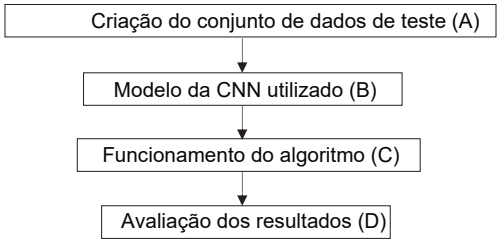


Fig. 8. Fluxograma representando as etapas do método proposto.

A. Conjunto de Dados de Teste

O conjunto de teste utilizado neste trabalho é constituído por imagens que foram obtidas a partir de uma câmera de 12 megapixels Full HD (1920x1080). Essas imagens foram capturadas de quatro pontos de vista diferentes, para que assim tenha diferentes ângulos, posicionamentos e iluminação para a detecção de anomalias.

Alguns exemplos dos posicionamentos das câmeras utilizadas na captura são mostrados na Fig. 9. Todas as filmagens foram realizadas na parte da frente de uma residência. Uma das câmeras foi posicionada na frente da residência focalizando o portão de acesso, para que seja possível efetuar o reconhecimento facial.

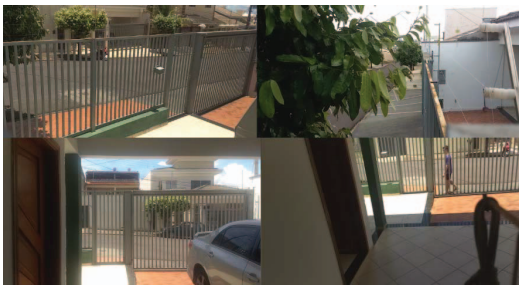


Fig. 9. Imagens obtidas com a câmera utilizada para gravação dos vídeos demonstrando os diferentes pontos de vista de captura.

B. Modelo da CNN Utilizado

Existem diversos modelos de redes neurais convolucionais [9][10][12][13], esses modelos possuem como principal objetivo a classificação, detecção ou segmentação de objetos. Todos os algoritmos escolhidos neste trabalho levaram em consideração o tempo de resposta e sua capacidade de classificação [23]. Foi escolhida a Mask R-CNN utilizando a biblioteca TensorFlow [24]. A biblioteca TensorFlow foi desenvolvida pelo Google Brain Team com o intuito de facilitar o desenvolvimento de aplicações de alto desempenho (CPUs, GPUs). Ela oferece um vasto suporte para aprendizado de máquina (*Machine Learning*) e aprendizado profundo (*Deep Learning*) [25].

Neste projeto não foi realizado o treinamento da R-CNN, para isso foi utilizado um treinamento disponível que contêm 80 objetos reconhecíveis, e treinada com mais de 330 mil imagens (cocodataset). Sua execução ficou limitada devido ao ambiente utilizado no desenvolvimento, a configuração da R-CNN foi: 1 GPU, Resolução máxima: 1024x1024, 1000 passos por época, total de 50 épocas, 100 instâncias por detecção e mínimo de acurácia igual a 0,7.

C. Funcionamento do Algoritmo

Este trabalho é constituído em duas etapas, a da captura e limitação de quadros por segundo (QPS), e a de tratamento dos quadros capturados para reconhecimento de anomalias e de moradores. O fluxograma de funcionamento do algoritmo desenvolvido é mostrado na Fig. 10.

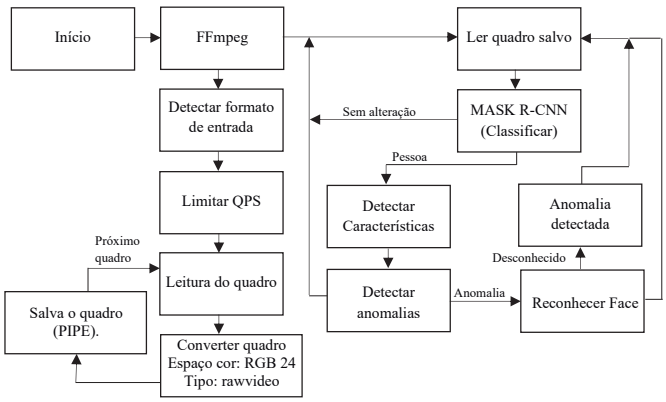


Fig. 10. Fluxograma de funcionamento do algoritmo.

A primeira etapa, responsável pela captura e limitação dos quadros, faz uso da biblioteca FFmpeg [22] para obter os

quadros no formato desejado e limitar a quantidade de quadros por segundo. Quando um quadro é capturado, os recursos do FFmpeg efetuam a conversão do quadro para o formato de imagem com espaço de cor RGB 24, após a conclusão, é aguardado o próximo quadro para efetuar novamente a primeira etapa.

A segunda etapa é responsável pelo tratamento do quadro e da detecção das anomalias. Quando um novo quadro é capturado, primeiramente é feito um redimensionamento deste quadro para a resolução de 800x450 pixels.

Esse quadro redimensionado é analisado pela Mask R-CNN, retornando os objetos detectados. Quando uma pessoa é detectada, suas características são processadas pelo algoritmo de detecção de características, desenvolvido neste trabalho. As características obtidas são utilizadas para fazer o rastreio da pessoa, sendo usada para fazer um histórico sobre aquela pessoa.

Com as características de cada pessoa, é possível fazer a análise de anomalias a partir do histórico das ações daquela pessoa, como: posições anteriores, contagem de quadro inicial (primeira vez que foi detectado) e contagem de quadro final (última vez que foi detectado por alguma câmera). Com as informações obtidas, o algoritmo de detecção de anomalia inicia a procura de anomalias. Quando uma anomalia é detectada, se faz necessária a identificação de moradores para evitar falsas anomalias, utilizando para isso o reconhecimento facial.

O fluxograma do reconhecimento dos moradores é apresentado pela Fig. 11. Inicialmente é executado o algoritmo de HOG [16] para fazer a detecção de uma face. Quando uma face é detectada, o algoritmo de pontos faciais faz o reconhecimento da pessoa. Caso a pessoa não seja reconhecida, o algoritmo utiliza o quadro anterior para tentar novamente o reconhecimento da pessoa. Após quarenta vezes sem sucesso no reconhecimento da face, a pessoa é considerada desconhecida, e a ação dela é considerada uma anomalia.

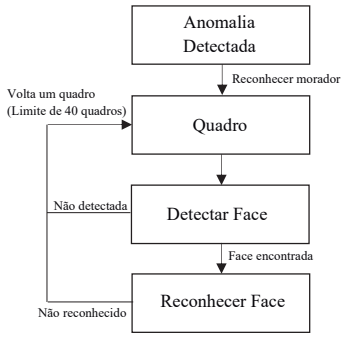


Fig. 11. Fluxo de funcionamento do reconhecimento de morador.

Após o reconhecimento de uma anomalia ou de um morador, é tomado novamente o processo de análise das imagens da câmera de segurança, uma vez que o algoritmo está em repetição constante, desde que exista a conexão entre o algoritmo e a câmera de segurança por meio do FFpmeg (primeira etapa).

D. Avaliação dos Resultados

Para a avaliação dos resultados foi levado em consideração apenas o desempenho de detecção dos algoritmos desenvolvidos neste trabalho. Por se tratar de um trabalho com foco em tempo real, um fator importante para o sucesso foi o tempo de execução.

Uma avaliação parcial durante o desenvolvimento do algoritmo foi feita, esta avaliação consistiu em testar os algoritmos de detecção de anomalias separadamente, assim sendo possível identificar diversas falhas existente em cada detecção individual.

O maior problema detectado foi o tratamento do ambiente ser apenas em 2D. Utilizando imagens individualmente de cada câmera, não foi possível aproveitar os diversos ângulos para simular um ambiente 3D.

Quando testado individualmente, o sistema de detecção da anomalia 'pular o muro' apresentou falha nas imagens das câmeras com posicionamento com ângulo frontal, voltadas para a frente da casa. Caso alguém pulasse o muro na residência do outro lado da rua, quando bem enquadrado e identificado, poderia dar alerta de anomalia.

Outra falha detectada foi a possibilidade de crianças brincando na rua ser considerada duas anomalias. A primeira seria 'passar diversas vezes em frente à residência', e a segunda 'ficar muito tempo parado em frente à residência'. A detecção dessas atividades das crianças poderia gerar muitos avisos de anomalias detectadas.

Para realizar a avaliação final das anomalias foram utilizados 1.680 quadros, sendo 420 quadros para cada local de posicionamento da câmera definido na residência, e 610 quadros contendo o rosto de uma pessoa.

O método proposto atingiu resultados satisfatórios no processo de reconhecimento das anomalias, reconhecendo todas as anomalias que foram simuladas, emitindo um baixo número de falsas anomalias por se tratar de um ambiente controlado. Em contrapartida, o resultado em reconhecer o morador em uma das câmeras que foi posicionada para fazer o reconhecimento facial não foi satisfatório, pois não foi reconhecido o rosto do morador como pode ser visualizado na Tabela I. Foi observado que quando o ângulo da câmera é superior a 16° não foi possível fazer o reconhecimento facial, e somente foi possível detectar a face da pessoa quando ela olhou para a câmera.

TABELA I
RESULTADO RECONHECIMENTO MORADOR

Ângulo	Total de quadros	Faces encontradas	Faces não encontradas	Faces reconhecidas
< 16º	360	253	107	218
≥ 16º	250	3	247	0

Foi realizada a execução do algoritmo desenvolvido neste trabalho em três computadores com configurações diferentes, com intuito de ter parâmetros de referência para o computador adequado para trabalhar em tempo real. Os tempos de processamento (em milissegundos – *ms*) do algoritmo podem ser visualizados na Tabela II. Com os tempos resultantes,

conclui-se que se faz necessária a existência de uma placa gráfica (GPU) para a execução em tempo real.

TABELA II
TEMPO DE RESPOSTA DO ALGORITMO

GPU	Tempo de processamento (*ms*)	Limitação do QPS
sem GPU	11.000	1
MX 940M	3.456	1
GTX 1060	285	3

IV. CONCLUSÕES

Os resultados deste trabalho mostram que a metodologia desenvolvida pode ser aplicada em cenários que exigem uma resposta rápida, desde que seja utilizado um computador com uma placa gráfica (GPU) de alto desempenho, como uma Nvidia GTX 1060, onde seu tempo de médio foi de 285 *ms*, possibilitando a execução deste algoritmo em tempo real.

Acredita-se que a precisão do reconhecimento facial pode ser melhorada com a utilização de câmeras com melhor qualidade de imagem com o valor de DPI (*Dots Per Inch*) maior. Vale ressaltar que todos os sistemas de segurança apenas podem inibir ou dificultar uma anomalia, sendo assim o sistema que tenha maior precisão e menor tempo de resposta pode possibilitar uma maior segurança.

Este trabalho também pode ser utilizado para outras áreas além da segurança contra delitos. Por se tratar um algoritmo de fácil adaptação é possível também usar para segurança de crianças na residência, delimitando, por exemplo, uma piscina como área proibida. Sendo assim, uma anomalia seria detectada quando alguma criança (pessoa sem permissão) acesse o local delimitado.

Como trabalhos futuros, melhorias podem ser realizadas, como a criação de um algoritmo que tenha a capacidade de trabalhar com mais de uma câmera ao mesmo tempo, criando assim uma rede entre as câmeras, possibilitando o rastreio das pessoas em todas as câmeras. Poderá ser utilizada uma CNN para detecção das anomalias, com o foco em diminuir as falsas anomalias. E a utilização de um algoritmo de reconhecimento facial que tenha a capacidade de funcionar em um ângulo superior a 16° mantendo uma alta taxa de reconhecimento.

REFERÊNCIAS

[1] "SSP - Secretaria da Segurança Pública", 2018. [Online]. Available: http://www.ssp.sp.gov.br/

[2] "Abese - Associação Brasileira das Empresas de Sistemas Eletrônicos de Segurança", 2018. [Online]. Available: https://abese.org.br/index.php/412-pesquisa-abese-mapeia-mercado-de-seguranca-eletronica/

[3] T. Jamundá, "Um sistema de vigilância com detecção de intrusão utilizando inteligência artificial", M.S. thesis, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, 2002.

[4] T. Chen, Y. Yin, S. Huang, Y. Ye, "The smoke detection for early fire-alarming system base on video processing", in *IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, IIH-MSP'06, Pasadena, CA, USA, 2006.

[5] D. L. Siqueira, A. M. C. Machado, "People Detection and Tracking in Low Frame-rate Dynamic Scenes", *IEEE Latin America Transactions*, vol. 14, no. 4, pp. 1966-1971, 2016.

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, "Backpropagation applied to handwritten zip code recognition", *Neural Computation*, vol.1, no. 4, pp. 541-551, 1989.

[7] F. H. D. Araújo, A. C. Carneiro, R. V. Silva, F. N. S. Medeiros, D. M. Ushizima "Redes Neurais Convolucionais com Tensorflow: Teoria e Prática", III Escola Regional de Informática do Piauí. Livro Anais – ERI 2017, vol. 1, no. 1, Piauí, pp382-406, 2017.

[8] T. Guo, J. Dong, H. Li, Y. Gao, "Simple convolutional neural network on image classification", in *IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, Beijing, China, 2017.

[9] J. Yang, J. Li, "Application of deep convolution neural network", in *IEEE 14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China, 2017.

[10] C. Silva, D. Welfer, F. P. Gioda, C. Dornelles, "Cattle Brand Recognition using Convolutional Neural Network and Support Vector Machines", *IEEE Latin America Transactions*, vol. 15, no. 2, pp. 310-316, 2017.

[11] C. Silva, D. Welfer, F. P. Gioda, C. Dornelles, "Cattle Brand Recognition using Convolutional Neural Network and Support Vector Machines", IEEE Latin America Transactions, vol. 15, no. 2, pp. 310-316, 2017.

[12] K. He, G. Gkioxari, P. Dollár, R. Girshick, "Mask R-CNN", in *IEEE International Conference on Computer Vision* (ICCV), Venice, Italy, 2017.

[13] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, 2017.

[14] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. "Spatial transformer networks", in *Twenty-ninth Conference on Neural Information Processing Systems* (NeurIPS), 2015.

[15] G. C. Silva, "Detecção e Contagem de Plantas utilizando Técnicas de Inteligência Artificial e Machine Learning", Graduação em Engenharia Eletrônica, Centro Tecnológico, Universidade Federal de Santa Catarina, 2018.

[16] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection", in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'05), San Diego, CA, USA, vol. 1, pp. 886-893, 2005.

[17] D. E. King, "Max-margin object detection". arXiv preprint arXiv:1502.00046 (2015). [Online]. Available: https://arxiv.org/abs/1502.00046

[18] T. Baltrušaitis, P. Robinson, and L-P. Morency, "OpenFace: An open source facial behavior analysis toolkit", in *IEEE Winter Conference on Applications of Computer Vision* (WACV), Lake Placid, NY, USA, 2016.

[19] G. M. Oliveira-Junior, "Máquina de Vetores Suporte: estudo e análise de parâmetros para otimização de resultado", Graduação em Ciência da Computação, Universidade Federal de Pernambuco, 2010.

[20] H. B. Soares, "Análise e classificação de imagens de lesões da pele por atributos de cor, forma e textura utilizando máquina de vetor de suporte", Ph.D. dissertation, doutorado em Engenharia Elétrica e Computação, Universidade Federal do Rio Grande do Norte, 2008.

[21] S. Sural, G. Qian, S. Pramanik. "Segmentation and histogram generation using the HSV color space for image retrieval", in *Proc. IEEE International Conference on Image Processing*, NY, USA, vol. 2, pp. 589-592, 2002.

[22] FFmpeg, 2018. [Online]. Available: https://www.ffmpeg.org/.

[23] R. Gandhi, "R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Object Detection Algorithms", 2018. [Online]. Avaliable: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.

[24] Tensorflow, 2018. [Online]. Available: https://www.tensorflow.org/?hl=pt-br.

[25] M. Abadi et al., "Tensorflow: Large-scale Machine Learning on Hererogeneous Distributed Systems", in *Proc. Distributed, Parallel, and Cluster Computing*, 2016.

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

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

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

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

Mário Augusto Pazoti graduado em Ciência da Computação pela Universidade do Oeste Paulista (Unoeste), Brasil (2001) e Mestre em Ciência da Computação e Matemática Computacional pela Universidade de São Paulo (USP), Brasil (2005). Atualmente é professor na Universidade do Oeste Paulista (Unoeste), Brasil.

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

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