http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2022.v14.n1.e386
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 16/12/2021 Revisado: 18/05/2022 Aceito: 22/06/2022

USO DE DEEP LEARNING APLICADO NO RECONHECIMENTO DE AÇÕES HUMANAS A PARTIR DE VÍDEOS EM ALTA RESOLUÇÃO VISANDO IDENTIFICAR MOVIMENTOS SUSPEITOS

USE OF DEEP LEARNING APPLIED TO RECOGNITION OF HUMAN ACTIONS FROM HIGH-RESOLUTION VIDEOS TO IDENTIFY SUSPICIOUS MOVEMENTS

Henrique Krupck Secchi, Silvio Antonio Carro

Universidade do Oeste Paulista –UNOESTE, Faculdade de Informática de Presidente Prudente, Presidente Prudente, SP.
E-mail: krupck@outlook.com; silvio@unoeste.br

RESUMO – A utilização de visão computacional desempenha um papel importante para fins de segurança. Porém, a combinação com técnicas de aprendizado profundo e redes neurais convolucionais ainda são pouco exploradas por demandarem bastante capacidade de processamento computacional. Este trabalho tem por objetivo combinar essas técnicas com o objetivo de gerar um algoritmo que seja capaz de identificar e rastrear indivíduos em vídeos, além disso, monitorar suas ações com o propósito de identificar movimentos que possam significar um ato criminoso, utilizando o algoritmo do YOLO para a identificação, filtro de Kalman para o rastreamento e BlazePose para a identificação dos movimentos. Como resultado foi obtido uma taxa de precisão de 95% em vídeos bem definidos e uma taxa de 81% de precisão utilizando vídeos de câmeras de segurança mais populares.
Palavras-chave: Visão computacional; CNN; Detecção; Segurança; Rastreamento.

ABSTRACT – The use of computer vision plays an important role for security purposes. However, the combination with deep learning techniques and convolutional neural networks are still little explored because they demand a lot of computational processing capacity. This work aims to combine these techniques in order to generate an algorithm that is capable of identifying and tracking individuals in videos, in addition to monitoring their actions with the purpose of identifying movements that could signify a criminal act, using the YOLO algorithm for identification, Kalman filter for tracking and BlazePose for movement identification. This work resulted in a 95% accuracy rate on well-defined videos and an 81% accuracy rate using video from the most popular security cameras.
Keywords: Computer vision: CNN: Detection; Security; Tracking;

1. INTRODUÇÃO

A segurança pública é um tema que, de forma praticamente diária está em pauta na imprensa do Brasil. A sensação de insegurança, somada ao medo, está presente na vida de grande parte da sociedade civil brasileira,

principalmente nos grandes centros urbanos, onde a criminalidade se concentra de maneira mais densa (NAVEGA, 2018).

Na China, para combater a criminalidade, o governo utiliza o maior e mais moderno sistema de vigilância do mundo. Até o ano de 2017, o país possuía 170 milhões de câmeras espalhadas por todo o seu território. As imagens são submetidas a um algoritmo de reconhecimento facial para que, dessa forma, os órgãos competentes possam monitorar a maior parte da população do país. Sendo assim, é possível identificar e deter os suspeitos e criminosos de maneira muito mais ágil e eficiente (BBC, 2017).

Atualmente, o reconhecimento de imagens é uma subárea da visão computacional que está bastante em alta com aplicações nos mais variados campos como autenticação biométrica, veículos autônomos, medicina, aprimoramento de segurança, dentre outras aplicações (ANANTHAKUMAR, 2018; YUAN *et al.,* 2017). Esse tipo de aplicação se fez possível por causa das chamadas redes neurais artificiais, que não é um conceito novo na computação, mas que ganhou bastante força nos últimos anos graças ao crescente poder de processamento das máquinas modernas (JIACONDA, 2016).

Motivado pela popularização do uso de câmeras de segurança digitais e com a crescente evolução das técnicas de visão e aprendizado computacionais, wste trabalho propõe, por meio da combinação de diferentes algoritmos, o desenvolvimento de um sistema de identificação e rastreamento de pessoas que auxilie a área de segurança e podem ser utilizadas em ambientes monitorados por câmeras de vigilância de baixo custo. Para a identificação do individuo foram utilizados algoritmos de redes neurais convolucionais e aprendizado profundo, sendo utilizada a técnica YOLO, que está em bastante evidência atualmente, por ser bastante eficiente e com ótimo desempenho nesse contexto.

Para o rastreamento foi utilizado outro tipo de algoritmo, o filtro de Kalman, que não surgiu com o propósito de ser utilizado para essa finalidade, mas se mostrou bastante eficaz para o rastreamento de objetos em vídeos (WELCH; BISHOP, 2006).

A técnica utilizada para a identificação dos movimentos dos indivíduos é o algoritmo Blazepose, uma solução de Machine Learning que tem como saída 33 pontos de referência do corpo humano (BAZAREVSKY, 2020). Com os pontos, foram gerados vetores entre eles e foi realizado o cálculo dos ângulos entre os vetores para identificar qual o movimento do indivíduo.

Com o cálculo dos movimentos realizados foi possível saber qual é a ação que o individuo está fazendo na cena, e a partir disso, identificar se é um gesto suspeito, e caso seja, o algoritmo emite uma sinalização informando que na cena possui um elemento suspeito que possa indiciar um ato criminoso.

2. REDES NEURAIS CONVOLUCIONAIS

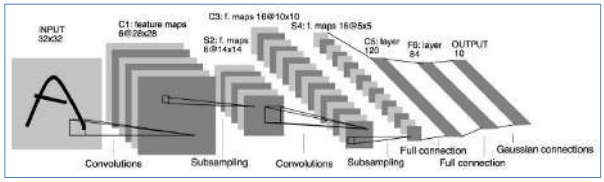
2.1. Introdução

Para o processamento de imagens, um tipo específico de redes neurais é utilizado, conhecidas como redes neurais convolucionais (CNN - Convolucional Neural Network). Segundo Lecun *et al.* (1998), uma CNN é um algoritmo de Aprendizado Profundo que pode captar uma imagem de entrada, atribuir uma importância (pesos e viés) a vários aspectos da imagem para conseguir classificá-las.

As redes neurais convolucionais foram desenvolvidas tomando como base o córtex visual de animais, que é composto por milhões de agrupamentos celulares complexos, sensíveis a pequenas sub-regiões do campo visual, chamadas de campos receptíveis (HUBEL, 1968). Essas regiões, nas redes neurais convolucionais, são também denominadas de campos receptivos, e são formadas por subconjuntos selecionados do vetor de características (representação numérica do objeto) a ser analisado.

A rede neural convolucional é composta principalmente por várias camadas, que são denominadas como: entrada, convolucional, pooling, totalmente conectada e de saída. Com a Rede Neural Convolucional é possível extrair características e realizar o mapeamento da mesma com um treinamento da rede mais rápida, além de ter alta precisão. O seu principal uso é para classificações e previsões de imagens (LECUN *et al.,* 1998; SHEN; WANG, 2018) ().

Figura 1. Ilustração da arquitetura LeNet com a imagem de entrada e as camadas: convolucionais, pooling, totalmente conectadas e saída.

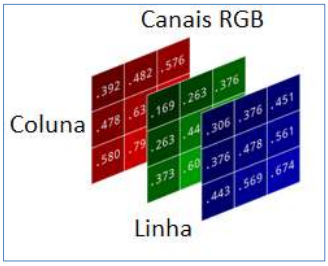


Fonte: (LECUN *et al*., 1998)

2.2. Entradas

Quando se fala em Redes Neurais Convolucionais para o reconhecimento e classificação de imagens, as entradas são matrizes tridimensionais com altura e largura que variam de acordo com as dimensões da imagem e profundidade, definida pela quantidade de canais de cores. Na maior parte dos casos, as imagens utilizam três canais, RGB, com os valores de cada pixel.

Figura 2. Exemplo de matriz de pixels 3x3x3, contendo os três canais RGB.

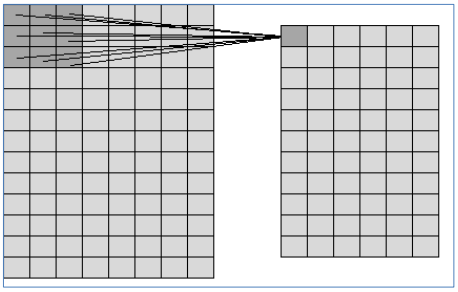


Fonte: Os autores.

2.3. Convoluções

As convoluções funcionam como filtros que enxergam pequenos quadrados e vão seguindo por toda a imagem captando os traços mais marcantes (ALVES, 2018). De maneira mais prática, com uma imagem 8x13x3 e um filtro que cobre uma área de 3x3 da imagem com movimento de 1 salto (chamado de stride), o filtro passará pela imagem inteira, por cada um dos canais, formando no final um *feature map* ou *activation map* de 6x11x1 (Figura 3).

Figura 3. Na esquerda a matriz de pixels original e na direita a matriz de pixels sendo formada após ser submetida ao filtro.



Fonte: Os autores.

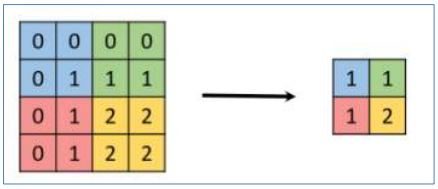
2.4. Funções de ativação

As funções de ativação servem para trazer a não-linearidades ao sistema, para que a rede consiga aprender qualquer tipo de funcionalidade, independente da sua complexidade. Há várias funções, como softmax, sigmoid, tanh, mas a mais indicada para redes convolucionais é a Relu por ser mais eficiente em termos de desempenho e sem grandes diferenças de precisão ao ser comparada a outras funções. Essa função zera todos os valores negativos da saída da camada anterior (ALVES, 2018).

2.5. Pooling

Após uma camada de convolução, comumente existe uma camada chamada de pooling. A função desta é reduzir de maneira progressiva a dimensão da matriz de entrada, preservando as suas principais características. Deste modo, a redução diminui o custo computacional da rede e tende a evitar o *overfitting* (KARPATHY, 2018). A forma mais comum de pooling consiste em substituir os valores de uma região pelo valor máximo (GOODFELLOW *et al*., 2016). Esta técnica é conhecida como max pooling, e serve para eliminar valores irrelevantes, reduzindo o dimensionamento da matriz de pixels e acelerando o desempenho computacional da rede, além de criar uma invariância e pequenas mudanças e distorções locais (Figura 4) (ARAÚJO *et al.,* 2017).

Figura 4. Aplicação do max pooling em uma imagem 4x4 utilizando um filtro de 2x2.

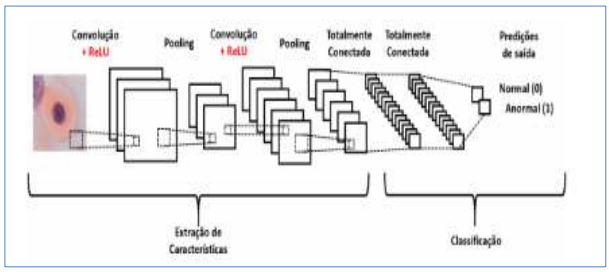


Fonte: (ARAÚJO *et al.*, 2017).

2.6 Fully Connected

Ao final da rede é colocada uma camada totalmente conectada (figura 5), cujo entrada é a saída da camada anterior e sua saída são N neurônios, com N sendo a quantidade de classes do seu modelo para finalizar a classificação (ARAÚJO *et al.,* 2017).

Figura 5. Ilustração da extração de características de uma imagem por uma Rede Neural Convolucional e sua posterior classificação.



Fonte: (ARAÚJO *et al.,* 2017).

As camadas são formadas por unidades de processamento denominadas neurônios, e o termo "totalmente conectado" significa que todos os neurônios da camada anterior estão conectados a todos os neurônios da camada posterior (ARAÚJO et al, 2017).

2.7. Introdução ao YOLO

A YOLOv3 abreviação de You Only Look Once é uma rede muito utilizada pela sua rapidez e acurácia (LIU *et al.,* 2019). Com esta arquitetura, houve melhorias tanto na precisão da detecção de objetos quanto na otimização usando GPU, tornando-se mais eficiente o uso computacional (FACHRIE, 2020). A arquitetura YOLO é apresentada na Seção 3.2.

2.8. Filtro de Kalman

Com a imagem recortada, foi aplicado o algoritmo do filtro de Kalman para conseguir identificar a variância na matriz de pixels, e caso a variância seja menor que o limiar estabelecido, a equação vai retornar que o objeto continua sendo o mesmo, e dessa forma, rastrear o objeto ao longo da cena.

O Filtro de Kalman consiste em um conjunto de equações que possibilitam a implementação recursiva de um estimador, gerando predição ótima dos estados futuros de um sistema linear a partir de uma observação presente (WELCH; BISHOP, 2006). O sistema que busca gerar estimativas ótimas dos estados de um sistema é descrito pela seguinte equação 3:

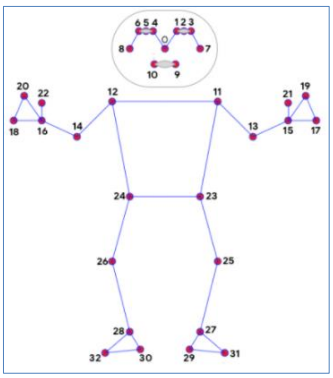
$$x_{k+1} = Ax_k + Bu_k + w_k$$
$$y_k = Cx_k + v_k \quad (3)$$

No qual $x_k \in R^n$ é o vetor de estados, $A \in R^{nxn}$ é a matriz de estado, $B \in R^{nxm}$ é a matriz de entrada, $u_k \in R^m$ é o vetor de entrada, w_k representa a incerteza associada a modelagem do processo, no qual é assumido como sendo uma distribuição gaussiana, com média nula, $y_k \in R^p$ o vetor de saída, $C \in R^{pxn}$a matriz de saída e v_k a incerteza associada a medição da saída. Da mesma forma, v_k é assumido como sendo gaussiano, com média nula e w_k e v_k não possuem correlação (WELCH; BISHOP, 2006).

2.9. O Algoritmo Blazepose

Blazepose é uma arquitetura de rede neural convolucional para a estimativa de pose do corpo humano. Durante a inferência, a rede produz 33 pontos do corpo (figura 6). Os pontos são distribuídos nos seguintes locais: nariz, olho direito, olho esquerdo, orelhas, boca, ombro, cotovelo, pulso, dedo mindinho, dedo indicador, polegar, quadril, joelho, tornozelo, calcanhar e pé (BAZAREVSKY *et al.,* 2020).

Figura 6. Todos os pontos exibidos de maneira visual.

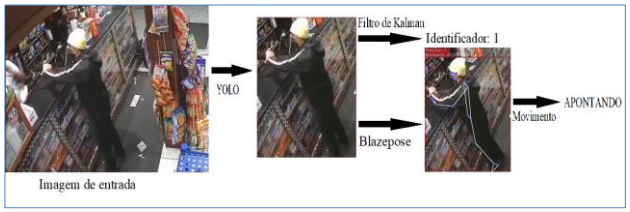


Fonte: (Mediapipe, 2020)

3. MATERIAIS E METODOLOGIA

3.1. Arquitetura do Projeto

Figura 7. Arquitetura do Projeto em um esquema visual.



Fonte: Os autores.

O projeto utilizou a arquitetura YOLO v3, para realizar a primeira etapa do algoritmo, a etapa de detecção do objeto pessoa. O filtro de Kalman é um conjunto de equações e foi utilizado para realizar o monitoramento do objeto no decorrer do vídeo e o algoritmo Blazepose foi utilizado para identificar os pontos de referência do corpo humano. Com os pontos, foi possível calcular o ângulo entre as partes do corpo, e dessa forma identificar o movimento do indivíduo conforme ilustra a Figura 7.

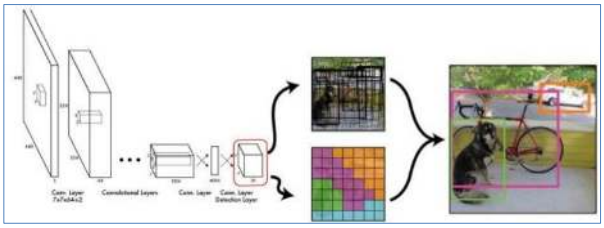
O algoritmo foi desenvolvido e testado com o seguinte hardware: Processador: 7th Generation Intel® Core™ i3, 3.9GHz, 64 bits; Windows 10; Memória RAM: 8GB DDR4; Disco rígido: SSD 120GB e placa de vídeo: GEFORCE GTX 1050ti 4GB;

Para treinamento da rede YOLO foi utilizado o COCO Dataset, que é um dataset público que contém mais de 330 mil imagens segmentadas específico para realizar treinamento de redes neurais.

3.2. Arquitetura YOLO

A primeira etapa do método consiste em submeter o vídeo ao algoritmo YOLO (Figura 8) para que possa ser realizada a identificação do objeto de interesse, o indivíduo no caso. O YOLO realiza a classificação e a previsão de caixas delimitadoras para objetos detectados. É altamente otimizado para desempenho de detecção e pode ser executado muito mais rápido do que executar duas redes neurais separadas para detectar e classificar objetos separadamente.

Figura 8. Arquitetura do YOLO em um esquema visual.



Fonte: (ALVES, 2020).

O YOLOv3 é uma rede neural convolucional que tem um total de 24 camadas convolucionais e seguidas por 2 camadas totalmente conectadas. Cada camada tem sua própria importância e as camadas são separadas por sua funcionalidade (ALVES, 2020).

As primeiras 20 camadas convolucionais seguidas por uma camada de pooling e uma camada totalmente conectada são pré treinadas no conjunto de dados COCO, que é um conjunto de dados de classificação de 1000 classes (ALVES, 2020).

O pré treinamento para classificação é realizado no conjunto de dados com resolução de imagem de 224 x 224 x 3. As camadas compreendem 3 x 3 camadas convolucionais e 1 x 1 camadas de redução (ALVES, 2020).

Para a detecção de objetos, no final, as últimas quatro camadas convolucionais seguidas por duas camadas totalmente conectadas são adicionadas para treinar a rede. A detecção de objetos requer detalhes mais precisos, portanto, a resolução do conjunto de dados é aumentada para 448 x 448 (ALVES, 2020).

Em seguida, a camada final prevê as probabilidades de classe e caixas delimitadoras. Todas as outras camadas convolucionais usam ativação ReLU (1), enquanto a camada final usa uma ativação linear (2).

$$ReLU(x) = max\{0, x\} \quad (1)$$

$$f(x) = ax \quad (2)$$

A entrada é uma imagem com resolução de 448 x 448 e a saída é a previsão de classe do objeto detectado dentro da caixa delimitadora. Para este trabalho, as classes de saída foram filtradas para que o algoritmo retorne apenas o objeto 'Person', identificando apenas seres humanos.

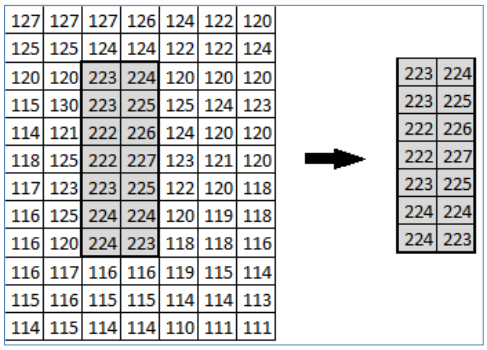
O YOLOv3 possui dois tipos de redes neurais, a YOLOv3 propriamente dita e uma arquitetura que preserva somente as principais

camadas, chamada YOLOv3 tiny. A YOLOv3 tiny possui a vantagem de performance já que se trata de uma versão com menos camadas, reduzindo assim, a quantidade de processamento realizada por segundo. Ambas as arquiteturas foram testadas com o intuito de verificar qual arquitetura se saiu melhor em uma análise de performance e precisão.

3.3. Etapa de recorte

A próxima etapa é realizar o recorte do objeto e selecionar somente o pedaço da imagem que corresponde ao objeto identificado, esse processo se faz necessário para as etapas seguintes, tanto a parte de rastreamento quanto a parte de identificação dos movimentos (Figura 9).

Figura 9. Matriz de pixels com o objeto identificado e o recorte do objeto.



Fonte: Autoral.

3.4. Utilizando o Filtro de Kalman

Com a imagem recortada, foi aplicado o algoritmo do filtro de Kalman para conseguir identificar a variância na matriz de pixels, e caso a variância seja menor que o limiar estabelecido, a equação vai retornar que o objeto continua sendo o mesmo, e dessa forma, rastrear o objeto ao longo da cena.

3.5. O Algoritmo Blazepose

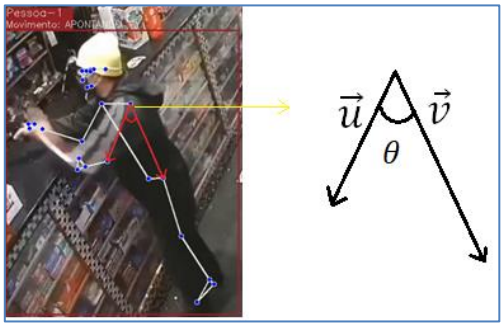
A imagem que foi recortada após a detecção do YOLO é submetida ao algoritmo Blazepose.

Com o objeto contendo os pontos, e cada ponto contendo a posição X, Y e Z dentro da matriz de pixels, a próxima etapa foi gerar vetores entre os pontos possibilitando calcular os ângulos formados entre os vetores. É possível obter o ângulo entre dois vetores v (x_1,y_1) e u (x_2,y_2) utilizando a fórmula para produto interno:

$$\cos\theta = \frac{\langle v,u \rangle}{|v|\cdot|u|} \quad (4)$$

Com os ângulos, é possível identificar cada ponto em relação a outro do objeto, com isso, foi possível identificar qual é o movimento do indivíduo na cena (Figura 10).

Figura 10. Exemplo visual de um objeto detectado, rastreado e uma representação gráfica da geração dos vetores o calculo dos ângulos entre os vetores para a identificação do movimento.



Fonte: Autoral.

3.6. Lista de movimentos suspeitos

Ao identificar o movimento, os resultados são comparados com uma lista de movimentos possíveis, Esta contém os seguintes movimentos: 'em pé', 'sentado', 'rendido' e 'apontando', sendo, os dois últimos considerados suspeitos, e, caso o movimento esteja na classe de movimentos suspeitos, o algoritmo pinta de vermelho o contorno do indivíduo durante todo o decorrer da cena, indicando que este é um indivíduo suspeito e que pode cometer um ato criminoso a qualquer momento.

4. RESULTADOS

Para a realização deste trabalho foram estudadas as duas arquiteturas do YOLOv3, a YOLOv3 e a sua versão reduzida, a YOLOv3 tiny, a fim de obter a arquitetura com melhor equilíbrio entre desempenho e precisão.

O primeiro teste foi fazer um comparativo de desempenho das duas versões do YOLO sendo executadas tanto na CPU quanto na GPU. Esse teste foi realizado utilizando apenas as redes YOLO, sem a aplicação das etapas de rastreamento e identificação do movimento. O intuito foi verificar o quão performático elas se saíram no hardware utilizado neste trabalho (Tabela 1).

Tabela 1. Comparativo de desempenho nas versões YOLO.

Arquitetura	Process.	Resolução	FPS
YOLOv3	CPU	1088x614	10~14
YOLOv3	GPU	1088x614	23~27
YOLOv3 tiny	CPU	1088x614	20~23
YOLOv3 tiny	GPU	1088x614	48~53

Fonte. Os autores.

É possível notar que mesmo sendo executadas na CPU as redes possuem uma desempenho satisfatório e na GPU os desempenhos de ambas foram ótimos, em um dos casos se aproximando de 60 FPS (quadros por segundo). Vale ressaltar que este teste não avaliou a precisão das redes, e sim o desempenho que elas obtiveram.

Para realizar os próximos passos, foram utilizadas duas categorias de vídeos. A primeira categoria contém vídeos com altíssima nitidez e com os objetos em destaque na cena, justamente para facilitar a detecção do objeto, rastreamento e detecção do movimento. A segunda categoria de vídeos são imagens retiradas de câmeras de segurança, possuem menor nitidez e os movimentos dos objetos não estão totalmente evidentes (Tabela 2).

Tabela 2. Descrição dos tipos de vídeos utilizados nos teste.

Qtd.	Categoria	Descrição
6	1	Vídeos que ressaltam os objetos e seus movimentos.
3	2	Vídeos retirados a partir de câmeras de segurança.

Fonte: Autoral.

A próxima etapa de testes foi avaliar o desempenho e a precisão que as redes obtiveram, agora aplicando a etapa rastreamento, com o filtro de Kalman e a etapa de identificação do movimento. Este passo utilizou as duas categorias de vídeos, a fim de analisar a precisão em cada categoria de maneira individual (Tabela 3).

Tabela 3. Testes da detecção dos objetos na cena.

Arq.	Process.	FPS	Categoria	Precisão detecção
YOLOv3	CPU	Nulo	1	Nulo
YOLOv3	GPU	1~3	1	95%
YOLOv3 tiny	CPU	8~13	1	85%
YOLOv3 tiny	GPU	17~24	1	85%
YOLOv3	CPU	Nulo	2	Nulo
YOLOv3	GPU	1~3	2	81%
YOLOv3 tiny	CPU	8~13	2	65%
YOLOv3 tiny	GPU	17~24	2	65%

Fonte: Os autores.

Após realizar o treinamento e submeter o algoritmo as arquiteturas notou-se que o ganho de desempenho da versão tiny foi muito superior a versão completa da rede e, nos casos em que os indivíduos na cena estão bem definidos, a rede teve uma ótima eficiência. Porém, nos vídeos retirados a partir de câmeras de segurança a versão tiny perdeu consideravelmente eficiência e não se saiu satisfatória (Tabela 3). Os casos que foram marcados como NULO representam situações em que o desempenho se saiu tão ruim que se tornou inviável a realização dos testes.

O passo seguinte foi para conferir a precisão na detecção do objeto, o seu monitoramento no decorrer da cena e a precisão na identificação do seu movimento. A partir deste momento, a arquitetura YOLOv3 tiny foi descartada do trabalho por seu desempenho ter sido considerada abaixo do satisfatório no teste anterior. Considerando apenas a YOLOv3, temos (Tabela 4)

Tabela 4. Testes em todas as etapas: Detecção, Monitoramento e Identificação dos movimentos.

Categoria	Precisão detecção dos indivíduos	Precisão no rastreamento.	Precisão identificação dos movimentos
1	95%	97%	79%
2	81%	92%	63%

Fonte: Os autores.

A grande dificuldade do algoritmo foi a detecção dos chamados falsos positivos (Tabela 5), isto é, os objetos que não deveriam ser

classificados como 'suspeitos' e são classificados de forma errônea simplesmente por erguerem o braço em algum momento da cena.

Tabela 5. Matriz de confusão dos testes verdadeiros e falsos, positivos e negativos.

		Valor previsto	
		SUSPEITO	**NÃO SUSPEITO**
Class.	**SUSPEITO**	Verdadeiro Positivo 93**%**	Falso Negativo 0%
	NÃO SUSPEITO	Falso positivo 45%	Verdadeiro Negativo 99%

Fonte: Os autores.

O teste utilizando a matriz de confusão permitiu uma análise detalhada das predições corretas e falsas (Tabela 5). Nota-se que os valores produzidos na coluna "Falsos positivos" resultaram em uma porcentagem de erro bem expressiva, evidenciando a complexidade do problema em análise, o que poderá ser atenuado com a adoção de técnicas que consigam predizer e descartar movimentos potencialmente suspeitos, mas sem consequências, como por exemplo, um tropeço ao caminhar em um ambiente.

5. CONSIDERAÇÕES FINAIS

Este trabalho foi motivado pela constante taxa de criminalidade que assola o país, e pelo crescimento de trabalhos utilizando redes neurais convolucionais aplicados nas mais diversas áreas. O presente trabalho utilizou dois tipos de redes neurais convolucionais: o Yolo e o Blazepose em diferentes etapas, cada uma com um propósito distinto. Além disso, utilizou métodos matemáticos para realizar o rastreamento e o calculo dos movimentos.

A primeira rede escolhida foi o Yolo, que atualmente é o que se tem de mais preciso e performático se tratando de identificação de objetos. O Yolo em sua versão completa obteve uma taxa de precisão 95% em vídeos bem definidos e uma taxa de 81% de precisão utilizando vídeos de câmeras de segurança, o que pode ser considerado um valor bem expressivo, dado que nesses vídeos a qualidade da imagem é bastante limitada.

O algoritmo YOLO está com uma alta acurácia no que diz respeito à detecção dos indivíduos e na parte do rastreamento, em ambas as categorias. A parte da identificação do movimento está com um nível bom na primeira categoria e um nível razoável na segunda categoria de vídeos.

Entretanto, o algoritmo se mostrou deficiente na predição dos falsos positivos, e em alguns casos ele classificou de maneira errônea os movimentos suspeitos.

Conclui-se que o trabalho é relevante, mostrando a possibilidades de utilizar novas combinações, que até então não haviam sido reunidas para essa finalidade. Diferentes algoritmos são utilizados para detecção, monitoramento e identificação do movimento da pessoa. Os resultados obtidos nos testes foram considerados satisfatórios, o que justifica ainda mais a relevância do trabalho.

Como trabalhos futuros, pode-se utilizar como base a mesma metodologia proposta, e a partir dela aplicar alguma técnica a mais para corrigir o problema dos falsos negativos, Ademais, pode-se utilizar o algoritmo de reconhecimento para um circuito de câmeras, ou ainda, utilizar outra técnica para extração das características do indivíduo.

REFERÊNCIAS

ALVES, G. **Detecção de Objetos com YOLO – Uma abordagem moderna**. 2020. Disponível em: https://iaexpert.academy/2020/10/13/deteccao-de-objetos-com-yolo-uma-abordagem-moderna/. Acesso em: 05 ago. 2021.

ALVES, G. **Understanding ConvNets (CNN).** 2018. Disponível em: https://medium.com/neuronio/understanding-convnets-cnn-712f2afe4dd3. Acesso em: 01 ago. 2021.

ANANTHAKUMAR, A.. **Efficient Face And Gesture Recognition For Time Sensitive Application. Georgia**. 2018. Disponível em: https://ieeexplore.ieee.org/document/8470351/authors#authors. Acesso em: 09 ago. 2021. https://doi.org/10.1109/SSIAI.2018.8470351

ARAÚJO, F. H. D. *et al.* Redes Neurais Convolucionais com Tensorflow: Teoria e Prática. *In***:** Escola Regional de Informática do Piauí**.** 3**.,** 2017. Picos, MA. **Anais** [...]. Picos: Sociedade Brasileira de Computação, 2017. Disponível em:https://docplayer.com.br/57119969-Redesneurais-convolucionais-com-tensorflow-teoria-e-pratica.html.. Acesso: 25 ago. 2020.

BAZAREVSKY, V. *et al.* **BlazePose:** On-device Real-time Body Pose tracking. Seattle: CVPR Workshop on Computer Vision for Augmented and Virtual Reality, 2020. Disponível em: https://arxiv.org/abs/2006.10204>. Acesso em 08 nov. 2021.

BBC. **Como funciona o 'Big Brother' da China, com 170 milhões de câmeras que fazem identificação visual.** BBC, 2017. Disponível em: https://www.bbc.com/portuguese/internacional-42361047. Acesso em: 08 fev. 2021.

FACHRIE, M. **A simple vehicle counting system using deep learning with YOLOv3 Model**. 2020. Disponível em: https://www.researchgate.net/publication/341075968_A_Simple_Vehicle_Counting_System_Using_Deep_Learning_with_YOLOv3_Model. Acesso em: 26 set. 2021. https://doi.org/10.29207/resti.v4i3.1871

GOODFELLOW, J. **Deep Learning**. Cambridge, MA: MIT, 2016. Disponível em: https://books.google.com.br/books?hl=ptPT&lr=&id=omivDQAAQBAJ&oi=fnd&pg=PR5&dq=deep+learning+goodfellow+pdf&ots=MNO3gnpzQR&sig=hcU98sXLnC5jYgQBpfMWnDZtPCI#v=onepage&q=deep%20learning%20goodfellow%20pdf&f=false. Acesso em: 22 set. 2021.

HUBEL, D. H.; WIESEL, T. N. Receptive fields and functional architecture of monkey striate cortex. **The Journal of Physiology**, v. 195, n. 1, p. 215–243, mar. 1968.. https://doi.org/10.1113/jphysiol.1968.sp008455

JIACONDA. **A Concise History of Neural Networks. Towards Data Science**, 2016. Disponível em: https://towardsdatascience.com/a-concise-history-ofneuralnetworks-2070655d3fec. Acesso em: 09 ago. 2021.

KARPATHY, A. **CS231n Convolutional Neural Networks for Visual Recognition**. Disponível em: http://cs231n.github.io/convolutional-networks/. Acesso em: 15 ago. 2021.

LECUN, Y. et al. **Gradient-Based Learning Applied to Document Recognition. Montréal**, 1998. Disponível em: http://yann.lecun.com/exdb/publis/pdf/lecun98.pdf. Acesso em: 29 jul. 2021.

LIU, R. *et al.* An Improved YOLOV3 for Pedestrian Clothing Detection. *In*: INTERNATIONAL CONFERENCE ON SYSTEMS AND INFORMATICS. 6, 2019. Disponível em: https://ieeexplore.ieee.org/abstract/document/9010512. Acesso em: 15out. 2021. https://doi.org/10.1109/ICSAI48974.2019.9010512

MEDIAPIPE. **Pose Landmark Model (BlazePose GHUM 3D)**, 2020. Disponível em: https://google.github.io/mediapipe/solutions/pose.html#pose-landmark-model-blazepose-ghum-3d. Acesso em: 08 nov. 2021.

NAVEGA, P. C. G.. Controle da Criminalidade: Problema de Polícia ou de Políticas?. **Jus Navigandi**, 2018. Disponível em: https://jus.com.br/artigos/64862/controle-da-criminalidade-problema-de-policia-ou-de-politicas. Acesso em: 08 mar. 2021.

SHEN, W.; WANG, W. Node Identification in Wireless Network Based on Convolutional Neural Network. *In*: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND SECURITY. 2018. Disponível em: https://ieeexplore.ieee.org/abstract/document/8564297. Acesso em: 18 out. 2021. https://doi.org/10.1109/CIS2018.2018.00059

YUAN, L. *et al.* **A convolutional neural network based on TensorFlow for face recognition**. Lanzhou, 2017. Disponível em: https://ieeexplore.ieee.org/document/8054070 Acesso em: 13 ago. 2021.

WELCH, G..; BISHOP, G. **An introduction to the kalman filter, Chapel Hill**: University of North Carolina at Chapel Hill, 2006. Disponível em: https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf. Acesso em: 06 maio 2021.