Análise experimental para a detecção de objetos em vídeos de câmeras de vigilância

Uma abordagem para porte de arma, incêndio e pichação

Natan Moura
natan.moura@ufba.br
FORMAS Research Group
Instituto de Computação
Universidade Federal da Bahia
Salvador, Bahia – Brasil

Daniela Barreiro Claro
dclaro@ufba.br
FORMAS Research Group
Instituto de Computação
Universidade Federal da Bahia
Salvador, Bahia – Brasil

João Medrado Gondim
joao.gondim@ufba.br
FORMAS Research Group
Instituto de Computação
Universidade Federal da Bahia
Salvador, Bahia – Brasil

ABSTRACT

Muitas técnicas têm sido desenvolvidas em prol da vigilância automatizada, desde alarmes inteligentes aos sistemas de câmeras. A automatização da análise de imagens em vídeos, principalmente a detecção de objetos tem crescido nos últimos tempos.Este trabalho tem por principal objetivo detectar objetos específicos para auxiliar no alerta sobre porte de armas, incêndios e pichações, em tempo real, aos órgãos da Secretaria de Segurança Pública da Bahia.

KEYWORDS

detecção, objetos, *Yolo*, Câmera de Vigilância

1 INTRODUÇÃO

A vigilância de ambientes tem sido o foco para diversos trabalhos, gerando métodos capazes de detectar a ocorrência de variações inesperadas, tais como violência, vandalismo e incêndio.

Esse estudo analisou diversas técnicas para o reconhecimento de objetos em bases de artigos (*IEEE EXTREME*, *SCIENCEDIRECT*, *ACM* e *SCOPUS*) e revelou diversos métodos e algoritmos, dentre os quais se destacam o Yolo, *Faster R-CNN*, *SSD* na detecção de objetos.

Dentre os principais métodos, o *Yolo* [6] analisa a imagem somente uma vez, maximizando a velocidade da detecção e mantendo a acurácia. Ajustes foram realizados nas diferentes versões, sendo a versão 5 a utilizada neste trabalho devido ao desempenho superior na detecção de objetos pequenos. Assim, o principal objetivo deste trabalho foi analisar o comportamento da ferramenta *YoloV5* na detecção de objetos pequenos (treinando-os em distintos *datasets*), tais como armas, facas e sprays, de modo a verificar a possibilidade da utilização do *Yolo* na análise de incidentes em tempo real (porte de arma, incêndios, pichações), para notificação desses à Secretaria de Segurança Pública da Bahia.

Este artigo está organizado em seções. A seção 2 expõe os trabalhos relacionados; a seção 3 apresenta a pesquisa bibliográfica e os datasets encontrados; a seção 4 apresenta a metodologia utilizada para treino dos métodos de detecção; a seção 5 expõe os resultados e a seção 6 apresenta as considerações finais e os trabalhos futuros.

In: I Concurso de Trabalhos de Iniciação Científica (CTIC 2021), Minas Gerais, Brasil. Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia). Porto Alegre: Sociedade Brasileira de Computação, 2021.

ISSN 2596-1683

2 TRABALHOS RELACIONADOS

Dentre os trabalhos relacionados, os autores em [8] se aproximam no quesito de detecção de armas com experimentação através do *YoloV3* para detecção de armas em vídeos. Eles utilizaram um dataset próprio e não fizeram pré-treino em dataset de imagens computadorizadas. Além disso, a versão do *Yolo* não é otimizada para detecção de objetos pequenos. Autores em [3] utilizam o dataset de pré-treino com imagens computadorizadas produzidas em Unity [3], porém não fazem uso de nenhum outro dataset, além de utilizarem *Faster R-CNN* ao invés do *Yolo*. Autores em [5] fazem um estudo da relação custo x eficiência de diversas CNNs na tarefa de classificação de imagens contendo fogo ou não, focando principalmente em sistemas de vigilância. Autores em [2] utilizam da variação em pixels de imagens para detectar vandalismo, como pichação, não utilizando rede neural nem algoritmos como *Yolo*.

Diferentemente dos demais, o presente trabalho é baseado majoritariamente na utilização do *YoloV5* sobre datasets distintos. Diferente dos autores em [8], este trabalho utiliza o *YoloV5* sobre dataset *Unity* para pré-treino e *Granada* para detecção de armas. Em relação ao trabalho em [3], embora haja uma semelhança no treino com *Unity*, o presente trabalho utiliza o *Granada* para treino de reconhecimento das armas e o *Yolo* ao invés de *Faster R-CNN*. Em comparação aos autores em [5], a arquitetura definida em [7] foi adaptada para detecção de fogo, utilizando o dataset descrito em [1]. E quanto à detecção de grafite, o dataset utilizado foi criado e utilizado no *YoloV5*, diferindo dos demais métodos utilizados em [2].

3 PESQUISA BIBLIOGRÁFICA

A pesquisa bibliográfica objetivou responder à pergunta: *Qual a técnica deve ser utilizada para detecção de objetos pequenos?* . Uma busca nas bases (*IEEE EXTREME*, *SCIENCEDIRECT*, *ACM* e *SCOPUS*) foi realizada visando a detecção de violência em vídeos de uma forma mais abrangente. As palavras-chave nesta etapa foram: *violence detection* e *computer vision*. No total, 99 trabalhos foram recuperados dos quais, após filtros de duplicata e trabalhos não relacionados, foram reduzidos à 26. Os 26 trabalhos foram manualmente planilhados conquanto aos métodos referenciados (Figura 1), datasets utilizados (Figura 2) e tipos de metodologia.

Embora o *Yolo* apareça somente em 3, 2% dos trabalhos na Figura 1, esta técnica foi escolhida por ser uma das poucas encontradas capaz de detectar objetos pequenos com rapidez e precisão. O método mais referenciado, *Optical Flow*, foca primeiramente na variação dos

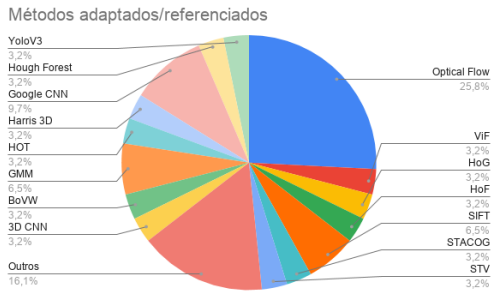

Figura 1: Métodos encontrados na pesquisa bibliográfica

pixels em uma imagem e assim não tem desempenho satisfatório em se tratando de detectar a presença de armas em vídeos providos por câmeras de vigilância.

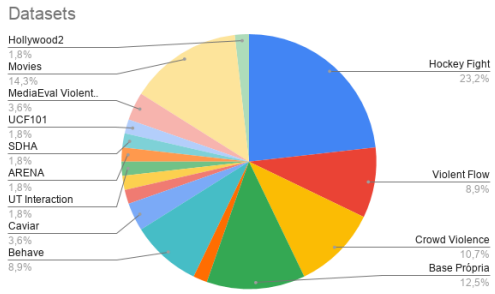

Figura 2: Datasets listados em artigos na pesquisa bibliográfica

A Figura 2 aborda os datasets públicos ou privados utilizados nos trabalhos relacionados, observando que 12, 5% destes possuem base privada e foram criados exclusivamente para uso nos trabalhos. Importante apontar que após análise do conteúdo destes datasets foi identificada uma deficiência de mapeamento para detecção de objetos, já que estes contém principalmente imagens de violência física em esportes e filmes. Portanto, passos posteriores consistiram na busca e testes de datasets mais adequados para o presente trabalho, os quais são detalhados nas seçoes seguintes.

3.1 Datasets

A combinação do uso dos diferentes datasets neste trabalho é uma das suas principais contribuições para a detecção de objetos pequenos. O primeiro dataset utilizado para testes no *Yolo* foi o *Granada* [4]. Este dataset contém 2971 imagens de pistolas (que foram divididas em 2081 para treino - 70%, 890 para testes - 30%), anotadas em mais de 3000 *labels*, formado por armas em punho, desenhos de armas, e imagens de boa qualidade. Disponibilizado em domínio público pela Universidade de Granada, foi criado com intuito de treinar redes neurais convolucionais (através de deep learning) na detecção de armas em diferentes posicionamentos. Os testes foram realizados com toda a base juntamente com o *YoloV5*. Complementando o dataset *Granada*, o dataset *U500*[3] foi utilizado para pré-treino da rede com YoloV5. Esse dataset é formado por 500 imagens contendo pistolas e rifles criados artificialmente na plataforma Unity.

Em relação à detecção de fogo, o dataset utilizado neste trabalho para classificação de incidência ou não de fogo foi obtido em [1]. O dataset é denominado *FiSmo (Fire and Smoke Images)* e é composto da união de outros datasets junto de imagens do Flickr, Youtube e simulações. Esse dataset foi dividido em 1604 imagens contendo fogo e 3952 imagens não contendo fogo para fins de treinamento.

O dataset para detecção de ações de Graffiti é formado por 516 imagens, sendo que 15% deste é separado aleatoriamente para compor a parte de testes, correspondendo a 78 imagens. O *FormasGraffitiDataset* é composto por imagens extraídas de vídeos de *bombing*[1], ou de vídeos contendo grafitagem legal, extraídos frame a frame do Youtube e selecionados manualmente. Todos os frames selecionados foram mapeados[2]. As *labels* deste dataset são divididas em 3 classes: *Spray*- correspondendo ao objeto que cria o *graffiti*; o *Tagger*- a pessoa que realiza a ação e *Graffiti* - o objeto da ação. Esse dataset foi desenvolvido para esse trabalho e permite a identificação destes elementos através da detecção de objetos de forma que combinados possam indicar possíveis situações de pichação.

4 METODOLOGIA

A metodologia englobou a detecção de armas, detecção de fogo e a detecção de pichação.

4.1 Detecção de armas

O ambiente principal de treino das ferramentas foi o *Google Colab*, disponibilizando as especificações: Nvidia K80 ou T4, 12/16 gb de VRAM respectivamente, com 12GB de Memória Ram. A escolha da placa era definida automaticamente pelo Colab, inicialmente alocando as T4 para novos notebooks e reduzindo a capacidade para K80 conforme uso. Os experimentos iniciais consistiram em utilizar o *YoloV5* com o dataset *Granada*, com o intuito de aumentar a precisão dos pesos. Para uso em imagens de câmeras de vigilância, foi utilizado o método exposto em [3], treinado a partir dos pesos padrões do YoloV5 versão L (com boa relação entre precisão e velocidade), no dataset *Unity* com 500 imagens, e submetendo estes ao treino posterior no dataset *Granada*. Importante salientar que os treinos iniciais no *Granada* foram em 300 épocas, e os treinos no *Unity* e *Granada* com 100 épocas respectivamente. Observou-se que não necessitaria mais de 300 épocas na segunda parte, por conta do modelo não ter ganhos significativos de aprendizado.

4.2 Detecção de grafite

Para a detecção de grafite foi utilizado o *Google Colab* e o *YoloV5* em sua versão L. Primeiramente, o dataset *FormasGraffitiDataset* foi utilizado para treino com os pesos padrões do YoloV5. Entretanto, diferente da detecção de armas, os primeiros testes em vídeos não vistos foram iniciados a partir da época 70 de treinamento, pois como o método apresentava precisão média de 0.67 e precisão para detecção de *taggers* de 0.9. Assim, foi iniciada a análise de comportamento, a qual foi repetida para 100 épocas e apresentou

[1]Bombing: termo utilizado para definir os atos de pichação
[2]Os mapeamentos foram feitos através da ferramenta *makesense.ai*

resultados similares, identificando pouca diferença em relação ao peso anterior.

4.3 Detecção de fogo

Ao invés do *YoloV5* para a detecção de fogo, diferentes métodos foram utilizados [7], tendo o *Tensorflow* como backend. Importante apontar que diferentemente dos autores em [7], neste experimento, o Conv2D, camada de convolução padrão, foi utilizada já que a utilização de menos recursos computacionais não era um fator preponderante. O ambiente utilizado para treino foi o *Kaggle*, ao invés do *Google Colab*. O ambiente *Kaggle* disponibiliza NVidias K80, 12GB de VRAM, CPU de 2 Cores e 13GB de RAM. A escolha do *Kaggle* e da arquitetura citada se dá pela existência de ambiente já estruturado e pré-pronto para treinos de classificação de imagens sobre datasets, entretanto etapas posteriores do trabalho incluem treino do *YoloV5* para detecção de fogo, utilizando o próprio FiSmo.

5 RESULTADOS

5.1 Detecção de armas

Os resultados iniciais estão relacionados aos pesos obtidos pelo treino com 300 épocas no dataset *Granada*. A *precisão* obtida foi próxima de 0.8 na própria base, indicando que cerca de 80% das detecções classificadas como *verdadeiro positivo* foram validadas. Os pesos foram testados em um vídeo do Youtube não visto pela rede neural anteriormente (vídeo de assalto de um noticiário), conseguindo então identificar as armas no vídeo (Figura 3).

Figura 3: Detecção de arma feita pelo YoloV5 através dos pesos de 300 épocas com 1 dataset

Apesar de conseguir detectar armas no vídeo, muitas delas apresentam confiança menor que 0.5, o que pode ser rejeitado como *falso positivo*. Então, para melhorar a precisão, outros treinos são submetidos aos novos pesos do YoloV5 utilizando as 100 épocas em *Unity* e em seguida 100 épocas no dataset *Granada*, obtendo melhora nos resultados, aproximadamente 0.88 de precisão (Figura 4).

A confiança aumentou 0.2 (neste exemplo), identificando um possível verdadeiro positivo, o que é um avanço na detecção do modelo. Os novos pesos ainda foram capazes de detectar as armas em situação anteriormente não detectadas, como ilustrado na Figura 5.

Figura 4: Detecção de arma feita pelo YoloV5 através dos pesos de 100 épocas com 2 novos datasets

Figura 5: Comparação do frame obtido da análise do peso com 300 épocas (esquerda) e com 100 épocas (direita)

5.2 Detecção de grafite

Após início do treinamento, foi constatado uma *precisão* inicial baixa, menor de 0.1. Entretanto foi possível notar melhor desempenho na detecção dos *taggers* do que outras classes, provalmente fruto da capacidade prévia dos pesos padrões do *YoloV5l* em detectar humanos. Conforme avanço dos treinos (cerca de 70 épocas), o método foi capaz de aprender e aumentar sua precisão para cerca de 0.67 em relação a precisão média para todas as classes, tendo melhor desempenho em identificar *tagger* com 0.9 e pior em detectar grafites, próximo de 0.5. A Figura 6a apresenta o primeiro frame de pichação analisado neste trabalho.

Os primeiros testes em vídeos, não antes vistos pela ferramenta, foram realizados na época 70, quando esta possuía um média de *precision* geral de 0.67. A Figura 6b identifica a detecção da presença das 3 classes buscadas, marcando o *tagger* em rosa, *graffiti* em laranja e *spray* em vermelho.

O resultados indicam a possibilidade da detecção da ocorrência de pichação, entretanto trazem alguns alertas como falsos positivos em detectar quaisquer humanos como *taggers* e alguns objetos como *spray*, indicando a necessidade de ajustes. A principal ideia é apenas confirmar a situação de pichação quando a detecção for das 3 classes no mesmo frame e também com confiança superior a 0.7.

5.3 Detecção de fogo

Em relação à detecção de fogo, após treino da arquitetura utilizando *Tensorflow*, os resultados foram obtidos. Os pesos obtidos do *kaggle* foram submetidos a um vídeo de incêndio, não visto pela rede neural anterior e também não inserido no dataset. Diferente do

(a) Frame do primeiro vídeo.

(b) Frame do segundo vídeo.

Figura 6: Análise da pichação.

Yolo que desenha as *bounding boxes* do objeto, este método cerqueia a imagem com um borda vermelha quando há a presença de fogo. Assim, nos momentos iniciais de presença baixa de fogo, o método não o identifica (Figura 7).

Figura 7: Frame de vídeo com início de um foco de incêndio não detectado pelo método.

Após começar um incêndio maior, o método preenche as bordas com linhas vermelhas que indica a presença do fogo (Figura 8).

Figura 8: Frame de vídeo com foco de incêndio detectado.

Importante destacar que este trabalho está inserido em um projeto de pesquisa Geração de textos a partir das imagens no MIDAS vinculado à Secretaria de Segurança Publica da Bahia para detecção automatizada de incidentes em câmeras de vigilância com o intuito de alertar o Órgão responsável uma vez que o incidente tenha sido detectado. Cada incidente aqui ilustrado é direcionado a um órgão específico através de um classificador que juntamente com um léxico visa alertar o Órgão. Por exemplo, os Bombeiros em caso de incêndio, a Polícia Militar em caso de arma em punho. A utilização conjunta dos modelos visa permitir a identificação de diversos objetos necessários para a rotina da SSP em câmeras de vigilância.

6 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho compila o resultado da experimentação de técnicas na detecção de objetos em vídeos de câmeras de vigilância. As técnicas trazem indícios para alarmes na SSP na ocorrência destas 3 abordagens. Há necessidade de aprimoramento na precisão das detecções em imagens reais, já que muito datasets são canônicos (colocando os objetos em posição de destaque), o que não reflete as situações cotidianas. Etapas posteriores deste trabalho consistirão no aprimoramento da precisão e eficácia dos métodos com a introdução de novos datasets e associação de técnicas, além de incluir testes em streaming de imagens de câmeras, para avaliação do funcionamento em tempo real. Outro ponto a ser tratado é a junção destes métodos em uma ferramenta unificada, capaz de detectar diversos objetos a partir de um único sistema, o que ampliaria a usabilidade. A união das abordagens potencializa a ampliação da vigilância automatizada.

AGRADECIMENTOS

Esse estudo foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (PIBIC-AF 2020/2021), Projeto nº 19675 e Plano de Trabalho nº 38378 através da Universidade Federal da Bahia (UFBA).

REFERÊNCIAS

[1] Mirela Cazzolato, Letricia Avalhais, Daniel Chino, Jonathan Ramos, Jessica Souza, Jose Rodrigues Jr, and Agma Taina. 2017. FiSmo: A Compilation of Datasets from Emergency Situations for Fire and Smoke Analysis.

[2] Mohammed Ghazal, Carlos Vazquez, and Aishy Amer. 2007. Real-time automatic detection of vandalism behavior in video sequences. In *2007 IEEE International Conference on Systems, Man and Cybernetics*. 1056–1060. https://doi.org/10.1109/ICSMC.2007.4414038

[3] Jose L. Salazar González, Carlos Zaccaro, Juan A. Álvarez García, Luis M. Soria Morillo, and Fernando Sancho Caparrini. 2020. Real-time gun detection in CCTV: An open problem. *Neural Networks* 132 (2020), 297–308. https://doi.org/10.1016/j.neunet.2020.09.013

[4] Fransco Pérez Hernandez and Alberto Castillo Lamas. [n.d.]. Granada Pistol Dataset. https://github.com/ari-dasci/OD-WeaponDetection. Accessado: 6-04-2021.

[5] Khan Muhammad, Jamil Ahmad, Zhihan Lv, Paolo Bellavista, Po Yang, and Sung Baik. 2018. Efficient Deep CNN-Based Fire Detection and Localization in Video Surveillance Applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* PP (03 2018). https://doi.org/10.1109/TSMC.2018.2830099

[6] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. 2015. You Only Look Once: Unified, Real-Time Object Detection. *CoRR* abs/1506.02640 (2015). arXiv:1506.02640 http://arxiv.org/abs/1506.02640

[7] Adrian Rosebrock. 2019. Fire and smoke detection with Keras and Deep Learning. https://www.pyimagesearch.com/2019/11/18/fire-and-smoke-detection-with-keras-and-deep-learning/. [Online; Last accessed 10 Jun 2021].

[8] A. Warsi, M. Abdullah, M. N. Husen, M. Yahya, S. Khan, and N. Jawaid. 2019. Gun Detection System Using Yolov3. In *2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*. 1–4. https://doi.org/10.1109/ICSIMA47653.2019.9057329