

Detecção de placas de veículos com foco na proteção de dados pessoais

Bruno José Souza * Alessandro Zimmer ** Roberto Zanetti Freire *

* Programa de Pós-Graduação em Engenharia de Produção e Sistemas
Escola Politécnica, Pontifícia Universidade Católica do Paraná,
Rua Imaculada Conceição, 1155, 80215-901, Curitiba, Paraná, Brazil
(e-mail: bruno.jose@pucpr.edu.br, roberto.freire@pucpr.br).

** Center of Automotive Research on Integrated Safety Systems and
Measurement Area - CARISSMA, Ingolstadt, Bayern, Germany
(e-mail: alessandro.zimmer@thi.de)

Abstract: In recent years, there has been an increase in the development of systems related to data protection. Following the recent General Data Protection Regulation (GDPR), systems are being adapted to capture images without compromising the safety of users through improper disclosure. This paper intended to evaluate the performance of different versions of deep learning algorithms aimed at the detection of vehicle license plates. The research proposes the adaptation of one method to hide the license plates that were detected in images, thus providing the protection of personal data. The selected deep learning techniques were YOLO v3 (You Only Look Once) and YOLO v4. The training of the selected algorithms was performed with the THI License Plate Dataset (TLPD). The YOLO v3 algorithm reached an Average Precision (AP) equal to 98.68%, recall of 98.00%, and Average Intersection over Union (Average IoU) of 79.38%, being elected the best among the techniques compared in this study.

Resumo: Nos últimos anos, percebe-se um aumento no desenvolvimento de sistemas relacionados à proteção de dados. Seguindo o recente Regulamento Geral de Proteção de Dados (GDPR), os sistemas estão sendo adaptados para capturar imagens sem comprometer a segurança dos usuários por meio de divulgação indevida. Este trabalho teve como objetivo avaliar o desempenho de diferentes algoritmos de aprendizado profundo voltados a detecção de placas de veículos. A pesquisa propõe a adaptação de um método para ocultar as placas presentes nas imagens, proporcionando assim a proteção dos dados pessoais. As técnicas de aprendizado profundo selecionadas foram YOLO v3 (*You Only Look Once*) e YOLO v4. O treinamento dos algoritmos selecionados foi realizado com o *THI License Plate Dataset* (TLPD). O algoritmo YOLO v3 atingiu um *Average Precision (AP)* igual a 98,68%, *recall* de 98,00% e uma *Average Intersection over Union (Average IoU)* de 79,38%, sendo eleita a melhor entre as técnicas comparadas neste estudo.

Keywords: Vehicle license plate; Deep learning; Data protection; Object detection; YOLO.

Palavras-chaves: Placas de veículos; Aprendizado profundo; Proteção de dados; Detecção de objetos; YOLO.

1. INTRODUÇÃO

Os avanços tecnológicos trouxeram aos dispositivos melhorias na coleta, armazenamento e compartilhamento de dados. Este avanço da tecnologia é explorado de forma inteligente em vários cenários, como no campo da captura de vídeo em espaços públicos (Sreenu and Durai, 2019) e na identificação e rastreamento de pessoas em tempo real (Liu et al., 2018a). A facilidade no compartilhamento de dados, no entanto, possibilita que informações pessoais sejam utilizadas para outros fins que não as intenções originais, fazendo uso de imagens e dados pessoais de forma indevida (RIBARIC et al., 2016).

Nos últimos anos, existe um esforço para desenvolver sistemas de proteção de dados pessoais (Wachter and Mittels-tadt, 2019). De acordo com o GDPR (Regulamento Geral

de Proteção de Dados), o tratamento de dados sem consentimento só é permitido em situações onde o direito e a liberdade dos sujeitos são respeitados. Quando a aquisição de dados é realizada de forma anônima, as informações nas imagens devem ser insuficientes para o reconhecimento das pessoas (Goddard, 2017). A utilização de técnicas de visão computacional é uma forma de abordar este tema, pois tem a capacidade de classificar (Yu et al., 2017) e também estimar com precisão a localização de um determinado objeto presente em uma imagem (Zhao et al., 2019).

Levando em consideração a demanda de dados para aplicações no setor automotivo (Protzmann et al., 2019) e seguindo as diretrizes de proteção de dados, no contexto de tráfego de veículos, é interessante ter a disposição uma ferramenta que consiga detectar placas e ocultar essas informações das imagens.

Algumas técnicas de visão computacional foram utilizadas para realizar a detecção de placas de veículos desde então. Alguns desses métodos são: a detecção de bordas, conforme apresentado em (Marr and Hildreth, 1980); e métodos baseados em arquiteturas distintas de redes neurais neurais convolucionais (CNNs) (LeCun et al., 1995), como por exemplo a *Faster Region-based convolutional neural networks* (Ren et al., 2016) e a *You Only Look Once* (YOLO) (Redmon et al., 2016).

O objetivo deste artigo é avaliar o desempenho de diferentes versões do algoritmo de aprendizado profundo YOLO, destinado à detecção de placas de veículos com o objetivo de proteger informações pessoais. Dentro disso, é feita uma análise quantitativa dos resultados obtidos e a customização de um dos algoritmos para proteção de dados pessoais (ocultar a placa) presente nas imagens do conjunto de dados.

As principais contribuições deste trabalho são:

- (1) É proposta a utilização de um algoritmo de *deep learning* como abordagem para automatizar uma atividade normalmente realizada de forma manual, que seria a proteção de informação pessoal ocultando as placas de veículos;
- (2) Uma comparação de desempenho das versões do algoritmo YOLO v3 e YOLO v4 é realizada considerando métricas de avaliação de desempenho de sistemas de visão computacional estabelecidas na literatura.

O restante deste artigo está organizado da seguinte maneira: A seção 2 apresenta alguns trabalhos relacionados a detecção de placas de veículos, justificando a relevância do tema abordado neste trabalho. A seção 3 descreve resumidamente o conjunto de dados utilizado, os algoritmos de aprendizado profundo abordados, bem como as métricas de avaliação selecionadas. Na seção 4 apresentam-se o experimento e os testes realizados durante esta pesquisa. A discussão a respeito dos resultados é apresentada na seção 5, enquanto as conclusões e sugestões para trabalhos futuros são fornecidas na seção 6.

2. DETECÇÃO DE PLACAS DE VEÍCULOS

Na literatura especializada existem alguns diversos trabalhos que abordam a detecção de placas de veículos em diferentes aplicações, como por exemplo em (Kessentini et al., 2019) e (Wang et al., 2018) onde os autores utilizam técnicas de *deep learning* para realizar o reconhecimento dos caracteres presentes na placa. Nesta seção do artigo são apresentados alguns trabalhos relacionados à detecção de placas de veículos e algumas técnicas utilizadas pelos autores.

Lien et al. (2019) abordam uma arquitetura de aprendizado profundo do tipo YOLO v2, com o objetivo de realizar a detecção de placas de veículos. Uma rede ResNet (rede residual) foi incluída pelos autores para melhorar a precisão da detecção. O treinamento resultou em um total de 98,2% de precisão.

Selmi et al. (2020) apresentam em seu artigo um detector de placas de veículos para cenários complexos. Seu diferencial é a utilização de uma rede chamada MASK R-

CNN (*mask region convolutional neural network*) usada para detecção e segmentação das imagens. Neste artigo os autores se basearam em quatro base de dados de diferentes países, incluindo imagens em diferentes condições e de diferentes qualidades. Seu sistema apresenta uma taxa de precisão de 99,3% para a base de dados AOLP e 98,9% para a base de dados Caltech.

Silva and Jung (2020) apresentam em sua pesquisa um detector de placas de veículos onde o método proposto é identificar o veículo e a placa utilizando uma CNN duas vezes consecutivas, e então realizar o reconhecimento dos caracteres da placa utilizando uma segunda CNN. Este método foi testado com um *dataset* contendo placas de veículos do Brasil e da Europa, e durante os testes de desempenho conseguiu reconhecer as placas em 89,15% das vezes.

Jamtsho et al. (2020) apresentam um ALPR (*Automatic License Plate Recognition*) capaz de funcionar em tempo real utilizando uma rede do tipo YOLO v2. Em seu algoritmo, a detecção de veículos é realizada previamente para eliminar falsos positivos. Os autores então usam uma única rede neural convolucional onde verificam a existência da placa e se ela está localizada na região determinada como veículo. Com os testes realizados, os autores obtiveram 98,6% de precisão na detecção de placas.

3. MATERIAIS E MÉTODO

3.1 Base de Dados

A base de dados utilizada neste artigo é denominada *THI License Plate Dataset* (TLPD) (Chan et al., 2020). Este conjunto de dados é formado por imagens de veículos da Europa em diferentes cenários e condições climáticas. O TLPD contém um total de 17.105 imagens de carros, caminhões e motocicletas em várias condições de iluminação e resolução. As imagens foram extraídas de 39 vídeos diferentes feitos a 30 fps, em estacionamentos e ruas da cidade de Ingolstadt na Alemanha e alguns lugares da França.

Os vídeos foram gravados em três condições diferentes: dia, noite e neve. O fato de este banco de dados incluir imagens em ambientes com diferentes condições faz com que o algoritmo de treinamento funcione de forma mais robusta para diferentes cenários. Algumas imagens da base de dados serão apresentadas na seção de resultados, já com a detecção das placas.

O conjunto de dados foi separado em 67% para treinamento, 17% para validação e 16% para teste. A porcentagem de imagens por categoria pode ser verificada conforme Tabela 1.

Tabela 1. Percentual por categoria - TLPD.

Categoria	Percentual de imagens
Dia	60%
Noite	20%
Neve	15%
Motocicletas	5%

O número total de imagens para cada categoria pode ser consultado conforme Tabela 2.

Tabela 2. Distribuição das imagens.

Categoria	Número de imagens	Percentual
Treinamento	11.462	67%
Teste	2.881	17%
Validação	2.762	16%

3.2 YOLO aplicada ao problema de Detecção de Objetos

O algoritmo *You Only Look Once* (YOLO) (Redmon et al., 2016) é uma rede neural convolucional simples, porém rápida. A rede possui esse nome porque só precisa processar a imagem uma vez para classificar ou detectar o objeto em questão. O algoritmo YOLO funciona dividindo a imagem de entrada em uma estrutura $S \times S$, onde cada parte (célula) é responsável por detectar um objeto se ele tiver seu centro dentro da célula. Então, cada célula fornece um certo número de caixas delimitadoras e também uma pontuação para cada uma delas. Esta pontuação indica o quanto o sistema acredita que um objeto existe dentro da caixa delimitadora. Cada caixa delimitadora é rotulada com 5 valores: x, y, w, h e c , onde (x, y) representa o centro das coordenadas da caixa delimitadora, (w, h) representam a largura e altura da caixa delimitadora, respectivamente, e c representa a *IoU* entre a região da caixa delimitadora e a região real do objeto.

A rede YOLO v1 tinha 24 camadas convolucionais seguidas por 2 camadas totalmente conectadas, mas esta estrutura mudou conforme suas versões evoluíram. Desde sua segunda versão, apresentada em (REDMON and FARHADI, 2017), o conceito de normalização em lote já está presente e melhorou sua taxa média de precisão. A mudança mais significativa da terceira versão foi em seu extrator de características (REDMON and FARHADI, 2018). O algoritmo YOLO v2 usa um extrator chamado Darknet-19, que possui 19 camadas convolucionais, enquanto o YOLO v3 usa Darknet-53, com 53 camadas convolucionais e também algumas conexões simplificadas.

A YOLO v4 surgiu com o objetivo de melhorar os resultados de suas versões anteriores (BOCHKOVSKIY et al., 2020). Para isso, possui algumas mudanças significativas em sua arquitetura. A arquitetura do YOLO v4 é composta por uma estrutura CSPDarknet53, um módulo de *pooling* de pirâmide espacial adicional, um ramo PANet (Liu et al., 2018b), e o topo da estrutura é baseado na YOLO v3.

Neste estudo, conforme mencionado na introdução deste artigo, serão comparadas as duas últimas versões desta arquitetura, a YOLO v3 e a YOLO v4.

3.3 Métricas de Avaliação

Existem diversas maneiras de avaliar o desempenho de um algoritmo de visão computacional, onde as métricas são derivadas de informações obtidas a partir da matriz de confusão. As métricas usadas neste trabalho serão apresentadas a seguir nesta seção.

Uma detecção classificada como verdadeiro positivo (TP) é definida por $IoU > T$, onde T é um limite predefinido. Neste trabalho, a avaliação é feita para T igual a 0,5.

O *IoU* é o critério que fornece as informações de similaridade entre o objeto detectado e o objeto real.

É definido pela área de intersecção dividida pela união da área do objeto e a área detectada (RAHMAN and WANG, 2016). Já o *recall* é uma métrica que representa o número total de detecções corretas em porcentagem, é calculado dividindo as detecções classificadas como verdadeiros positivos, pela soma dos verdadeiros positivos e falsos negativos (SANTOS, 2020), conforme mostrado na equação 1.

$$Recall = \frac{TP}{TP + FN}. \quad (1)$$

A métrica *Precision* (equação 2) tenta representar o quão bom é o modelo, sendo definida pelo número de detecções corretas (verdadeiros positivos) dividido pelo número total de detecções (SANTOS, 2020).

$$Precision = \frac{TP}{TP + FP}. \quad (2)$$

Outra forma de realizar a avaliação é a curva *precision* \times *recall*, onde é possível obter a *Average Precision* (AP) e a *mean average precision* (mAP). A *Average Precision* é definida como a área sob a curva, e o mAP é definido como a média da *Average precision* para todas as classes (CHAN, 2019). O *Average IoU* é o critério que fornece as informações de similaridade entre o objeto detectado e o objeto real. A equação 3 mostra como o *Average IoU* é calculado. A Figura 3 apresenta os resultados obtidos.

$$Average IoU = \frac{\sum IoU > T}{TP + FP}. \quad (3)$$

Por fim, a última métrica considerada neste estudo foi o tempo de execução, que representa o tempo que a técnica precisa para detectar a placa do veículo (em ms) (SAYAWA, 1999).

4. EXPERIMENTOS

4.1 Configuração dos Algoritmos

O treinamento dos algoritmos selecionados foi realizado em um computador conforme especificações da Tabela 3.

Tabela 3. Especificações técnicas do computador.

Sistema operacional	Windows 10 Enterprise 64 bits
Processador	Intel i9-10900F CPU @2.80 GHz
Memória RAM	128 GB DDR4
Placa de vídeo	NVIDIA GeForce RTX 2080 Ti

A tabela 4 mostra quais parâmetros foram definidos para cada um dos algoritmos. Para definir o número de épocas de treinamento, é sugerido utilizar o resultado da multiplicação do número de classes a serem treinadas por 2.000, porém, é indicado que o número mínimo de épocas seja igual a 6.000 (BOCHKOVSKIY et al., 2020).

Uma vez que o objetivo deste experimento envolve o treinamento de apenas 1 classe número de épocas de treinamento adotado foi de 6.000. Outro parâmetro ajustado foi a quantidade de filtros das camadas convolucionais que precedem a camada YOLO nos arquivos de treinamento.

Tabela 4. Parâmetros de treinamento.

Parâmetro	YOLO v3	YOLO v4
<i>Batch</i>	64	64
Tamanho da imagem	416 × 416	416 × 416
<i>Momentum</i>	0.9	0.949
<i>Decay</i>	0.0005	0.0005
<i>Learning rate</i>	0,001	0,001
<i>Anchor boxes</i>	9	9
Épocas de treinamento	6.000	6.000
Tempo de treinamento	8 horas	7,5 horas

O valor padrão é 255 mas sugere-se que este número seja alterado de acordo com a equação 4 (Bochkovskiy, 2020), o que resultou em um total de 18 filtros.

$$Filtros = 3 \times (\text{Número de classes} + 5). \quad (4)$$

5. RESULTADOS

A avaliação de desempenho dos algoritmos foi realizada nos dados de teste, e para isso 4 métricas foram utilizadas como critérios de avaliação: *recall*, *Average Precision*, *Average IoU* e tempo de execução.

O *recall* foi avaliado para um limite de *IoU* igual a 0,5. Isso significa que para uma detecção ser classificada como positiva verdadeira, seu *IoU* deve ser maior ou igual a esse valor. O gráfico na Figura 1 mostra os valores de *recall* para *IoU* de 50%.

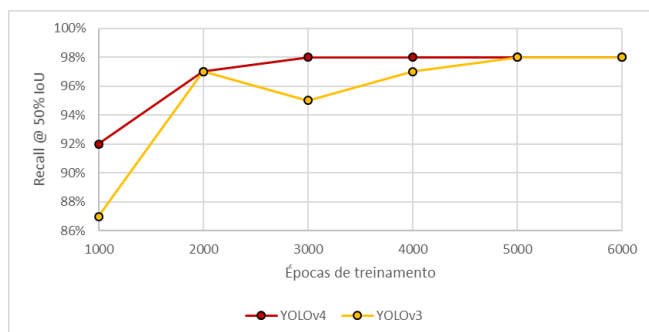


Figura 1. Resultados de *recall*.

Verifica-se que os resultados do *recall* são semelhantes para os dois algoritmos, onde ambos alcançaram 98%.

A *Average Precision* é uma métrica baseada no cálculo da área sob a curva *precision* × *recall*. A Figura 2 mostra os valores de *AP* para um limite de *IoU* igual a 50%.

Verifica-se que para um limite de 50% de *IoU*, os algoritmos alcançam resultados próximos, onde melhor resultado foi alcançado pela YOLO v4 sendo 98,79% contra 98,71% da YOLO v3.

Os resultados indicam que os algoritmos conseguiram atingir um *Average IoU* superior a 75%, sendo o melhor resultado 78,30% obtido pela YOLO v3. A Figura 4 apresenta uma comparação dos resultados obtidos para o tempo de execução.

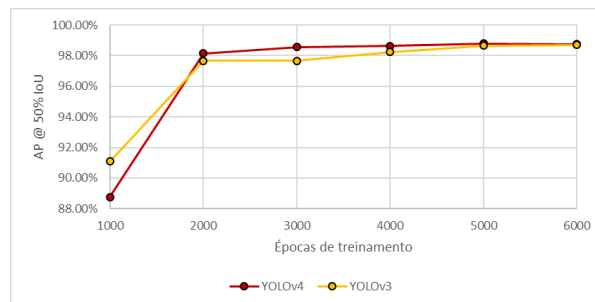


Figura 2. Resultados de *Average Precision*.

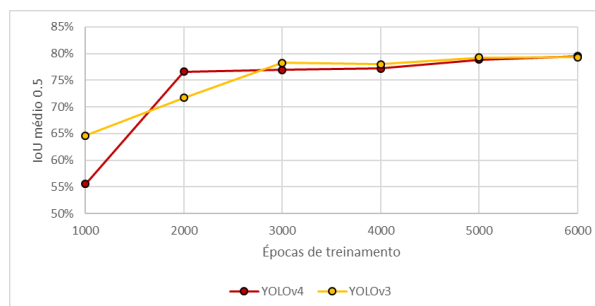


Figura 3. Resultados de *average IoU*.

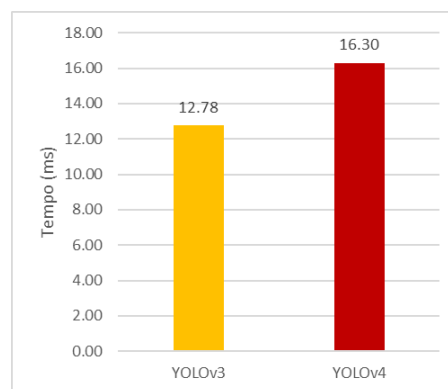


Figura 4. Tempo de execução.

Os resultados indicam que a YOLO v3 necessita de um tempo de execução menor, sendo 12,78 milissegundos contra 16,30 milissegundos da YOLO v4.

O algoritmo YOLO v3 apresentou melhores resultados para as métricas de avaliação, por este motivo foi selecionado para realização dos testes. Para realizar a proteção das informações pessoais, é necessário que a área da placa fique oculta, em função disso foi feita uma customização na função para que a caixa delimitadora que fosse preenchida.

As Figuras 5, 6 e 7 fazem parte do conjunto de testes, e são utilizadas para demonstrar a funcionalidade do algoritmo. A Figura 5 mostra os resultados para detecção de placas na dianteira e traseira de veículos em diferentes ângulos.



Figura 5. Detecção da placa em diferentes ângulos.

Pode-se verificar que o algoritmo é capaz de detectar placas em diferentes tipos de veículos, como utilitários, caminhões e motocicletas, não se restringindo apenas a veículos de passeio. Para as imagens classificadas como noturnas, foram verificadas três diferentes condições de iluminação: iluminação do tipo *flash*, que é fornecida pelo celular no momento da captura da imagem, iluminação ambiente, e imagens com ausência de iluminação.

A Figura 6 apresenta o funcionamento do algoritmo em imagens noturnas, enquanto a Figura 7 mostra a capacidade do algoritmo de realizar a detecção de placas de veículos, mesmo quando as placas não estão totalmente visíveis nas imagens. Isso significa que, se as imagens forem parcialmente cortadas ou houver um objeto na frente da área da placa, o algoritmo ainda pode realizar a detecção.

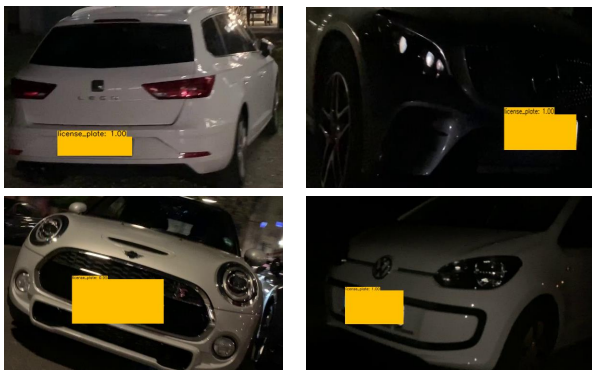


Figura 6. Detecção de placas em imagens noturnas.

Também foram feitos testes para verificar se o algoritmo tem capacidade de cumprir sua função quando há uma condição climática de neve no ambiente ou mesmo cobrindo parte da placa do veículo.

Algumas imagens com placas de veículos brasileiros também foram testadas para ver se o algoritmo teria a capacidade de detectá-las por apresentarem tamanho e padrão de cores diferentes. Os resultados apresentados na Figura 8 mostram que a YOLO v3 foi capaz de realizar a detecção.



Figura 7. Detecção de placas com área visível reduzida.



Figura 8. Detecção de placas de veículos brasileiros.

6. CONCLUSÃO

Nesta pesquisa, foi realizado o treinamento e a avaliação de desempenho de algoritmos de aprendizado profundo voltados à detecção de placas de veículos para proteção de dados pessoais. Para isso, as técnicas YOLO v3 e YOLO v4 foram selecionadas para a etapa de implementação. Essas técnicas foram selecionadas levando-se em consideração seu tempo de execução, que torna os algoritmos capazes de serem implementados em um sistema embarcado e funcionar em tempo real.

Para realizar a avaliação de desempenho dos algoritmos, foram escolhidas as seguintes métricas: *Average Precision*, *recall*, *Average IoU* e tempo de execução. Ao analisar o desempenho das técnicas verifica-se que ambos os algoritmos apresentam resultados similares, porém o algoritmo YOLO v3 leva uma vantagem considerável quando analisado o tempo de execução, e por este motivo foi utilizado na realização de testes.

Os testes indicaram que o algoritmo é robusto o suficiente para se adequar a diferentes circunstâncias. Durante o treinamento, também foram observados alguns casos específicos de dificuldades encontradas pelo algoritmo para realizar a detecção, como nos casos em que existem adesivos no veículo que combinados com condições específicas de iluminação podem causar uma falsa detecção.

Com o intuito de desenvolver um sistema que realize a proteção de dados e opere em tempo real, é possível sugerir alguns passos para a continuidade deste projeto:

- (1) Realizar testes com redes mais compactas como as versões YOLO *Tiny*, com o objetivo de melhorar o desempenho em termos de tempo de execução;
- (2) Incluir um maior número de imagens com caminhões e motocicletas, visando um melhor desempenho do algoritmo para essas categorias de veículos;
- (3) É interessante considerar a inclusão de imagens em cenários mais abrangentes, contendo um maior número de veículos para verificar se isso influencia no aprendizado do algoritmo;
- (4) Testar diferentes arquiteturas de redes e promover uma comparação entre os algoritmos;

7. AGRADECIMENTOS

Os autores agradecem a Fundação Araucária através do projeto PRONEX 042/2018 pelo financiamento desta pesquisa. Os autores agradecem ao CARRISMA (*Center of Automotive Research on Integrated Safety Systems and Measurement Area*) por disponibilizar a base de dados utilizada nesta pesquisa.

REFERÊNCIAS

- Bochkovskiy, A. (2020). Yolo v4, v3 and v2 for windows and linux. URL <https://github.com/AlexeyAB/darknet>.
- BOCHKOVSKIY, A., WANG, C.Y., and LIAO, H.Y.M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- CHAN, L.Y. (2019). *Automatic Blurring of License Plates and Faces for European GDPR 2016/679 Compliance using Transfer Learning in R-CNNs*. Ph.D. thesis, Technische Hochschule Ingolstadt - Germany.
- Chan, L.Y., Zimmer, A., da Silva, J.L., and Brandmeier, T. (2020). European union dataset and annotation tool for real time automatic license plate detection and blurring. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 1–6. IEEE.
- Goddard, M. (2017). The eu general data protection regulation (gdpr): European regulation that has a global impact. *International Journal of Market Research*, 59(6), 703–705.
- Jamtsho, Y., Riyamongkol, P., and Waranusast, R. (2020). Real-time bhutanese license plate localization using yolo. *ICT Express*, 6(2), 121–124.
- Kessentini, Y., Besbes, M.D., Ammar, S., and Chabbouh, A. (2019). A two-stage deep neural network for multi-norm license plate detection and recognition. *Expert systems with applications*, 136, 159–170.
- LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.
- Lien, C.C., Chien, Y.C., Teng, F.Y., and Yang, C.C. (2019). Deep license plate recognition in ill-conditioned environments with ill-conditional data augmentation. In *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, 1–7. IEEE.
- Liu, G., Liu, S., Muhammad, K., Sangaiah, A.K., and Doctor, F. (2018a). Object tracking in vary lighting conditions for fog based intelligent surveillance of public spaces. *IEEE Access*, 6, 29283–29296.
- Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018b). Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8759–8768.
- Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167), 187–217.
- Protzmann, R., Hübner, K., Ascheuer, N., Bauknecht, U., Enderle, T., Gebhard, U., Raack, C., and Witt, A. (2019). Large-scale modeling of future automotive data traffic towards the edge cloud. In *Photonic Networks; 20th ITG-Symposium*, 1–3. VDE.
- RAHMAN, M.A. and WANG, Y. (2016). Optimizing intersection over union in deep neural networks for image segmentation. In *International symposium on visual computing*, 234–244. Springer.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788.
- REDMON, J. and FARHADI, A. (2017). Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7263–7271.
- REDMON, J. and FARHADI, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6), 1137–1149.
- RIBARIC, S., Ariyaeeinia, A., and Pavesic, N. (2016). De-identification for privacy protection in multimedia content: A survey. *Signal Processing: Image Communication*, 47, 131–151.
- SANTOS, B.S. (2020). *Metodologia Baseada em KDD para a classificação de mulheres fumantes quanto ao consumo de cigarros industrializados*. Ph.D. thesis, Pontifícia Universidade Católica do Paraná.
- SAYAWA, M. (1999). Dicionário de informática e internet: inglês/português. São Paulo: Nobel.
- Selmi, Z., Halima, M.B., Pal, U., and Alimi, M.A. (2020). Delp-dar system for license plate detection and recognition. *Pattern Recognition Letters*, 129, 213–223.
- Silva, S.M. and Jung, C.R. (2020). Real-time license plate detection and recognition using deep convolutional neural networks. *Journal of Visual Communication and Image Representation*, 71, 102773.
- Sreenu, G. and Durai, M.S. (2019). Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, 6(1), 48.
- Wachter, S. and Mittelstadt, B. (2019). A right to reasonable inferences: Re-thinking data protection law in the age of big data and ai. *Colum. Bus. L. Rev.*, 494.
- Wang, J., Huang, H., Qian, X., Cao, J., and Dai, Y. (2018). Sequence recognition of chinese license plates. *Neurocomputing*, 317, 149–158.
- Yu, S., Jia, S., and Xu, C. (2017). Convolutional neural networks for hyperspectral image classification. *Neurocomputing*, 219, 88–98.
- Zhao, Z.Q., Zheng, P., Xu, S.t., and Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11), 3212–3232.