

# Use of Deep Learning for Firearms Detection in Images

1<sup>st</sup> Guilherme Vinícius Simões Cardoso

Department of Electrical Engineering  
Federal University of Espírito Santo  
Vitória - ES, Brasil  
guilherme.v.cardoso@aluno.ufes.br  
cardoso.guivi@gmail.com

2<sup>nd</sup> Patrick Marques Ciarelli

Department of Electrical Engineering  
Federal University of Espírito Santo  
Vitória - ES, Brasil  
patrick.ciarelli@ufes.br

3<sup>rd</sup> Raquel Frizera Vassallo

Department of Electrical Engineering  
Federal University of Espírito Santo  
Vitória - ES, Brasil  
raquel@ele.ufes.br

**Abstract**—Demand for weapons has grown along with crime rates, a contemporary problem haunting countries. This has motivated scientists to devise solutions that can aid in public safety in general. This paper proposes the detection of firearms in images through convolutional neural networks, using the YOLO (You Only Look Once) object detector. To improve learning, YOLO was used to generate annotations in an unmarked database, integrating a new database. This proposal was evaluated in a database containing 608 images, in which 304 images had weapons. Experiments carried out indicated an accuracy of 89.15% and a sensitivity of 100.00%, surpassing results presented in the current literature. These results show that the proposed methodology can be applied for the detection of firearms in images.

**Index Terms**—YOLO, CNN, Images, Firearms

**Abstract**—A procura por armas tem crescido juntamente com os índices de criminalidade, um problema contemporâneo que assombra diversos países. Isto tem motivado cientistas a traçar soluções que possam auxiliar na segurança pública de maneira geral. Assim, neste trabalho é proposta a detecção de armas de fogo em imagens por meio de redes neurais convolucionais, utilizando o detector de objetos YOLO (You Only Look Once). Para melhorar o aprendizado, o YOLO foi utilizado para gerar anotações em um banco de dados não marcado, integrando um novo banco. Tal proposta foi avaliada em uma base de dados contendo 608 imagens, sendo 304 imagens com armas. Experimentos realizados apontam uma precisão de 89,15% e uma sensibilidade de 100,00%, superando resultados apresentados na literatura. Estes resultados mostram que a metodologia proposta pode ser aplicada para a detecção de armas de fogo em imagens.

**Index Terms**—YOLO, CNN, Imagens, Armas de fogo

## I. INTRODUÇÃO

No início do século XXI, o poder legislativo aprovou uma Lei que marcou o Brasil e dividiu opiniões. Em uma tentativa de controlar o crescimento acentuado dos índices de criminalidade, o poder legislativo aprovou, no final do ano de 2003, o Estatuto do Desarmamento (Doravante: ED) [1]. Responsável por definir regras e estabelecer diretrizes para o controle de armas no Brasil, a Lei nº 10.826/2003 trouxe uma série de critérios rigorosos que diz respeito ao acesso a armas, além de proibir o porte civil [2].

Estudiosos afirmam que o ED foi um “divisor de águas”, sendo um dos principais fatores que auxiliaram na redução dos

índices de crimes com armas de fogo [3]. Segundo o Instituto de Pesquisa Econômica Aplicada - IPEA, no ano de 2003, a taxa de homicídios por cem mil habitantes era de 29,14. Um pouco mais de 70% desses homicídios se davam pelas armas de fogo, chegando a uma taxa de 20,42. Dois anos após a aprovação da Lei, as taxas de homicídios caíram cerca de 10%, chegando a uma taxa de 26,13 homicídios no ano de 2005, e cerca de 70% desses homicídios ainda se davam pelas armas de fogo, chegando a uma taxa de 18,14 [4], [5].

Apesar disso, de acordo com o último fechamento apresentado pelo IPEA, por meio do Atlas da Violência [6], houve um novo crescimento das taxas de homicídios e um crescimento na proporção entre os crimes com armas de fogo. Entre 2011 e 2016, mais de 70% dos estados brasileiros apresentaram uma variação positiva, ou seja, um crescimento na variação de homicídios, fechando o ano de 2016 com uma taxa equivalente a 30,3 por cem mil habitantes. Desses homicídios, 71,1% se dão por armas de fogo, alcançando uma taxa de 21,6 no ano de 2016 [4]–[6].

Mudanças ocorridas no ED por meio de decretos pelo atual governo podem contribuir de maneira negativa e agravar ainda mais as taxas de homicídios por meio de armas de fogo [7]. Um caminho para auxiliar no combate desse problema é a detecção automática de armas de fogo por meio de imagens. Nos primeiros trabalhos relacionados a esse tema, é apresentado o uso de descritores de imagens projetados manualmente e técnicas de aprendizado de máquina para detectar armas de fogo. Recentemente, com o advento das redes neurais profundas, a identificação de objetos em cenas tem passado por relevantes avanços, sendo possível até realizar a segmentação de objetos em nível de pixel com uma razoável precisão [8].

Neste contexto, o presente artigo propõe uma abordagem para a detecção de armas de fogo em imagens utilizando redes neurais convolucionais (CNN - *Convolutional Neural Network*). A partir do uso do detector de objetos YOLO (You Only Look Once) [9] e baseado no trabalho realizado por Olmos *et al.* [10], será avaliado um método para a detecção de armas de fogo em imagens. Resultados experimentais apontam resultados superiores aos encontrados na literatura, com uma precisão de 89,15% e uma sensibilidade de 100,00%.

Para descrever e discutir a abordagem aqui proposta, este artigo está estruturado como segue: na Seção II são mencionados os trabalhos relacionados ao tema deste artigo; na Seção III são descritas as bases de dados, a rede *YOLO* e o método proposto; na Seção IV são apresentados os resultados experimentais e finalmente na Seção V são apresentadas as conclusões e trabalhos futuros.

## II. TRABALHOS RELACIONADOS

Os primeiros trabalhos na literatura que apresentam algumas abordagens para detectar armas de fogo utilizam imagens em escalas de cinza provenientes de ondas de raios-x. Gesick *et al.* [11] avaliam três abordagens separadas envolvendo *pattern matching*, transformada *Daubechies Wavelet* [12], e uma abordagem utilizando *SIFT - Scale Invariant Feature Transform* [13]. Xiao *et al.* [14] propuseram utilizar métodos de reconhecimento de objetos e de aprendizado de máquina, utilizando o extrator de características *Haar-Like* e o classificador *AdaBoost* em cascata. Tais abordagens necessitam de imagens específicas obtidas através de ondas de raios-x, sendo esta uma das principais limitações, além de não apresentar resultados precisos, devido à reação de todos os metais presentes com os *scanners* de raios-x.

Outros trabalhos utilizam métodos clássicos de visão computacional para detectar armas de fogo em imagens *RGB*. Como exemplo, os autores de [15] e [16] desenvolveram um *software* que é dividido em duas etapas, a primeira consiste em segmentar a imagem utilizando *K-means*, em seguida é utilizada a técnica *SURF - Speeded Up Robust Features* para detectar pontos de interesse. Halima *et al.* [17], de maneira similar, utilizam *SIFT* para obter características e *K-means* para obter os *clusters* necessários para o treinamento do classificador *SVM - Support Vector Machine*.

Os trabalhos apresentados até aqui são lentos e dependem de intervenção humana, além de utilizar métodos clássicos de visão computacional.

Olmos *et al.* [10] propuseram um “alarme automático de detecção de armas em vídeos usando *deep learning*”, que, segundo os autores, é o primeiro trabalho na literatura que utiliza CNN. Para tal, duas abordagens foram avaliadas: *Sliding Windows* e *Region Proposals*. As duas técnicas foram comparadas em um mesmo banco de dados, mas com distintos bancos de treinamento. A abordagem *Region Proposals*, implementada pelo detector de objetos *Faster R-CNN* atingiu melhores resultados, sendo avaliada em alguns vídeos.

Utilizando as bases de dados criada por Olmos *et al.*, Elmir *et al.* [18] compararam e avaliaram duas abordagens: *Fast R-CNN* e *MobileNet-CNN*. Seu objetivo principal foi avaliar o esforço necessário para utilizar as abordagens de maneira *online*. O modelo que apresentou melhores resultados foi o antecessor do *Faster R-CNN*, o *Fast R-CNN*.

## III. METODOLOGIA PROPOSTA

Nesta seção é descrita a metodologia proposta neste artigo para o problema de detecção de armas de fogo em imagens.

### A. Base de dados

As bases de dados utilizadas neste trabalho foram construídas por Olmos *et al.* e estão disponibilizadas em um repositório público<sup>1</sup>. A Tabela I descreve cada banco de dados contendo informações do número total de imagens, imagens com armas e imagens anotadas. As bases originais *BK4*, *WeaponS* e *Test set* foram renomeadas como DB-1, DB-2 e DB-T, respectivamente.

TABLE I: Características dos bancos de dados.

DB-	Base de Dados	Total img	# img com armas	# anot.
1	DB-1	9857	795	0
2	DB-2	3000	3000	3000
T	DB-T	608	304	0

A base de dados DB-1 contém 9857 imagens de resoluções variadas, divididas em 102 classes, sendo uma delas de armas, contendo 795 imagens. Entretanto, vale ressaltar que as imagens do banco DB-1 não possuem anotações a respeito das localizações dos objetos.

A base de dados DB-2 contém 3000 imagens de armas de resoluções variadas. Neste banco, as imagens contêm anotações a respeito dos objetos.

Finalmente, a base de dados DB-T é a base utilizada para teste. Contém 608 imagens de resoluções fixas de  $160 \times 120$  pixels, onde metade dessas imagens são de armas. O banco DB-T não possui anotações, mas as suas imagens foram organizadas da seguinte maneira: as 304 primeiras imagens (1-304) não são de armas e as 304 imagens restantes (305-608) são de armas. Algumas imagens de cada um dos três bancos de dados são apresentadas na Figura 1.

### B. YOLO

O detector de objetos *YOLO* é um algoritmo desenvolvido para processamento em tempo real, que faz parte de um *framework* para redes neurais construído na linguagem C, chamado de *Darknet* [9], [19]. Este detector utiliza uma única CNN, capaz de identificar simultaneamente múltiplas *bounding boxes* e a probabilidade destas marcações.

A Figura 2 mostra o modelo do *YOLO*, onde uma imagem de entrada é dividida em  $S \times S$  regiões, e cada região gera  $B$  *bounding boxes* atribuindo um valor de confiança e a probabilidade do objeto ser de uma determinada classe.

Neste trabalho foi usada a versão 2 do *YOLO* (*YOLOv2*). Esta versão utiliza o modelo de classificação *darknet-19*, onde são utilizados filtros  $3 \times 3$  para extração de características e filtros  $1 \times 1$  para reduzir os canais de saída, além de utilizar *pooling* global para previsões. Além disso, foi adicionada uma normalização de *batches* para estabilizar o treinamento, acelerar a convergência e regularizar o modelo.

A versão do *YOLO* usada possui seus pesos pré-treinados sobre o conjunto de dados *COCO Dataset* [20].

<sup>1</sup><https://github.com/SihamTabik/Pistol-Detection-in-Videos>



Fig. 1: Exemplos de imagens dos bancos DB-1, DB-2 e DB-T.

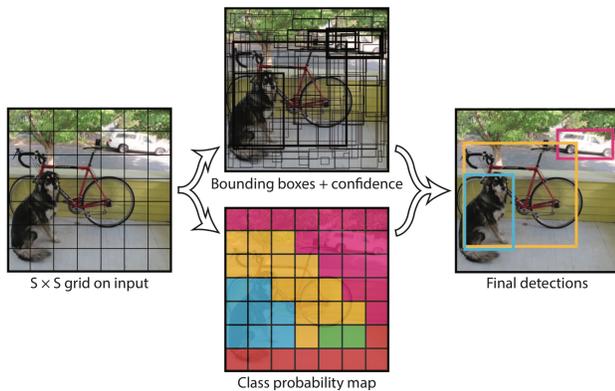


Fig. 2: Funcionamento da YOLO [21].

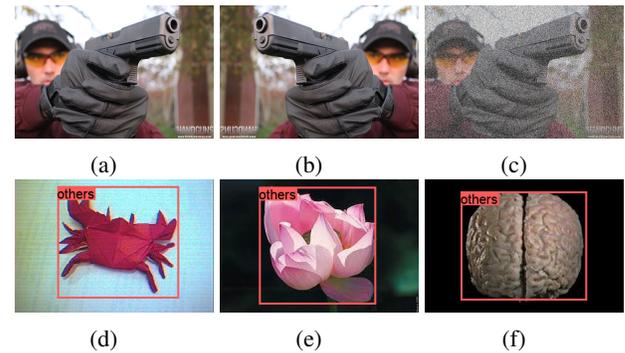


Fig. 3: (a) - Exemplo de imagem do banco DB-2; (b) - *Flip-Vertical* de a; (c) - ruído Gaussiano de a; e (d) - (f) apresentam exemplos de anotações criadas no banco DB-1.

### C. Método de treinamento

Para treinamento, foram avaliados os resultados do *YOLO* treinado em duas bases de dados - DB-2 e DB-3 - com algumas particularidades, explicadas a seguir.

O primeiro treinamento foi realizado com o banco DB-2, o mesmo banco criado e utilizado por Olmos *et al.* [10]. Tal treinamento foi necessário para comparar e avaliar as duas técnicas, resultando na utilização do *YOLO* neste trabalho, o que será discutido na próxima seção.

O segundo treinamento foi realizado com a base de dados DB-3, formada pela união das bases DB-1 e DB-2. Antes, porém, foram aplicadas duas técnicas de *Data Augmentation* para aumentar a quantidade de imagens do banco de dados DB-2: *Flip-Vertical* e ruído Gaussiano, como mostram as Figuras 3a - 3c. O ruído Gaussiano foi aplicado para cada imagem com desvio padrão igual a média da respectiva imagem e com média igual a 128. Este processo triplicou a quantidade de imagens com armas, proporcionando um total de 9000 imagens

com as suas respectivas anotações. Em seguida, as imagens de DB-2 foram adicionadas às imagens de DB-1, formando o banco de dados DB-3 com um total de 18062 imagens.

O *YOLO* necessita que as imagens dos conjuntos de treino e validação possuam as suas respectivas anotações para efetuar o treinamento da rede. Todavia, o banco de dados DB-1 não contém estas anotações, apesar de possuir imagens de armas, como mostra a Tabela I. Além disso, as imagens de armas contidas no banco DB-1 apresentam baixa resolução e o tamanho relativo das armas nas imagens ocupa quase toda a área útil em todas as 795 imagens. Por isso, tais imagens foram retiradas do banco, uma vez que o banco DB-1 será usado como amostras negativas da classe “pistol”. Após este passo, o *YOLO*, com os seus pesos pré-treinados sobre a base de dados *COCO Dataset*, foi executado sobre as imagens restantes da DB-1 para gerar anotações dos diferentes objetos existentes, os quais receberam o rótulo de “others”, como mostra as Figuras 3d - 3f.

Vale lembrar que o *YOLO* é *open source* e foi construído em linguagem C, um ponto bastante importante na criação das anotações, uma vez que o *YOLO* foi modificado para retornar as *bounding boxes*, pois, por padrão, retorna a probabilidade das classes via terminal. Ao final deste processo, anotações foram criadas para o banco DB-1. Entretanto, o *YOLO* gerou anotações baseados nos pesos pré-treinados do *COCO Dataset*, retornando anotações de 80 tipos de classes, onde foram modificadas para uma única classe (“others”). Em seguida, todas as imagens foram submetidas a uma inspeção visual com o auxílio de um *software*<sup>2</sup> *open source*, com o objetivo de retirar anotações criadas erroneamente e criar anotações onde o *YOLO* deveria detectar um objeto mas não o fez.

Por fim, ambos os bancos foram divididos aleatoriamente em dois conjuntos, sendo eles de treino e validação, em que o conjunto de treino representa 80% e o conjunto de validação 20%. Após isso, foi realizado *transfer learning* no *YOLO*, com a seguinte configuração dos hiperparâmetros: taxa de aprendizado de  $10^{-3}$ , um decaimento de 0,0005 e momento de 0,9, os mesmos valores utilizados por Redmon *et al.* [9]. Além disso, foi adicionado uma normalização de *batches* para estabilizar o treinamento, acelerar a convergência e regularizar o modelo.

#### IV. EXPERIMENTOS E RESULTADOS

Nesta seção são apresentadas as métricas de avaliação, os experimentos e os resultados obtidos.

##### A. Características de Implementação

A configuração da máquina utilizada nos experimentos foi:

- Sistema operacional Linux, distribuição Ubuntu 18.04 LTS;
- Processador Intel i7-8700K @3.70GHz;
- Memória RAM de 32GB DDR4;
- Placa de vídeo Nvidia Titan V, com 12GB de memória dedicada.

##### B. Avaliação dos Resultados e Discussão

Levando em consideração que todas as avaliações foram realizadas sobre o banco de testes DB-T e que cada imagem contém ou não uma arma, pode-se considerar o problema de detecção de armas como uma classificação binária. Para avaliar o desempenho da abordagem, foram utilizadas as seguintes métricas: Precisão (*Pr*), Sensibilidade (*Se*) e *F<sub>1</sub> Score* (*F<sub>1</sub>*).

A precisão avalia a capacidade do modelo em detectar corretamente as imagens que realmente possuam armas; a sensibilidade, também conhecida como *Recall*, avalia a capacidade do modelo em detectar todas as armas de fogo do banco de dados; e finalmente a medida *F<sub>1</sub> Score*, que é a média harmônica entre a precisão e a sensibilidade.

Todas as métricas supracitadas são apresentadas nas Equações (1) - (3), onde: *VP* são os verdadeiros positivos (número de imagens que o modelo detectou corretamente ao menos uma arma de fogo), *VN* são os verdadeiros negativos

(número de imagens nas quais o modelo corretamente não detectou armas de fogo), *FP* são os falsos positivos (número de imagens que o modelo erroneamente detectou ao menos uma arma de fogo), e *FN* são os falsos negativos (número de imagens nas quais o modelo erroneamente não detectou armas de fogo).

$$Pr = \frac{VP}{VP + FP} \times 100\% \quad (1)$$

$$Se = \frac{VP}{VP + FN} \times 100\% \quad (2)$$

$$F_1 = 2 \times \frac{Pr \times Se}{Pr + Se} \times 100\% \quad (3)$$

Utilizando os hiperparâmetros descritos na Seção III-C com os bancos DB-2 e DB-3, foi efetuada a avaliação dos resultados sobre o banco de teste DB-T. Calculando-se as métricas de *Pr*, *Se* e *F<sub>1</sub>*, os resultados atingidos treinando o *YOLO* com a base de dados DB-2 original e DB-3 são mostrados na Tabela II. Não obstante, nesta tabela há uma comparação entre os resultados adquiridos na literatura para este banco de dados (DB-T).

Elmir *et al.* [18] avaliaram duas abordagens utilizando as mesmas bases de dados, o *MobileNet-CNN* e o *Fast R-CNN*. Tais abordagens não apresentam resultados expressivos. As duas técnicas não foram capazes de detectar todas as armas do banco de teste e a precisão das armas detectadas foi inferior aos resultados obtidos por Olmos *et al.* [10].

Olmos *et al.* [10] obtiveram altos valores de precisão e sensibilidade, mas um elevado número de *FP*. Utilizando o algoritmo *Faster R-CNN* sobre a sua base de dados criada especificamente para esta técnica (DB-2), tal algoritmo foi capaz de detectar todas as armas do banco de teste (*Se* = 100,00%) e apresentou uma elevada precisão (*Pr* = 84,21%).

A primeira abordagem deste artigo buscou avaliar a eficiência do *YOLO* em detectar armas, comparadas com os resultados obtidos na literatura e utilizando as mesmas bases de dados. Tal abordagem utilizou o banco DB-2 para treinamento e validação e obteve resultados parcialmente superiores aos encontrados na literatura, como mostra a Tabela II. Resultados experimentais sobre o banco de testes DB-T, apontam uma elevada precisão (*Pr* = 95,10%), sendo explicada pela redução de falsos positivos, e uma alta sensibilidade (*Se* = 89,47%), sendo explicada pela redução dos verdadeiros positivos. Em resumo, o treinamento do *YOLO* com o DB-2 foi capaz de detectar quase todas as armas do banco de teste e obteve uma alta assertividade.

O processo de treinamento da primeira abordagem durou cerca de 3 horas, onde os resultados informados sobre o conjunto de teste foram obtidos após encontrar o melhor resultado sobre o conjunto de validação, o que demorou cerca de 40000 épocas de treinamento. O tempo médio de detecção de cada imagem de teste foi de 0,012 segundos, configurados com um *threshold* de 25%.

A primeira abordagem teve resultados satisfatórios mediante aos resultados encontrados na literatura, porém, foi observado

<sup>2</sup><https://github.com/tzualin/labelImg>

TABLE II: Resultados obtidos sobre o banco de teste DB-T.

Modelo	BD de treino	VP	FN	VN	FP	Pr. (%)	Se. (%)	F1 (%)
MobileNet-CNN [18]	DB-2	156	54	168	42	78,78	74,28	76,46
Fast R-CNN [18]	DB-2	232	72	248	56	80,76	76,31	78,37
Faster R-CNN [10]	DB-2	<b>304</b>	<b>0</b>	247	57	84,21	<b>100,00</b>	91,43
YOLOv2	DB-2	272	32	<b>290</b>	<b>14</b>	<b>95,10</b>	89,47	<b>92,20</b>
YOLOv2	DB-3	<b>304</b>	<b>0</b>	<b>267</b>	<b>37</b>	<b>89,15</b>	<b>100,00</b>	<b>94,26</b>

que a presença de uma única classe (“*pistol*”) no banco DB-2 contribui de forma negativa nos treinamentos, impedindo uma generalização mais efetiva. Buscando amenizar esta dificuldade, neste trabalho foi proposta a unificação dos bancos DB-1 e DB-2, o que deu origem ao banco intitulado de DB-3, o qual deu um equilíbrio entre as classes.

A segunda abordagem consiste no treinamento do *YOLO* com o banco DB-3. Esta abordagem apresentou uma alta sensibilidade (100,00%) e uma elevada precisão (89,15%), sendo superior aos trabalhos apresentados na literatura. Assim como o Olmos *et al.* [10], o *YOLO* foi capaz de detectar todas as armas do banco de teste e obteve um aumento de 5% na precisão.

A junção de um banco de dados contendo apenas armas e outro banco contendo imagens variadas forneceu à rede dados suficientes para melhorar seu desempenho em todas as métricas, na qual foi possível manter a quantidade de verdadeiros positivos e reduzir a quantidade de falsos positivos.

O processo de treinamento da segunda abordagem, durou cerca de 5 horas, e assim como na primeira abordagem, os resultados informados sobre o conjunto de teste foram obtidos após encontrar o melhor resultado sobre o conjunto de validação, o que demorou cerca de 170000 épocas de treinamento. O tempo médio de detecção de cada imagem de teste foi de 0,012 segundos, configurados com um *threshold* de 80%.

Vale lembrar que todos os resultados apresentados pela Tabela II foram calculados sobre o banco de teste DB-T. As 304 imagens de armas que compõe o banco, em sua grande maioria, ocupam grande parte da área útil das imagens e os tipos de armas variam entre pistolas e revólveres. Por outro lado, o tamanho relativo das armas do banco DB-2 nem sempre ocupa toda a área das imagens (Figura 1), como acontece no DB-T.

A Figura 4 apresenta a curva ROC (*Receiver Operating Characteristic Curve*) da metodologia proposta treinada com os dois tipos de bases de dados. Esta curva é um método gráfico para avaliar modelos de classificação [22]. A curva ROC avalia a dispersão da taxa de verdadeiros positivos  $\times$  taxa de falsos positivos em função do *threshold* aplicados na classificação. O ponto ótimo dessa curva é (0,1), onde a taxa de verdadeiro positivo é 1 e a taxa de falsos positivos é 0, ou seja, todas as classificações feitas pelo modelo estão corretas. Baseado nestas afirmações, pode-se observar que as duas abordagens apresentam elevados resultados de VP para baixos valores de FP, especialmente o método utilizando o banco DB-3, que alcança valores de VP iguais a 1 para baixos

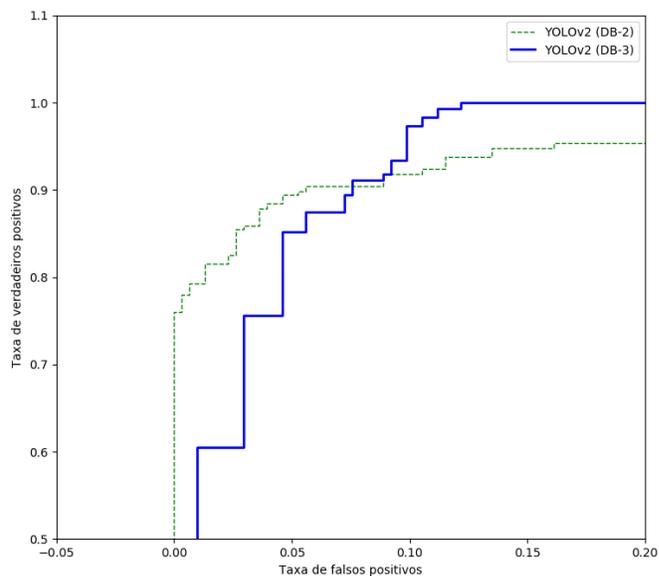


Fig. 4: Curva ROC obtida sobre o banco de teste DB-T.

valores de FP. Estes resultados indicam que o método utilizado para treinar a *YOLO* é promissor e mais adequado do que somente usar a DB-2.

A Figura 5 ilustra alguns exemplos de imagens rotuladas corretamente (VP) e outras rotuladas erroneamente (FP) como armas, resultantes do segundo experimento.

## V. CONCLUSÃO E TRABALHOS FUTUROS

O objetivo deste artigo foi propor um detector de armas de fogo em imagens utilizando CNN. Para tal, uma metodologia foi elaborada utilizando o detector de objetos *YOLO*, e avaliada em dois experimentos realizados com uma base de dados disponibilizada em um repositório público.

A primeira abordagem teve por objetivo avaliar a metodologia proposta com os resultados apresentados na literatura, onde o treinamento e a validação foram efetuadas apenas com amostras positivas de armas de fogo.

Nesta abordagem, o *YOLO* não foi capaz de localizar todas as armas do banco de teste, porém, este treinamento proporcionou uma alta precisão do modelo, chegando a 95,10%.

Em uma tentativa de aumentar a generalização do modelo, uma segunda abordagem foi realizada. Utilizando o *YOLO* para gerar anotações em imagens não marcadas e que representam amostras negativas de armas de fogo, somadas com a utilização de duas técnicas de *Data Augmentation* sobre as



Fig. 5: Exemplos de verdadeiros positivos e falsos positivos do segundo experimento.

amostras positivas, um novo banco foi elaborado, criando um equilíbrio entre amostras positivas e negativas.

Esta abordagem foi avaliada no mesmo banco de teste, onde superou os resultados apresentados na literatura. O equilíbrio gerado pela união dos bancos de dados proporcionou a detecção de todas as armas no conjunto de teste e uma elevada precisão do modelo.

Ambas as abordagens ainda apresentam uma quantidade de falsos positivos que interfere na precisão alcançada nos dois casos. Este é um ponto que deve ser melhorado, sendo uma das alternativas a utilização de um banco mais abrangente.

As duas abordagens apresentam resultados expressivos comparados à literatura, porém com bastante espaço para melhoras. O aumento da precisão é primordial, por isso, no futuro, técnicas que possam contribuir na redução dos falsos positivos vão ser implementadas e a construção de um novo banco será estudada assim como a utilização de versões mais recentes do YOLO.

#### AGRADECIMENTOS

O presente trabalho foi realizado com o apoio da CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, da FAPES - Fundação de Amparo a Pesquisa e Inovação do Espírito Santo através do Projeto 577/2018, assim como da NVIDIA Corporation através da doação da GPU Titan V para o processamento de imagens e *deep-learning*.

#### REFERENCES

[1] D. R. d. C. Cerqueira, *Causas e consequências do crime no Brasil*. Banco Nacional de Desenvolvimento Econômico e Social, 2014.

[2] Lei nº 10.826, de 22 de dezembro de 2003, Dispõe sobre registro, posse e comercialização de armas de fogo e munição, sobre o sistema nacional de armas – sinarm, define crimes e dá outras providências. ed., Brasil. Presidência da República. Casa Civil, Brasília, 2003.

[3] D. R. d. C. Cerqueira and J. M. P. d. Mello, “Menos armas, menos crimes,” 2012.

[4] “Atlas da violência: Taxa de homicídios,” <http://www.ipea.gov.br/atlasviolencia/dados-series/20>, Brasil. Instituto de Pesquisa Econômica Aplicada - IPEA, 2019, acesso: 22-01-2019.

[5] “Atlas da violência: Taxa de homicídios por armas de fogo,” <http://www.ipea.gov.br/atlasviolencia/dados-series/35>, Brasil. Instituto de Pesquisa Econômica Aplicada - IPEA, 2019, acesso: 22-01-2019.

[6] A. da Violência, “Ipea e fbsp,” Rio de Janeiro, 2018.

[7] Decreto nº 9.847, de 25 de junho de 2019, Regulamenta a lei nº 10.826, de 22 de dezembro de 2003, para dispor sobre a aquisição, o cadastro, o registro, a posse, o porte e a comercialização de armas de fogo e de munição e sobre o sistema nacional de armas e o sistema de gerenciamento militar de armas. ed., [http://www.planalto.gov.br/ccivil\\_03/\\_ato2019-2022/2019/decreto/D9785.htm](http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2019/decreto/D9785.htm), Brasil. Presidência da República. Casa Civil, Brasília, 2019.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[9] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” pp. 7263–7271, 2017.

[10] R. Olmos, S. Tabik, and F. Herrera, “Automatic handgun detection alarm in videos using deep learning,” *Neurocomputing*, vol. 275, pp. 66–72, 2018.

[11] R. Gesick, C. Saritac, and C.-C. Hung, “Automatic image analysis process for the detection of concealed weapons,” in *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*. ACM, 2009, p. 20.

[12] I. Daubechies, *Ten lectures on wavelets*. Siam, 1992, vol. 61.

[13] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[14] Z. Xiao, X. Lu, J. Yan, L. Wu, and L. Ren, “Automatic detection of concealed pistols using passive millimeter wave imaging,” in *2015 IEEE International Conference on Imaging Systems and Techniques (IST)*. IEEE, 2015, pp. 1–4.

[15] R. K. Tiwari and G. K. Verma, “A computer vision based framework for visual gun detection using harris interest point detector,” *Procedia Computer Science*, vol. 54, pp. 703–712, 2015.

[16] —, “A computer vision based framework for visual gun detection using surf,” in *2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, 2015, pp. 1–5.

[17] N. B. Halima and O. Hosam, “Bag of words based surveillance system using support vector machines,” *International Journal of Security and Its Applications*, vol. 10, no. 4, pp. 331–346, 2016.

[18] Y. Elmir, S. A. Laouar, and L. Hamdaoui, “Deep learning for automatic detection of handguns in video sequences,” in *3rd edition of the National Study Day on Research on Computer Sciences (JERI 2019)*, Saida, Algeria, April 27, 2019., ser. *CEUR Workshop Proceedings*, A. Amine, R. M. Hamou, M. Yahlali, and M. A. Boudia, Eds., vol. 2351. CEUR-WS.org, 2019. [Online]. Available: [http://ceur-ws.org/Vol-2351/paper\\_69.pdf](http://ceur-ws.org/Vol-2351/paper_69.pdf)

[19] J. Redmon, “Darknet: Open source neural networks in c,” <http://pjreddie.com/darknet/>, 2013–2016.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[22] R. Prati, G. Batista, and M. Monard, “Curvas roc para avaliação de classificadores,” *Revista IEEE América Latina*, vol. 6, no. 2, pp. 215–222, 2008.