UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE DO CAMPUS ARARANGUÁ
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Gabriel Assunção Domene

Abordagem Computacional para Detecção Automatizada por Imagem do Uso de Cinto de Segurança em Condutores baseado em Redes Neurais Convolucionais

Araranguá
2020

Gabriel Assunção Domene

Abordagem Computacional para Detecção Automatizada por Imagem do Uso de Cinto de Segurança em Condutores baseado em Redes Neurais Convolucionais

Trabalho de Conclusão do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde do Campus Araranguá da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.
Orientador: Prof. Antônio Carlos Sobieranski, Dr.

Araranguá
2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Domene, Gabriel Assunção
Abordagem Computacional para Detecção Automatizada por Imagem do Uso de Cinto de Segurança em Condutores baseado em Redes Neurais Convolucionais / Gabriel Assunção Domene ; orientador, Antônio Carlos Sobieranski, 2020.
32 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá, Graduação em Engenharia de Computação, Araranguá, 2020.

Inclui referências.

1. Engenharia de Computação. 2. Visão computacional. 3. Detecção de objeto. 4. Redes neurais convolucionais. I. Sobieranski, Antônio Carlos. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. III. Título.

Gabriel Assunção Domene

Abordagem Computacional para Detecção Automatizada por Imagem do Uso de Cinto de Segurança em Condutores baseado em Redes Neurais Convolucionais

Este Trabalho de Conclusão foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computaçãoe aprovado em sua forma final pelo Curso de Graduação em Engenharia de Computação.

Araranguá, 10 de Dezembro de 2020.

Documento assinado digitalmente
Fabricio de Oliveira Ourique
Data: 15/12/2020 18:39:02-0300
CPF: 916.167.860-00

Prof. Fabrício De Oliveira Ourique, Dr.
Coordenador do Curso

Banca Examinadora:

Documento assinado digitalmente

Antonio Carlos Sobieranski
Data: 16/12/2020 10:10:55-0300
CPF: 005.305.809-77

Prof. Antônio Carlos Sobieranski, Dr.
Orientador

Documento assinado digitalmente

Anderson Luiz Fernandes Perez
Data: 15/12/2020 13:56:21-0300
CPF: 020.803.459-58

Prof. Anderson Luiz Fernandes Perez, Dr.
Avaliador
Universidade Federal de Santa Catarina

Documento assinado digitalmente

Rodrigo Vinicius Mendonca Pereira
Data: 16/12/2020 10:46:00-0300
CPF: 028.663.779-07

Prof. Rodrigo Vinícius Mendonça Pereira, Dr.
Avaliador
Universidade Federal de Santa Catarina

Prof. Fábio Rodrigues De La Rocha, Dr.
Avaliador Suplente
Universidade Federal de Santa Catarina

Abordagem Computacional para Detecção Automatizada por Imagem do Uso de Cinto de Segurança em Condutores baseado em Redes Neurais Convolucionais

Automated Seat Belt Detection in Vehicle with External Images based on a Convolutional Neural Network Approach

Gabriel Assunção Domene * Antônio Carlos Sobieranski †

2020, Setembro

Resumo

O desenvolvimento sustentável de qualquer cidade inteligente depende diretamente vários fatores, dentre estes, um bom planejamento urbano de modo a utilizar de forma otimizada das rotas de fluxo veiculares e com segurança. No entanto, uma maior capacidade de vazão implica em um maior número de veículos circulantes nas ruas, que por conseguinte, querer uma melhoria significativa nos serviços de manutenção e sinalização, assim como a conscientização dos condutores quanto às normas e uso do cinto de segurança. Não obstante, um maior corpo técnico é requerido para dar suporte às situações anômalas, assim como fiscalizar a malha viária e punir os infratores. No que tange o uso do cinto de segurança, mecanismo essencial para assegurar redução da mortalidade em acidentes de trânsito, a fiscalização é trabalhosa, lenta e suscetível a erros, realizada por meio da inspeção humana pelo fiscal de trânsito. Neste contexto, o presente estudo busca apresentar uma abordagem computacional para a detecção automatizada do uso do cinto de segurança, utilizando-se de imagens capturadas a partir das vias de fluxo, como uma ferramenta auxiliar no monitoramento, prevenindo a infração das leis e reduzindo custos. Para a abordagem proposta, Redes Neurais Convolucionais foram utilizadas como reconhecedores, treinadas a partir de um *dataset* especificamente construído a partir de imagens de veículos, no que se considera o cenário ideal para inspeção. Os resultados experimentais apresentaram uma precisão média de 90,87% para um conjunto de 3000 imagens para o dataset utilizado, demonstrando a viabilidade técnica em construir sistemas de apoio à decisão para a fiscalização no trânsito.

Palavras-chaves: Visão Computacional. Cinto de segurança. Detecção de objeto. Redes Neurais Convolucionais.

*gabriel.domene@grad.ufsc.br
†a.sobieranski@ufsc.br

Abordagem Computacional para Detecção Automatizada por Imagem do Uso de Cinto de Segurança em Condutores baseado em Redes Neurais Convolucionais

Automated Seat Belt Detection in Vehicle with External Images based on a Convolutional Neural Network Approach

Gabriel Assunção Domene * Antônio Carlos Sobieranski †

2020, Setembro

Abstract

The sustainable development of any smart city relies directly on different factors, among them, a good urban planning as a way to optimize the vehicle route flow with safety. However, a higher flow rate capacity implies directly in a higher number of vehicles moving around the roads, what next, requires a significant better mainitance and signs, also as well as awarness of drivers in respect of laws and use of the seat belt. Besides that, more people working in the technical team is required to support anomalus situation, inspect the road network and punish offenders. Regarding the use of the seat belt, essential mechanism to assure the reduction of mortalities in road accidents, the inspection is laborious, slow and error prone, if done by human traffic inspector. In this context, the current study aims to present a computational approach for automatic detection in the seat belt, using captured imagens from road flow as a tool to aid in the monitoring, preventing the law offend and reducing costs. For the approach presented, Convolutional Neural Networks were used as detectors, trained on top of a specific dataset built of vehicles images, in what is considered an ideal scenario for inspection. The experimental results shows a mean average precision (mAP) of 90,87% for a 3000 images dataset, showing the technical viability in building systems capable of aiding decision inspection in traffic.

Key-words: Computer Vision. Seat-belt. Object detection. Convolutional Neural Network.

*gabriel.domene@grad.ufsc.br
†a.sobieranski@ufsc.br

1 Introdução

De acordo com o relatório de segurança em estradas, disponibilizado pela Organização Mundial da Saúde em 2018, o número de mortes provenientes de acidentes fatais no trânsito atingiu um total de 1.35 milhões no ano de 2016 (WHO, 2020). Entretanto esta não é a única métrica preocupante do relatório, existe também as vítimas de acidentes que sofrem com ferimentos e até mesmo graves sequelas permanentes, impactando suas vidas e elevando os custos de emergências médicas como um todo. Este tipo de fatalidade ocupa o topo do ranking de número de mortes em jovens adultos, aqueles que estão na faixa de 5 até 29 anos de idade, e o oitavo lugar no ranking geral de todas as faixas etárias (THEOFILATOS, 2017).

Analisando mais profundamente, nota-se que diversos fatores contribuem na geração destes números de mortes e acidentes. Entretanto existe um número também igual de fatores e práticas que buscam evitar estes acidentes em veículos, sendo que um em específico é o cinto de segurança (URI, 2007). Estatísticas sugerem que o uso correto da ferramenta pode fazer com que os números decaiam em até 50% para passageiros dos bancos da frente e 25% para passageiros do banco traseiro (WHO, 2020). Usualmente, os governos endossam políticas e leis de uso para o dispositivo, onde globalmente, 161 países possuem algum tipo de legislação e 105 destes são considerados padrões modelo na prática de uso. Porém, ressalta-se que apenas 7% destes países com práticas modelo são de países em desenvolvimento. O Brasil, que se enquadra nesta última categoria é considerado um país de rank 7, onde 0 é o pior rank e 10 o mais alto, nas leis de uso de cinto de segurança. Opostamente, o mesmo não se reflete no pensamento da população brasileira, que apesar das duras leis de trânsito e campanhas de incentivo, não utilizam o dispositivo (MANDACARU et al., 2017), resultando em 21% dos passageiros frontais sem cinto e 50% dos passageiros traseiros também, em um cenário de 93 milhões de veículos registrados.

No âmbito da fiscalização, os responsáveis pelo monitoramento desta atividade geralmente são policiais designados para beira de estradas e rodovias com o objetivo de realizar paradas periódicas e barreiras de afunilamento de trânsito para checagem dos motoristas. Uma prática ineficiente já que o policial não pode e muito menos consegue fiscalizar a totalidade de carros dependendo do fluxo da via (WANG; WAN; YUAN, 2018). Portanto, o mesmo pode assim deixar passar pessoas que estão infringindo a lei. Desta maneira, além de ser uma abordagem fraca é muitas vezes perigosa, complexa e de alto custo para ser desenvolvida estritamente por humanos, além de não ser contínua (24/7). Adicionando um sistema de monitoramento automatizado em vídeo centrado em um operador pode ser usado como auxílio de pista na decisão futura dos policiais que irão fazer a abordagem. Desse modo, tal sistema de video monitoramento pode elevar consideravelmente o nível de segurança de trabalho e diminuir possíveis erros de percepção da pessoa responsável pela fiscalização do método tradicional (FERNáNDEZ-SANJURJO et al., 2019).

Com o advento das tecnologias de reconhecimento de padrões por imagem, abordagens computacionais tem sido propostas com o intuito de identificar determinados tipos de padrões em imagens ou vídeos (WANG et al., 2020)(Kim et al., 2016). Tais soluções podem ser utilizadas para detectar automaticamente a presença ou ausência do cinto de segurança por parte do condutor (GUO et al., 2011)(LE et al., 2017), e enviar um sinal/alerta para o operador do sistema anteriormente mencionado, auxiliando na filtragem dos diferentes pontos de atenção, assim como múltiplas câmeras em paralelo. Estas tecnologias tem se comportado como eficientes ferramentas de apoio à decisão no processo de fiscalização

pelo agente de trânsito, podendo serem utilizadas para reduzir o esforço braçal do agente atuante e diminuindo a tendência ao erro que aparece durante o processo, pois a utilização de múltiplas câmeras permite um melhor controle de diferentes pontos de atenção no mesmo instante de tempo. Tais tecnologias, no entanto, carecem de precisão devido a utilização de métodos clássicos de visão computacional, que apresentam baixa taxa de generalização (KOTCHERGENKO; LOPES; COMUNELLO, 2015). Haja visto que, a presença de fatores externos como posicionamento das câmeras, iluminação presente no ambiente, o tempo registrado numa determinada hora e local ou até mesmo o clima prolongado da região podem interferir permanentemente na situação observada. Estes fatores alteram constantemente as premissas iniciais de ambiente, técnica adotada comumente em projetos de métodos clássicos.

Com o objetivo de mitigar os problemas e limitações apresentadas pelas abordagens clássicas citadas anteriormente, o presente trabalho aborda a detecção automatizada do uso de cinto de segurança em condutores veiculares utilizando redes neurais convolucionais, como um mecanismo auxiliar confiável para sustentar decisões de fiscalizações. A utilização das CNN's tem se mostrado extremamente promissoras como um solucionador geral de problemas devido a sua capacidade de generalização, foco o qual se insere a presente pesquisa. Desta forma, as imagens submetidas à abordagem proposta obtiveram uma taxa de acerto na ordem de $\approx 90.87\%$ para um dataset construído especificamente a partir de 3000 imagens ($\approx 97\%$ para treino). Os resultados experimentais demonstram a viabilidade da solução proposta implementada por métodos não tradicionais como as redes neurais convolucionais, trazendo cada vez mais a tecnologia para um patamar superior, de forma a melhorar condições de segurança em veículos e para usuários do sistema de trânsito em geral.

2 Trabalhos Correlatos

Uma revisão da literatura demonstra que algumas abordagens computacionais podem ser aplicadas para detecção automática da presença ou para ausência de cinto de segurança. De uma forma geral, duas categorias de métodos podem ser claramente observadas na literatura: (1) métodos clássicos e (2) métodos baseados em alguma abordagem conexionista (redes neurais clássicas ou convolutivas). Dentro do âmbito das abordages clássicas, pode-se notar aplicações baseadas em busca de características, tais como detectores de borda, especialmente Canny, a tradicional transformada de Hough, máquina de vetores de suporte (SVM), classificadores em cascata (*cascade classifiers*) como Haar, dentre outras técnicas clássicas que podem ser aplicadas isoladamente ou de forma híbrida (GONZALEZ; WOODS, 2008). Do outro lado das técnicas possíveis, nota-se o retorno de métodos baseados em redes neurais, especialmente impulsionado pelo advento de redes neurais convolucionais (CNN). CNN's já são aplicadas em diversos outros domínios de aplicação como robótica, agricultura, medicina, tornando-as abordagens práticas e eficientes na produção de resultados (Ren et al., 2017). CNN's estão sendo implementadas cada vez mais em diferentes áreas que antes não se existiam aplicações, tais como em sistemas de vigilâncias, e para cinto de segurança não poderia ser diferente.

Entretanto, a primeira forma correspondendo às abordagens clássicas são altamente suscetíveis a erros por variáveis introduzidas no ambiente, como distorção de forma, intensidade de iluminação ou tamanho de objeto. Portanto, a necessidade de técnicas que possam ultrapassar esta barreira e apresentar maior generalidade e boa taxa de precisão são essenciais para novas aplicações.

Dos métodos encontrados na literatura, mais especificamente baseados em abordagens clássicas ou *machine learning*, destacam-se:

- Huiwen Guo com detecção de cinto de segurança baseado em imagens: O autor propõe uma detecção baseada em regiões, onde primeiramente busca a localização do motorista através de proporções do carro para em seguida analisar o motorista propriamente. O alvo da pesquisa é o cinto de segurança e ele é encontrado por detecção de borda, através de manipulação do espaço HSV as bordas são encontradas e comparadas com a norma internacional GB14167-2006. Após um filtro com Hough Lines e uma verificação pessoal adotada os resultados obtidos para 100 exemplos são: 81% corretamente verificados, 17% verificados erroneamente e 2% faltantes na detecção (GUO et al., 2011).

- Dian Yu com detecção do cinto de segurança baseado na orientação do gradiente: Similar ao anterior, os métodos utilizados buscam seguir um fluxo de detecção partindo da placa, detecção do para-brisa do veículo, detecção da face do condutor para enfim analisar o possível cinto de segurança e sua avaliação. Para avaliar o gradiente, algumas manipulações na imagem são feitas para melhorar a visualização sem afetar o contexto original. Por fim o autor utiliza Sobel para calcular os valores obtidos. Os resultados finais são: 61 corretos, 16 falsos e acurácia de 79.2% para cintos afivelados e 117 corretos, 19 falsos e acurácia de 86% para cintos não afivelados (YU; ZHENG; LIU, 2013)

- Wei Li com detecção de cinto baseado em classificador em Cascata Adaboost: A detecção do veículo é feita utilizando primariamente o Adaboost que busca a região do para-brisa e na sequência o motorista. Uma região de interesse para o cinto é montada pela proporção da posição comumente do motorista e computado o mapa de gradiente da região aplicando Sobel. Para refinar o passo anterior o autor aplica Canny seguido de Hough line para cálculo final e avaliação 34% na métrica de *recall* (LI et al., 2013).

- Beilei Xu com uma abordagem de aprendizado de máquina para detecção de ocupantes em veículos: Apesar do título do trabalho ser relacionado com aprendizado de máquina, as técnicas envolvidas não chegam propriamente em redes neurais mas sim em máquinas de vetores de suporte. A abordagem seguida é similar aos outros trabalhos, localização de região de para-brisa e regiões de interesse, neste em específico não busca o cinto propriamente mas sim o motorista. Após essa coleta de descritores os autores agregam eles em uma técnica chamada de Fisher Vector para em seguida introduzir em um algoritmo de máquina de vetor de suporte. Os resultado deste trabalho são bastantes contundentes, com uma acurácia na faixa de 93% até 96%, incluindo diferentes cenários de iluminosidade, tempo e característica de película de para-brisa (XU et al., 2014).

- Xun-Hui Qin com uma aplicação de monitoramento eficiente para detecção de cinto de segurança em veículos: Diferente de alguns trabalhos citados, o autor utiliza descritores locais de textura: Haar-like features primariamente para depois aplicar técnicas histogramas de gradientes orientados (HoG) juntamente com integrais matemáticas para cálculo de pixels. A segunda etapa consiste na aprendizagem a partir do método de AdaBoost para treinar um classificador coerente com o problema. Um pós processamento é feito para melhorar a precisão de algumas classificações, aplicando supressão não máxima e informações de contexto (QIN et al., 2014).

- Rafael M. Kotchergenko com reconhecimento de passageiro frontal com cinto de segurança: Busca refinar outro trabalho já apresentado por Xue, Yifei e Xuye (2012) adicionando novas etapas de reconhecimento. Separa em cinco etapas denominadas de: Subtração de background, reconhecimento do automóvel, reconhecimento do para-brisa, reconhecimento do passageiro frontal, reconhecimento do cinto de segurança. Separando por região e aplicando a transformada de Hough. As linhas de características são buscadas e comparadas com valores pré-definidos. Os resultados obtidos são positivos para 75% dos casos com a presença do passageiro frontal (KOTCHERGENKO; LOPES; COMUNELLO, 2015).

Do outro lado das técnicas, existem as baseadas em abordagens conexionistas, tais como redes neurais e suas variantes convolutivas. Apesar de nem todas apresentadas seguirem estritamente o único uso da rede, a implementação delas como parte de pré-processamento ou de refinamento já evidenciam o quão potente são e quanto podem impactar em seus resultados finais. Vale ressaltar que dentro das redes neurais, as abordagens com CNN são quase que unanimidades quando o assunto é imagem, e dentro destes trabalhos pode-se elencar:

- Bin Zhou com Detecção de cinto de segurança baseado na aprendizagem em imagem utilizando *Salient Gradient*: Nesta proposta, que utiliza métodos clássicos porém refina em suas últimas etapas com redes neurais, a extração inicial é feita por detecção de bordas com Canny e logo após a similaridade das bordas é avaliada por *Salient Gradient Map* para seleção das regiões. Com o intuito de remover possíveis interferências carregadas, um método de seleção baseado em região é implementada para suprimir possíveis ruídos e criar um vetor de características para treinar uma rede neural de função de base radial para detecção de cinto de segurança. Os resultados apresentados e avaliados via *cross validation* apresentam 84,3% de identificações corretas e 15.7% identificações erradas em um conjunto de 200 exemplos (ZHOU et al., 2018).

- Yanxiang Chen em Detecção de cinto de segurança em rodovias através de câmeras de segurança utilizando CNN e SVM: Os autores trabalham com a premissa da elaboração de uma rede neural convolucional capaz de extrair as características desejadas. Basicamente há três diferentes caminhos de detecção: uma para o cinto, outra para o para-brisa e por fim o veículo. Essas informações são compartilhadas no fim de suas camadas para contextualizar a informação e produção da saída. Desta maneira, disposto de um resultado preliminar os autores aplicam máquina de vetores de suporte para também pós-processar esse resultado e eliminar regiões incorretas. Os resultados obtidos são de 87% objetos corretamente identificados, 9% falsamente identificados e apenas 4% não identificados (CHEN et al., 2017).

- Wu Tianshu com uma implementação e detecção de cinto de segurança utilizando uma FPGA: A detecção de veículos é realizada através do algoritmo de detecção implementado na YOLO, entretanto por ser embarcado dentro de uma FPGA, algumas alterações são feitas para trabalhar dentro do limite de hardware. As métricas obtidas são relativamente boas por se tratar de uma FPGA, atingindo uma acurácia de 81% (TIANSHU et al., 2019).

- Jing Yongquan com um design para aceleração de GPU para detecção de cinto de segurança: Este artigo em específico não se trata somente da precisão de detecção

utilizando redes. Os autores buscam otimizar os processos das três etapas de detecção: detecção de veículo com Deconv-SSD, detecção de para-brisa com Squeeze-YOLO e detecção de cinto com FCN. As duas primeiras métricas não são tão relevantes no contexto da pesquisa, entretanto a última etapa realizada apresenta uma acurácia de 94% para o algoritmo produzido (YONGQUAN et al., 2019).

Categoria	Ano	Autor	Técnica	Dataset	Precisão
Clássicos	2011	(GUO et al., 2011)	Sobel	100	81%
Clássicos	2013	(YU; ZHENG; LIU, 2013)	Sobel	213	79,2%
Clássicos	2013	(LI et al., 2013)	Sobel + Canny + Hough Lines	12K	34% Recal
ML	2014	(XU et al., 2014)	SVM	93K	93%
ML	2014	(QIN et al., 2014)	Haar + Adaboost + SVM	200	-
Clássicos	2015	(KOTCHERGENKO; LOPES; COMUNELLO, 2015)	Hough Transform	32	75%
CNN	2017	(CHEN et al., 2017)	CNN + SVM	-	87%
Clássicos + RN	2018	(ZHOU et al., 2018)	Canny + Salient + RBF	200	84,3%
CNN	2019	(TIANSHU et al., 2019)	QNN + BNN	-	81%
CNN	2019	(YONGQUAN et al., 2019)	SSD + Squeeze Yolo + FCN	-	94%
CNN	**2020**	**Abordagem proposta**	**Yolo**	**436**	**90,87%**

Em virtude dos fatos mencionados, uma clara distinção se levanta sobre as técnicas a se utilizar, separando-as em duas classes majoritárias. As clássicas como baseada em características (bordas, transformada Hough, histograma de orientação de gradientes) e aprendizado de máquina (máquina de vetores de suporte, adaboost) mencionados em parágrafos anteriores com algumas leves modificações e/ou otimizações. A segunda grande área, denominada como moderna para efeitos de separação, como redes neurais e rede neural convolucional, que estão se tornando tendência e se diferenciando em diversos ramos de arquiteturas de aprendizado profundo (LECUN; BENGIO; HINTON, 2015) e diferentes *framework*s de rede, como (REDMON et al., 2015) e (REDMON; FARHADI, 2018). Partindo da literatura disponível, diversas metodologias de abordagens computacionais podem ser aplicadas para detecção automática ou para falta de cinto de segurança. No geral, os métodos na literatura conhecidos podem serem categorizados em métodos clássicos e métodos baseados em redes neurais.

3 Fundamentação Teórica

3.1 Aprendizado de Máquina

Segundo (SHALEV-SHWARTZ; BEN-DAVID, 2014), humanos baseiam seu aprendizado no senso comum para filtrar conclusões aleatórias e sem sentido, enquanto máquinas incubidas de tarefas de aprendizagem devem possuir princípios nítidos e bem definidos que protegerão o programa de atingir conclusões incorretas. O desenvolvimento de tais princípios é um objetivo central da teoria de aprendizado de máquina. De acordo com o seu livro, (GERONN, 2019) classifica as diferentes categorias de aprendizado baseadas em: treino feito com ou sem supervisão humana, se é possível aprender ao longo da execução e se o trabalho executado é feito comparando dados novos com dados antigos/detecção de padrão para construção de modelo preditivo.

Especialmente na parte supervisionada existem outras definições que separam os diferentes aprendizado como: (1) aprendizado supervisionado, (2) aprendizado não supervisionado, (3) aprendizado parcialmente supervisionado e (4) aprendizado por reforço, sendo as duas primeiras opções as mais comuns. A primeira opção será o conceito aproximado dentro da pesquisa, devido ao conjunto de dados trabalhado ser composto por imagens, logo, um par de exemplo e anotação é utilizado como entrada e percorrido dentro da rede.

Aprendizado supervisionado é uma sub categoria de aprendizado de máquina. Neste tipo de modelo o sistema contará com o auxílio de um conjunto de dados previamente estabelecido e com suas anotações correspondentes. Segundo regras gerais para se considerar efetivo o método, o objeto de aprendizado é provido do correto (ou aproximadamente correto) valor da função para diferentes entradas, o qual altera sua representação de função para tentar se igualar ao resultado provido inicialmente. Ao longo da etapa de treino, o sistema irá se basear em anotações do conjunto para guiar a evolução do modelo e alteração de parâmetros de rede.

Para contextualização de aplicação real, a classificação de e-mails é um exemplo que pode ser treinado para diferenciar conteúdos não desejados (spam) de conteúdos verídicos, pois o algoritmo irá treinar de acordo com as anotações prévias e futuramente classificar novos e-mails em uma dessas opções de classes (SHALEV-SHWARTZ; BEN-DAVID, 2014).

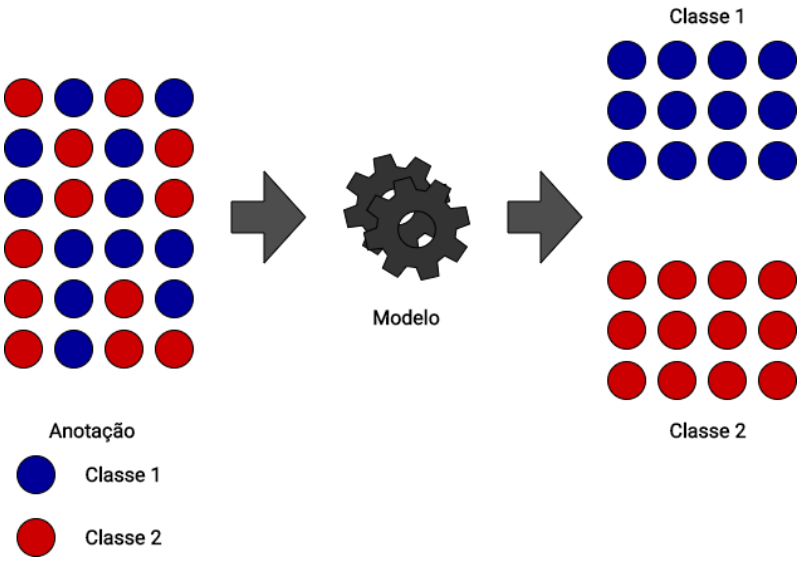

Figura 1 – Ilustração pertinente ao aprendizado supervisionado, na imagem, os círculos azuis e vermelhos representam as duas classes do problema a ser solucionado. O modelo é categorizado como supervisionado devido às anotações disponibilizadas para ajudar a modelagem do sistema em separar corretamente ou o mais próximo do correto as classes.

Fonte: Próprio autor

Em sistemas de treino com dados de imagens, o conjunto de dados é composto de um par de arquivos contendo a imagem propriamente e um arquivo de texto. Dentro deste arquivo, as anotações de início e fim da região em que o objeto se encontra visualmente são anotadas, assim a rede tem como gabarito a localização dos objetos e pode alterar seus parâmetros para minimizar o erro da predição. Outros algoritmos podem serem utilizados em tarefas de classificações, segundo (SHALEV-SHWARTZ; BEN-DAVID, 2014) *AdaBoost* (diminutivo de *Adaptive Boosting*) produz uma hipótese que é uma combinação linear de hipóteses mais simples. Resumidamente, o algoritmo se baseia na família de classes obtidas para compor uma predição linear em cima dos casos. Sendo que uma grande vantagem desta implementação é poder controlar o *tradeoff* entre aproximação e estimativa de erros variando um único parâmetro.

Uma abordagem diferente das já apresentadas é a técnica realizada pelo algoritmo *Principal Component Analysis (PCA)* que visa reduzir a dimensionalidade através de transformações lineares, devido ao grande número de informações que podem surgir a partir de imagens que são representadas por uma matriz n x m. Duas matrizes auxiliares são produzidas para se encarregar dessas transformações, uma matriz W é encontrada e denominada de matriz de compressão e uma matriz U (aproximadamente) denominada de matriz de recuperação. Dentro do PCA, o objetivo principal é reduzir a distância absoluta entre os vetores de origem e recuperação.

3.2 Redes Neurais Convolucionais

Segundo (GOODFELLOW; BENGIO; COURVILLE, 2016), redes convolucionais, também conhecidas por redes neurais convolucionais ou pelo seu acronômio CNNs, são um tipo especializado de redes neurais para processamento de dados, que possuem um padrão e/ou topologia de rede. Exemplos como imagens que podem serem interpretadas por uma matriz 2D de pixels. CNNs vem sendo extremamente úteis em aplicações práticas, e como seu próprio nome já indica, a rede emprega o uso de uma operação matemática chamada convolução.

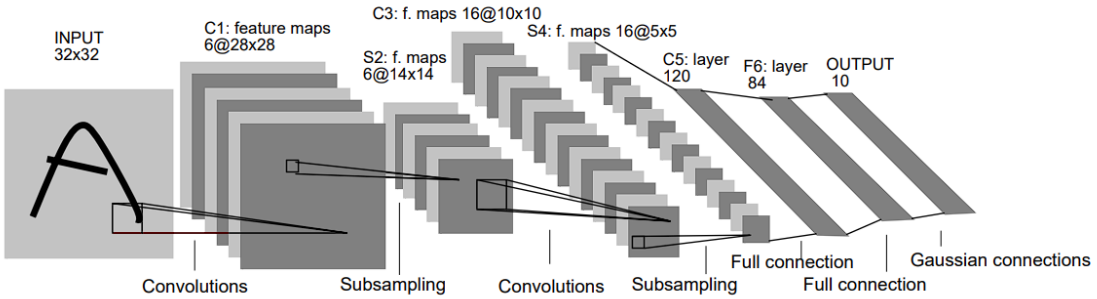

Figura 2 – Arquitetura da LeNet-5, um exemplo aplicado para reconhecimento de caracteres usando a rede neural convolucional mencionada. Cada camada é a representação de um mapa de características da rede. As camadas são dispostas como Cx para *convolutional layers*, Sx para *sampling layers* e Fx para *fully-connected* sendo x o index da camada.

Fonte: (LECUN et al.,)

Apesar de ser um termo bastante utilizado em matemática e cálculo, a convolução usada em CNN's nem sempre corresponde as mesmas definições usadas em outras áreas. Por isso, deve-se pensar mais como uma técnica para preservar compartilhamento de parâmetros, representações equivalentes e interações esparsas. Desta forma, o mecanismo anteriormente mencionado tende a obter uma melhor otimização do problema já que essas operações buscam salvar características mais importantes como bordas utilizando menos pixels do que o conteúdo original.

A equação 1 demonstra o modelo mais comum utilizado em bibliotecas de aprendizado de máquina.

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n), \tag{1}$$

onde K é o kernel aplicado sobre a imagem de entrada I, e i j a coordenada espacial em um domínio planar, m e n são as coordenadas espaciais do kernel proposto "deslizado"pela imagem.

3.2.1 Operação de convolução

Na etapa de convolução, o kernel (K) fixado na camada ativa irá realizar uma passagem pela matriz que consiste na imagem. Geralmente o kernel possui um tamanho 3x3 ou 5x5, ou como citado por (GERONN, 2019), "Convolução é uma operação matemática que desliza uma função sobre a outra medindo a integral de sua multiplicação pontual. A operação anterior altamente relacionada com as transformadas de Fourier e Laplace, e bastante utilizada em processamento de sinais.". Citado também por (DEY, 2018), a operação de convolução se caracteriza por "uma operação realizada em cima de duas imagens, uma sendo o *input* e a outra sendo a máscara (também chamado de *kernel*) como um filtro na imagem de *input*, produzindo uma imagem como *output*".

Uma das características desta etapa é a diminuição do tamanho original do dado, por exemplo, uma entrada de tamanho 5x5 com uma convolução de kernel 3x3 produzirá um *output* 3x3 pois o kernel sobreposto da matriz de entrada deslizará através das posições possíveis produzindo um novo tamanho de saída, preservando as características alvo do kernel. Porém ela não se restringe apenas a esse uso, segundo (DEY, 2018) a convolução aplica um filtro de propósito geral em cima da imagem de *input*, podendo atingir diferentes resultados a partir de diferentes *kernels* como suavização, aumentar nitidez ou relevo em operações, ou ainda atuando em operações de detecção de gradientes como bordas.

No caso específico de redes neurais convolucionais ao fim das *conv layers*, existe uma função de ativação responsável pela ativação do neurônio em questão. Passo importante do algoritmo pois é a etapa que define muitas vezes se uma informação importante será propagada ou não.

3.2.2 Pooling

Outra técnica de redução espacial presente em modelos de CNN é o *pooling*. Segundo (GOODFELLOW; BENGIO; COURVILLE, 2016) o agrupamento ajuda a tornar a representação aproximadamente invariante à pequenas rotações da entrada. A invariância para a rotação significa que se girarmos a entrada por uma pequena quantidade, os valores da maioria das saídas combinadas não mudam. Basicamente, preservar as características encontradas nas camadas fazem com que a rede atinja um nível melhor de eficiência na resolução do problema proposto.

A Figura 3 é um exemplo de *max pooling*, o kernel buscará o maior valor dentro dos números possíveis, entretanto existem outras técnicas de *pooling* que podem levar em conta uma média dos valores para seleção entre outros.

3.3 Aumento de dados

Um modelo relevante só é atingido em um de seus diversos aspectos, se o dado proposto a ser analisado possa ser reproduzido de maneira generalista para outros exemplos nunca visto. Em conjunto de dados (datasets) mais comuns e bem estruturados ao longo do tempo isto não é um grande problema, pois a comunidade tende a construir uma base sólida e diversa, com números atingindo 60 mil exemplos (MNIST dígitos manuscritos) ou então conjuntos gigantescos com 1.5 milhão de imagens de diferentes categorias (ImageNet)

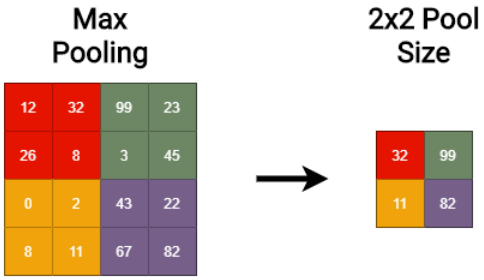

Figura 3 – Resultado da operação de *pooling* em cima de uma matriz inicial onde cada cor destacada representa a área limite da seleção do *max pooling*, resultando na diminuição da matriz inicial para a matriz de saída.

Fonte: Próprio autor

entre outros datasets. Aumento de dados vem sendo uma solução particularmente boa para um problema de classificação específico: detecção de objeto, imagens possuem um grande número de dimensões devido a sua resolução o que incluem uma enorme variedade de fatores diferentes, muitos dos quais são facilmente simulados (GOODFELLOW; BENGIO; COURVILLE, 2016).

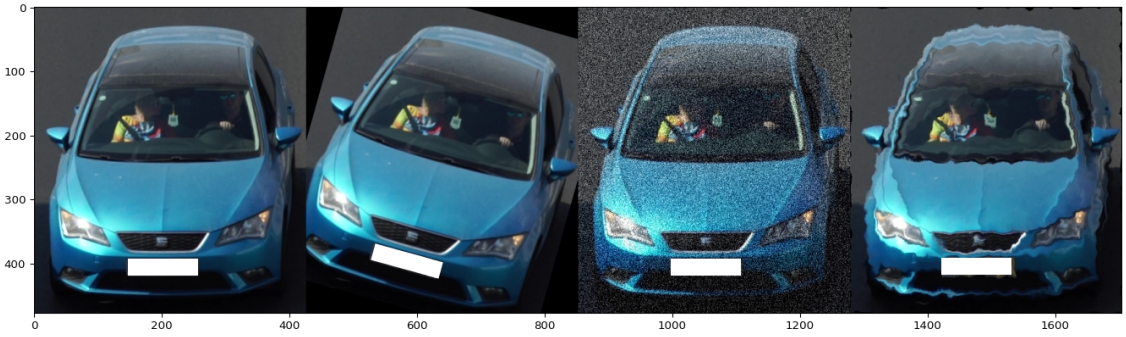

Figura 4 – Aumento de dados, da esquerda para direita: Imagem original, imagem rotacionada, imagem com ruído inserido e imagem com distorções

Fonte: Próprio autor

O ponto para utilização de aumento de dados é sobrepor esta falta através de transformações aplicadas de maneira realista e que não alterem o sentido do dado original reduzindo assim o possível *overfitting*, algo que poderia acontecer se fosse aplicado rotações de 180 graus no dataset de dígitos do MNIST, ou a exemplo, modificando formas geométricas tal como o número 6 e 9.

3.4 Frameworks para Redes Neurais Convolucionais

Com o advento e popularização de redes neurais convolucionais, muitos desenvolvedores começaram a criar diferentes *frameworks* para simplificar o uso de conceitos e aumentar a produtividade de modelos convolucionais. No geral, alguns se destacam pela sua facilidade de uso e comunidade ativa na resolução de problemas.

3.4.1 FAST AI

Diretamente da sua documentação, o time de desenvolvimento do FAST.AI[1] se intitula como uma biblioteca de *deep learning* para fácil e rápida implementação. O desenvolvimento é pensado justamente na facilidade para se construir códigos e com esse pensamento, o FAST.AI se apresenta como uma arquitetura de API é montada para utilização de diferentes métodos.

A implementação é feita de tal maneira que o conjunto de técnicas sejam desacopladas e de alto nível para o usuário. A sua flexibilidade é grande devido a liberdade da linguagem Python e de outra biblioteca chamada PyTorch.

Como citado por (HOWARD; GUGGER, 2020), FAST.AI é organizado ao redor de dois pilares: acessível e de rápida produtividade em diferentes áreas como visão computacional, processamento de linguagem natural, modelos tabulares e séries temporais. Por isso pode-se dividir o projeto em três diferentes camadas de *API*: Alto nível, médio nível e baixo nível. A camada mais alta é voltada para projetos iniciais com o uso de métodos baseados em *deep learning* já existentes. Desta forma, a primeira abstração visa mitigar erros comuns de iniciantes, devido ao fato das classes implementadas já serem integradas com certos métodos. A camada intermediária já disponibiliza alguns métodos mais desacoplados e livres, permitindo que a última camada (baixo nível) customize e desenvolva métodos para camada de nível superior.

3.4.2 TensorFlow

Outra opção de desenvolvimento robusto para aprendizado de máquina. A plataforma do TensorFlow[2] oferece diferentes camadas de abstração para implementação de código que vão desde as mais simples até tarefas mais complexas em produção. Algumas diferenças para os outros *framework*s são as API's de alto nível, características próprias especificamente elaboradas para facilitar a comunicação e leitura. O TensorFlow se baseia na API do Keras para definir e treinar redes neurais, permitindo assim um rápido desenvolvimento de código e prova de conceito.

A comunidade do *framework* é de código aberto permitindo contribuições constantes e suporte ao longo do tempo, nela é possível também a utilização de unidades de processamento gráfico para aumentar o processamento de dados. O projeto do TensorFlow pode ser considerado um dos maiores e mais completos dentro do campo de *machine learning*. Além da linguagem Python, o projeto possui desenvolvimento em JavaScript, Swift e implementação em dispositivos móveis como Android, iOS e Raspberry Pi. Todas essas áreas refletem em uma grande gama de possibilidades de projetos, desde visão computacional até processamento de texto.

3.4.3 YOLO

Segundo (REDMON et al., 2016), a implementação do sistema de detecção (REDMON; FARHADI, 2018) é feita em cima do *framework* da *Darknet* e se diferencia por ser uma abordagem totalmente diferente, onde é aplicado apenas uma rede neural em toda imagem. A rede divide a imagem em regiões e tenta prever as posições do objeto e a probabilidade para cada região. Esses objetos encontrados são ponderados de acordo

[1] Plataforma desenvolvida por Jeremy Howard e Rachel Thomasm disponível em: www.fast.ai
[2] Plataforma de machine learning desenvolvida pelo Google, disponível em: www.tensorflow.org

com as probabilidades previstas. Desta forma, os autores do *framework* buscam otimizar algumas características um tanto quanto lentas em outras formas de CNN.

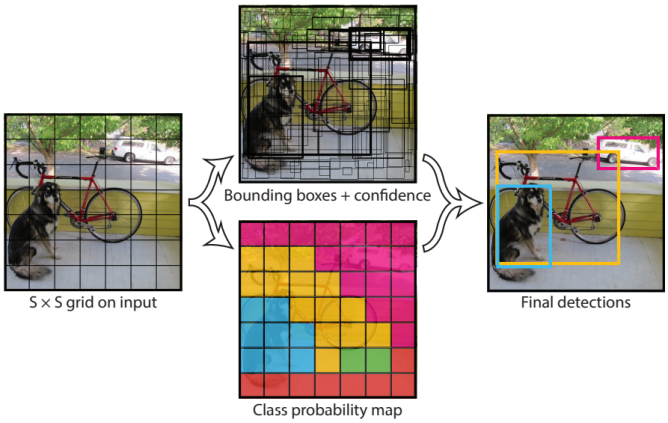

Figura 5 – Exemplo do processo adotado pelo *framework* YOLO. O diagrama de fluxos superior corresponde a representação de todas as regiões e valores de confiança encontrados para cada box e o caminho inferior é o mapa de probabilidade de classes. Ambos caminhos resultam na combinação e supressão para os maiores valores, devolvendo assim a saída da rede e a região e sua classe detectada.

Fonte: (REDMON; FARHADI, 2018)

O design do modelo acima permite a contextualização global dos objetos na imagem, e com regiões subdivididas cada célula se torna responsável por prever os objetos e probabilidades de ocorrências dentro do seu limite.

O *framework* vem sendo constantemente atualizado e modificado de acordo com sugestões da comunidade e novas ideias dos autores. Até a presente data, a versão utilizada é a YOLOV3 que teve uma importante mudança em sua arquitetura, como citado por (REDMON; FARHADI, 2018) correspondem a uma nova arquitetura híbrida entre a YOLOV2 e certos conceitos de redes neurais residuais (ResNet). Uma das principais alterações são as novas conexões entre as camadas, em inglês, *shortcut layers* e a adição de novas camadas de convolução, totalizando 53 nesta versão como visto na Figura 6.

4 Metodologia

4.1 Ambiente Experimental e Dataset

O sistema de processamento é organizado em cima da plataforma de nível gratuito *Google Colab*, na data utilizada a disponibilidade de recursos se traduzia em uma instância chamada **n1-highmen-2**, que possui um Intel Xeon 2vCPU @ 2.30GH 13 GB RAM e uma placa de processamento gráfico dedicada Tesla T4. Ressalta-se que o sistema possui um tempo de execução máximo de 12 (doze) horas nesta categoria, e após isso é reiniciado a instância. O código do *framework* utilizado está armazenado dentro do *github* no repositório de seu respectivo idealizador (ALEXEY, 2016).

Os recursos de dados utilizados são todos no formato de alta resolução pois experimentos prévios realizados em qualidade HD de 1080p não atenderam as expectativas e tampouco produziram resultados relevantes na pesquisa. Sendo assim, presume-se que

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figura 6 – Descrição da arquitetura de rede da Darknet-53, definição das operações presentes em cada camada

Fonte: (REDMON; FARHADI, 2018)

para qualquer reprodução futura e/ou melhora de experimento seja respeitado esta condição de imagens/vídeos em alta resolução, de no mínimo 4096x2304 e para vídeos, 30FPS. A elaboração do dataset utilizado não é feito com captura própria e sim a partir de conteúdos já existentes que capturam ruas, desta maneira, uma autorização prévia das imagens necessárias foi pedida aos detentores dos direitos autorais.

Algumas premissas foram estabelecidas para limitar o escopo do projeto, a principal delas é a posição da câmera que impacta de maneira essencial. Portanto, as ruas, avenidas e/ou rodovias com visão frontal do motorista e do carona foram as escolhidas para coleta. Dentre as opções citadas anteriormente, foram escolhidas especialmente aquelas em que os veículos passam por debaixo de pontes/viadutos ou câmeras dispostas em pontos altos da via. Quaisquer outras opções como visão lateral ou angulada de maneira rente à pista foi descartada por produzir variáveis demais no sistema tornando a pesquisa fraca perante a quantidade de dados.

Dentro dos dados coletados existem vídeos para serem analisados com o intuito de obter a maior quantidade de exemplos possíveis. Para tal, um script em Python foi elaborado para dissecar os frames dos vídeos e produzir novas imagens. Por se tratar de diferentes velocidades de reprodução e inclusive da velocidade da via, não existe um padrão para este script e sim uma análise visual de cada via de cada vídeo, já que uma via de maior velocidade pode resultar em mais veículos passando em frente a câmera do que uma com um limite menor. Devido a este fato a janela de coleta foi feita entre a cada 10 e 30 frames sempre buscando ocorrências de veículos e descartando as vias vazias.

Durante a seleção dos dados, um pré processamento é realizado para eliminar possíveis candidatos ruins ou imagens erradas que passaram nesta primeira coleta geral. Do montante inicial de 1438 frames salvos é produzido o recorte dos veículos que possuam

ocupantes utilizando cinto de segurança (objeto de estudo) ou não estejam utilizando (utilizados como exemplos negativos no treino). A ferramenta utilizada para anotar as regiões de interesse é licenciada por software livre e pode ser encontrada no repositório de seu desenvolvedor (TZUTA, 2015), e denominada de LabelImg, uma interface gráfica para anotação de imagens.

Figura 7 – Representação dos exemplos coletados para elaboração do conjunto de treino, visualização frontal e superior do motorista e carona em dias claro e de sol.

4.2 Data Preprocessing

Um modelo relevante só pode ser considerado bem sucedido caso consiga reproduzir de maneira precisa um novo e qualquer objeto de estudo nunca antes visto pelo modelo. Espera-se que para todos os novos exemplos postos a prova a rede consiga, neste caso em específico, discriminar corretamente as imagens e identificar quando da ocorrência de um cinto de segurança na imagem.

Entretanto, alguns obstáculos são facilmente encontrados pelo simples fato da baixa quantidade de dados. Em outros problemas mais antigos e mais estudados, a comunidade tende a criar e estruturar de maneira mais sólida a coleta de dados e construção dos datasets, pois já é algo que vem sendo feito por anos e por mais colaboradores dispostos a anotar as imagens.

Exemplos como o conjunto de dados do MNIST, um conjunto de imagens para dígitos manuscritos contém 60 000 exemplos somente para treino e outros 10 000 para testes. Já o ImageNet possui mais de 1.5 milhão de imagens de múltiplos objetos e suas respectivas regiões. O Open Images Dataset é outro famoso conjunto com cerca de 9.1 milhões de imagens para treino e 125 000 para testes. Por consequência o uso de técnicas de pré processamento e manipulação das imagens coletadas é algo extremamente necessário pois caso contrário o conjunto de dados seria substancialmente pequeno. Comparação esta com os números que a comunidade julga aceitável na elaboração de modelos neurais, ainda mais em situações atípicas como detecção de cinto de segurança.

Portanto, objetivando refinar os dados coletados, o aumento de dados foi feito em cima das imagens pensando em diversificar e generalizar as características antes de realizar o treinamento da rede, apesar de que a própria configuração da YOLO já apresente algumas técnicas de alterar saturação, exposição e o hue.

De volta ao ponto da coleta dos dados para treino, após uma seleção visual e anotação manual dos objetos, conseguiu-se apenas 546 ocorrências de cinto de segurança em 526 arquivos de imagens diferentes. Resultando em aproximadamente 403 imagens anotadas como positivo (cinto de segurança) e 123 imagens negativa. Neste caso, negativa significa apenas selecionar imagens sem cinto de segurança visível, estes exemplos podem serem observados na Figura 7. Tais números são valores baixos se comparados com outras pesquisa, em virtude disto o aumento de dados é feito para tentar sobrepor esta possível barreira de poucos exemplos.

Após a realização de algumas transformações julgadas válidas para o experimento, como contraste, brilho, ruído branco, inversão da orientação do eixo X, rotação aleatória, cisalhamento e um pouco de desfoque gaussiano pela imagem, o conjunto de dados atingiu cerca de 3000 imagens anotadas. Vale ressaltar que todas as modificações citadas anteriormente não são feitas ao mesmo tempo mas sim uma mistura de algumas delas para criação de efeitos que dificultem um pouco a imagem original, mas ainda assim permita sua visualização correta.

Em suma, o último ponto que deve-se salientar é referente ao aumento de dados. Esta técnica não é uma transformação que irá tornar o modelo sempre viável, afinal os dados não são criados totalmente de maneira a serem únicos, visto que todos eles são provenientes de algum exemplo já coletado. Assim, apenas permite-se que o modelo melhore a sua performance e reduza, quando possível, o *overfitting* a partir do aprendizado em cima das novas distorções inseridas propositalmente.

4.3 Abordagem Proposta

O método proposto se baseia na utilização de redes neurais convolucionais, mais especificamente o *framework* chamado YOLO (REDMON et al., 2015)(ALEXEY, 2016) o qual foi inspirado pela arquitetura de outra rede bem conhecida denominada de GoogLeNet (SZEGEDY et al., 2015). Uma grande diferença presente neste *framework* selecionado é a capacidade de observar a imagem como um todo, diferente por exemplo da Fast R-CNN (GIRSHICK, 2015), dando para a YOLO uma pequena vantagem em relação aos erros relacionados ao *background* quando comparados.

O sistema da rede utiliza um método tabular de células em cima de cada imagem de entrada, para assim detectar o objeto de acordo com o nível de confiança e posição central do objeto. O processo é refeito sucessivamente através de camadas de convoluções e *shortcut connections* como é chamado algumas ligações especiais entre as camadas, que liga diretamente uma camada a uma posterior (não próxima) como demonstrado na sua arquitetura de nome Darknet-53.

4.3.1 Training

Todos os procedimentos de trabalho são descritos na Figura 8 com um simples fluxograma de orientação. O caminho referente ao treino é ilustrado com um fundo amarelado, e nesta etapa, considerada crucial para o modelo, é realizada as iterações de

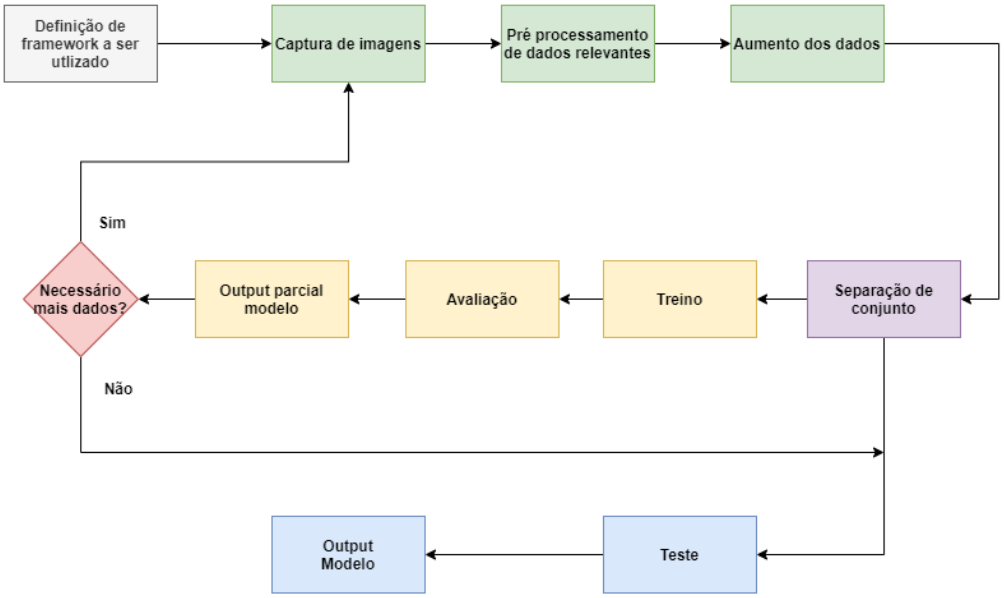

Figura 8 – Diagrama pensado para orientação do fluxo de trabalho da pesquisa, cada cor representa aproximadamente uma seção designada neste documento.

treino para minimização da função de erro, principal componente de análise para saber quão próximo as predições estão acontecendo.

Todas as imagens trabalhadas anteriormente e descritas na seção 4.1 são inseridas de maneira a preservar uma importante regra em aprendizado de máquina: dados de treinos não podem ser reutilizados em dados de testes. Desta maneira uma sub separação dos dados é feita e somente a parte referente ao treino é utilizado por agora.

Esta etapa do fluxograma não possui detalhes envolvidos já que a máquina precisa ficar executando e treinando até um certo limiar de erro seja atingido. Na literatura do *framework*, a parada da rede é recomendada no mínimo após a 2000 iteração ou quando observado que a média de erro seja menor ou igual que 0.05, ou quando também o mAP seja alto. A definição de mAP se refere aos valores médios de *average precisions (AP)* para cada classe, onde AP é o valor médio dos 11 pontos dentro da curva precision-recall (*PR-curve)* para cada *threshold possível*. A Figura 9 indica o comportamento do modelo durante a fase de treino até a sua parada: na **iteração 2300** com um **erro médio de 0.1736**, apesar de não ser exatamente o valor citado na literatura é possível que modelos nunca cheguem a este número, portanto com o número de iterações e mAP já satisfatório para o término do algoritmo.

A configuração original do arquivo não era feita com um tamanho de rede de 800px, porém buscando aumentar a precisão do modelo esta alteração foi feita para prevenir uma possível falha no treinamento. Desta forma, observou-se que a troca desses valores impactam diretamente no tempo e custo de processamento do treino, neste caso a duração.

4.3.2 Teste

Na bifurcação do caminho (durante a separação de conjunto) desenhado no fluxograma existe a condição de teste, onde os passos a serem seguidos são bastante similares aos procedimentos já realizados em cima do conjunto de treino. Os dados reservados para

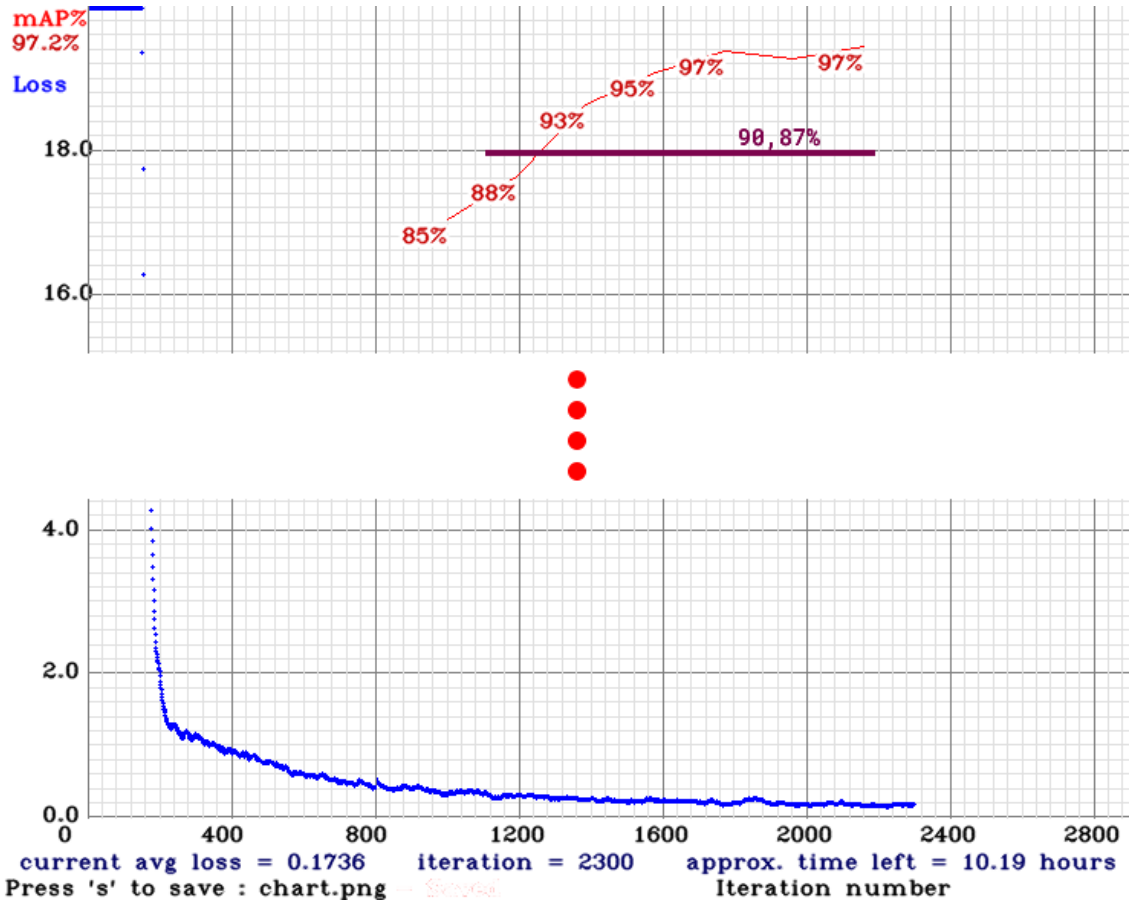

Figura 9 – Valores de precisão obtidos pela abordagem proposta para o conjunto de treinamento. Em azul está representado o valor do erro tendendo a zero e em vermelho o índice médio de precisão do conjunto de dados de treino. Inserido manualmente a linha em roxo para representar o valor de precisão apurado em teste.

teste já separados totalizam o valor de 436 arquivos nunca usados para treino, os quais são compostos por uma mixagem de exemplos de cinto de segurança, bem como a ausência do mesmo.

O balanceamento das classes são feitas de maneira a tentar construir uma equivalência entre os dois tipos. Para os exemplos positivos (cintos) há um total de 245 arquivos e para os negativos (sem cintos/ausência) há o complemento de 191 arquivos. Análises mais profundas são apresentadas na seção 5 onde valores novos são revelados bem como modificações no limiar selecionado para avaliação das detecções.

O resultado obtido para teste é adquirido pelo próprio *framework*. Como pode ser observado na Figura 10, após o comando para avaliar uma entrada de imagens, neste caso um conjunto denominado teste, que será processado pelo modelo e comparado com suas anotações. Assim como no treino, todos os exemplos coletados são compostos de uma imagem e suas respectivas anotações de região. Para preservar ao máximo a informação testada, o conjunto de teste só é utilizado nesta etapa. Os vídeos coletados são também de outras fontes, filmados em diferentes localidades que podem produzir condições visuais diferentes das demais encontradas em treino.

```
 calculation mAP (mean average precision)...
 Detection layer: 82 - type = 28
 Detection layer: 94 - type = 28
 Detection layer: 106 - type = 28
436
 detections_count = 474, unique_truth_count = 292
class_id = 0, name = seatbelt, ap = 90.87%       (TP = 247, FP = 16)

 for conf_thresh = 0.25, precision = 0.94, recall = 0.85, F1-score = 0.89
 for conf_thresh = 0.25, TP = 247, FP = 16, FN = 45, average IoU = 69.90 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.908662, or 90.87 %
Total Detection Time: 13 Seconds
```

Figura 10 – Resultado da avaliação do conjunto de testes diretamente no ambiente virtual na máquina do google colab, comando próprio do *framework* para cálculo de um *batch* de imagens.

4.4 Indicadores

Em outra passagem de seu livro, (GOODFELLOW; BENGIO; COURVILLE, 2016) cita que "determinar seus objetivos, em termos de qual métrica de erro usar, é o primeiro passo necessário porque a sua métrica de erro é o guia de todas suas decisões futuras". Sendo assim, para análise do modelo construído deve-se observar atentamente suas métricas reportadas, pois é a partir delas que descobrimos o quão eficiente foi o aprendizado do modelo computacional proposto. Existem diferentes maneiras de visualizar estes valores, desde uma simples matriz de resultados que poderá produzir métricas como precisão, acurácia, *f1-score*, *recall*, média de precisão e intersecção sobre união.

-	**Verdadeiro Positivo**	**Verdadeiro Negativo**
Previsto Positivo	*TP*	*FP*
Previsto Negativo	*FN*	*TN*

Figura 11 – Matriz de confusão, as previsões podem serem sumarizadas em uma das quatro categorias desta matriz e posteriormente usadas para cálculo de indicadores, na ordem da esquerda para direita e superior para inferior: Verdadeiro positivo, falso positivo, falso negativo e verdadeiro negativo.

Fonte: Próprio autor

A matriz de confusão demonstrada na Figura 11 apresenta as métricas mais utilizadas para validação de pares tais como os utilizados para a abordagem computacional proposta. Dentre as mais comuns, destacam-se precisão e *recall* demonstradas nas equações 2 e 3, respectivamente medindo a porcentagem do objeto alvo para a classe em questão que foi marcado corretamente e a porcentagem do objeto alvo (e somente ele) marcado

corretamente.

$$precision = \frac{TP}{TP + FP} \tag{2}$$

$$recall = \frac{TP}{TP + FN} \tag{3}$$

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = 2 \times \frac{precision \times recall}{precision + recall} = \frac{TP}{TP + \frac{FN + FP}{2}} \tag{4}$$

Entretanto para detecção de objetos, uma métrica importante que não se retira diretamente da tabela é a intersecção sobre união. Esta métrica mede a relação da intersecção do objeto com sua área detectada sobre a união do objeto e sua área detectada. Inclusive a definição de precisão e *recall* possuem levemente um outro significado segundo (EVERINGHAM et al., 2010), sendo: "para uma determinada tarefa e classe, a curva de precisão/*recall* é computada a partir de um método baseado nos ranks de *output*. *recall* é definido como a proporção de todos exemplos positivos acima daquele rank que são das classes marcadas como positivas. A precisão média (AP) representa o formato da curva de precisão/*recall* e é definida como a precisão média de um conjunto de onze níveis de *recall* igualmente separados".

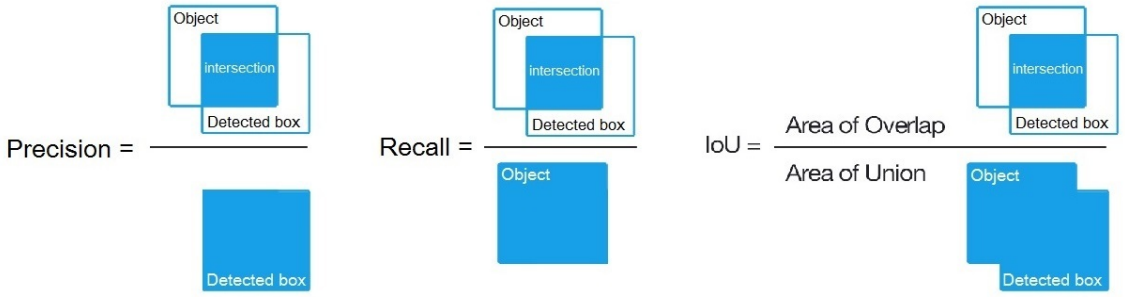

Figura 12 – Métricas de detecção de objetos em exemplos de imagens e como o conceito de região e objeto se unem para formar os indicadores.

Fonte: (ALEXEY, 2016)

Desta maneira, é de extrema importância a observação das métricas e não apenas elas separadamente, onde o próprio (GERONN, 2019) denota em seu texto sobre a conveniência de combinar duas métricas como precisão e *recall* em uma só chamada *f1-score*, sendo que este indicador é a média harmônica da precisão e do *recall* equacionado em 4, permitindo assim um tratamento mais regular dos dados e produzindo um valor de F1 alto apenas se ambos citados antes forem altos. Apesar de parecer conveniente sempre usar o *f1-score* como via de regra no modelo, é preciso observar qual o objetivo desejado, pois entra-se no dilema da compensação entre precisão/*recall*.

5 Resultados Obtidos e Validação

Com o objetivo de se combinar variantes do método anteriormente descrito e melhorar a taxa geral de performance do sistema, algumas análises foram feitas sobre especificações pontuais. Uma das primeiras alterações que resulta em significativas mudanças de precisão em função da alteração do tamanho da rede configurada no modelo, que impactam diretamente o conjunto de dados analisados. Primeiramente, os valores selecionados para treino da rede foram de: tamanho de janela da rede 800, exemplos com média de 875 pixels de largura e 766 pixels de altura. No entanto, os valores médios no conjunto de teste que foi utilizado na análise difere levemente para menor, onde a largura e altura média giram em torno de 581 e 586 pixels respectivamente, demonstrado na Figura 13.

Em virtude disto, algumas rodadas de alterações e experimentações no tamanho são produzidas sempre respeitando os valores múltiplos de 32, aqui a busca poderia ser realizada em maiores exemplos, porém apenas quatro foram selecionados pois assim obtêm-se uma ideia da tendência do mAP (se irá aumentar ou diminuir as métricas). Após esta coleta, os valores são comparados de acordo com o tamanho 800 original da rede e dispostos na Tabela 1 para visualização. O destaque em negrito sinaliza a escolha de valor para alteração do parâmetro, embora as diferenças obtidas sejam menores ou iguais a zero. O menor tamanho de rede se torna o candidato com preferência, já que as redes neurais são operações matemáticas, e reduzir processamento sempre é necessário para a boa performance em termos de execução do modelo.

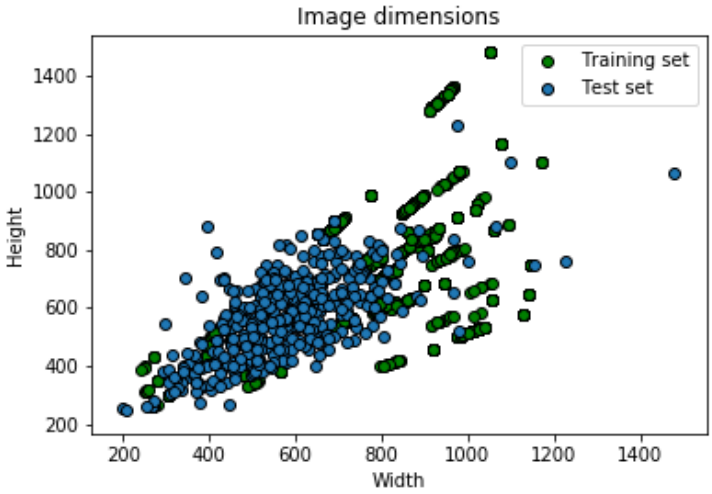

Figura 13 – Dimensões aproximadas dos arquivos de imagens usados para treino,representadas na cor verde, e para teste, demonstradas em azul

Contudo, quando ajustes são feitos no limiar de intersecção sobre a união das regiões, o IoU, a diferença produzida no indicador de mAP é quase que imperceptivelmente notada quando o maior valor ocorre em **mAP@0.3 = 0.9124**, e o valor que se obtêm no padrão da rede é **mAP@0.5 = 0.9086**, ou seja, uma diferença ínfima de **+0.0038** ou **+0.41%**, valor este que não justifica uma mudança em um parâmetro importante como a região de intersecção.

Em suma, seguindo os valores destacados em negrito da Tabela 2 são mantidos como guia para o projeto, justificando-se pelo indicador do *f1-score*, já apresentado em

Tabela 1 – Análise de alterações nos valores do tamanho de rede e consequências nas suas métricas de avaliação de modelo

Tamanho	Precisão	*recall*	F1	mAP
416x416	**0.94**	**0.85**	**0.89**	**0.9087**
640x640	0.88	0.89	0.89	0.9091
800x800	0.87	0.84	0.85	0.8646
960x960	0.86	0.74	0.79	0.8070

seções prévias como a 4.4, ser mais alto do que os outros e também pelas recomendações da literatura, mudanças no *conf_threshold* não afetam o valor final do mAP, apenas o valor do *f1-score*.

Tabela 2 – Análise de valores do limiar de intersecção para o tamanho de rede selecionado de 416 e a influência nas suas métricas de avaliação

Limiar	IoU Limiar	Precisão	*recall*	F1	mAP
0.25	0.25	0.94	0.85	0.89	0.9124
0.5	0.25	0.96	0.79	0.87	0.9124
0.25	**0.5**	**0.94**	**0.85**	**0.89**	**0.9086**
0.5	0.5	0.96	0.79	0.87	0.9086
0.25	0.75	0.47	0.42	0.44	0.2794
0.5	0.75	0.51	0.41	0.46	0.2794

Ademais, traduzindo as métricas em uma contextualização de fácil compreensão, o modelo treinado é capaz de realizar uma predição de **247** verdadeiros positivos, **16** falsos positivos e **45** falsos negativos baseados em um total de 436 exemplos separados para teste. Ressalta-se que no âmbito de detecção de objetos, o índice de verdadeiros negativos não são tão relevantes pois regiões da imagens fora da região proposta como alvo, isto é, o complemento de toda a região marcada como cinto de segurança é um verdadeiro negativo, o que eventualmente, se fosse considerado contabilizar este índice poderia ocorrer de saturar os outros três indicadores que compõe esta matriz de confusão.

Inegavelmente a Figura 14 demonstra a capacidade captar uma gama destes exemplos e de como eles são representados após sua introdução na rede, e uma rápida observação sobre eles apresentam algumas falhas de detecção para falsos positivos e falsos negativos que parecem ser provenientes de um padrão de distorção na imagem. No que tange aos falsos positivos, as imagens selecionadas aparentam ter um alto grau de complexidade e dificuldade para análise até mesmo para humanos. As cores de vestimentas são praticamente homogêneas e de tons escuros, o que resulta em um baixo contraste com o cinto de segurança e o interior do veículo como um todo. Outra observação importante é a obstrução parcial de objetos, algumas das vezes causadas pelo volante do motorista, partes do corpo ou até mesmo objetos soltos internamente e roupas mais destoantes do comum, o que deixa apenas uma pequena parte segmentada visível.

Em contrapartida ao mencionado, os falsos positivos parecem serem produzidos por pequenas variações da angulação da câmera e da luz na imagem. A luz pode ser considerada

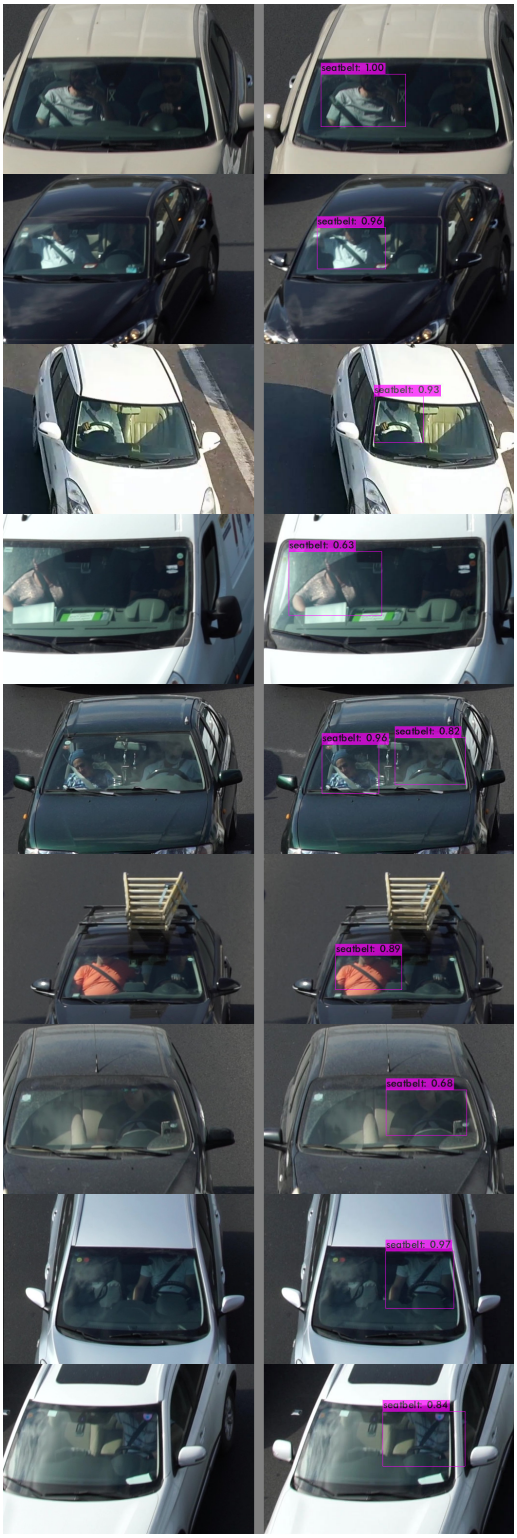

Figura 14 – Exemplos coletados na etapa de teste para verificar as marcações automáticas da rede, todas as imagens apresentadas não foram alteradas e nem avaliadas antes de passar pela rede.

a variável mais influenciadora na imagem pois altera perceptivelmente os tons de roupas, produzindo contraste em regiões devido a diferença de sombra e luz e criando a ilusão de um cinto de segurança sobre algum passageiro por exemplo. Surpreendentemente isto acontece inclusive em bancos vazios sombreados pela luz, em reflexos no para-brisa que produz uma forma geométrica similar ao cinto de segurança, uma das principais características do objeto de estudo. Estas ocorrências podem ser observadas na Figura 15, onde a imagem A é o resultado de um falso negativo e a imagem B é o resultado de um falso positivo.

Figura 15 – Exemplo de detecções com anomalias encontradas, em vermelho são marcações realizadas pela rede com seu grau de confiança e em verde marcações manuais previamente marcadas para efeito comparativo com a rede.

Uma última nota sobre detecção errôneas dos objetos é que exemplos claros para a percepção humanas não foram detectados, mas também não possuem as características citadas anteriormente na classificação de falso positivo ou falso negativo, levantando assim perguntas críticas do porque tais decisões foram tomadas e baseadas em que.

Por fim, apesar dos testes serem conduzidos em exemplos recortados de seu tamanho original de resolução 4k apenas por limitação de recursos, alguns exemplos originais são postos a prova do modelo, com suas devidas modificações do tamanho da rede citado em seções anteriores para um novo valor de altura e largura de 2560 pixels. O modelo consegue realizar as detecções de cintos de segurança nos motoristas tão bem quanto as detecções analisadas no conjunto de teste.

O modelo não limita-se somente à imagens, sendo que vídeos experimentais são obtidos com resultados igualmente satisfatórios. Porém uma afirmação completa e análise matemática dos resultados não pode ser elucidada pois a quantidade deste tipo de dados não é facilmente encontrada, e nem tão simples de ser produzidas com captura própria, levando assim a uma análise pouco fiel aos padrões de treino e teste comumente utilizados na comunidade e neste trabalho. De maneira geral, alguns testes em vídeos de definição

1080p apresentam indícios concretos da possibilidade da utilização de outras fontes de dados.

6 Conclusões e Trabalhos Futuros

Quando o assunto se resume a segurança e saúde, esforços devem serem sempre nivelados por cima e quando atrelados à tecnologia o desenvolvimento tem que ser ainda mais rápido e seguro. Apesar de ser uma tarefa simples, assegurar-se de usar o cinto de segurança nem sempre é prioridade dos motoristas e o retrato disto pode ser visto nos números coletados, longes do ideal. Desta maneira a automatização da tarefa de reforçar o uso deste dispositivo de segurança é extremamente relevante e claramente possível através de redes neurais convolucionais como um auxiliador durante a fiscalização local.

O modelo presente é capaz de distinguir diversos exemplos com um índice métrico importante, o mAP, na casa de **90.86%** com consideravelmente baixo número de exemplos em comparação com tópicos de pesquisa mais antigos e métodos tradicionais. Entretanto é óbvio que o ambiente de experimento montado é extremamente diferente do real, onde se tem menos controle das variáveis, principalmente no que tange condições do clima e da luz, mas isso não retira a oportunidade de novos caminhos para pesquisa.

Portanto, as possibilidades de trabalhos futuros podem serem extremamente proveitosas se condicionadas aos parceiros certos, por exemplo, governos e empresas relacionadas ao monitoramento e tráfico urbano, que são vitais para melhorar processos falhos na pesquisa, devido à suas limitações de recursos e falta de conjunto de dados mais estruturados. Implementações de um sistema composto com tecnologia infravermelho com as condições corretas de luz podem ser uma alternativa eficaz para pesquisas noturnas que não foram abordadas neste trabalho expandindo assim ainda mais o espectro de uso.

Por outro lado, outra abordagem interessante é o uso de hardware embarcado, área que vem constantemente crescendo e aumentando seu poder de processamento. Não é novidade que placas portáteis possuem agora tecnologia capaz de processar mais dados que estações antigas, vide a *Jetson Nano* da NVIDIA que é capaz de executar e treinar uma rede neural. Esta integração dos embarcados é algo promissor e certamente útil como trabalho futuro, com certeza será muito utilizada em aplicações voltadas para as novas cidades inteligentes, as quais demandam cada vez mais inovações para acomodar seus habitantes.

Em suma, os benefícios finais para sociedades são imensuravelmente grandes, e já podem melhorar desde a qualidade de trabalho, segurança e precisão dos agentes responsáveis pela fiscalização de trânsito como também a dos motoristas e consequentemente o reforço das leis locais. Como exposto, uma ação tão pequena e simples como afivelar o cinto de segurança se reverte em uma cadeia de redução de custos através de serviços e recursos usados para preservar o ativo da vida.

Referências

ALEXEY. *YOLOv4 - Neural Networks for Object Detection (Windows and Linux version of Darknet)*. 2016. Acessado em Outubro 10, 2020. Disponível em: <https://github.com/AlexeyAB/darknet>.

CHEN, Y. et al. Accurate seat belt detection in road surveillance images based on CNN and SVM. *Neurocomputing*, Elsevier B.V., v. 274, p. 80–87, 1 2017. ISSN 18728286.

DEY, S. *Hands-On Image Processing with ptyhon*. [S.l.]: Packt Publishing Ltd., 2018.

EVERINGHAM, M. et al. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, Springer, v. 88, p. 303–338, 6 2010. ISSN 09205691.

FERNáNDEZ-SANJURJO, M. et al. Real-time visual detection and tracking system for traffic monitoring. *Engineering Applications of Artificial Intelligence*, v. 85, p. 410 – 420, 2019. ISSN 0952-1976. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0952197619301691>.

GERONN, A. *Handson Machine Learning with Scikit-Learn, Keras, and Tensorflow*. United States: O'Reilly Media, 2019.

GIRSHICK, R. B. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. Disponível em: <http://arxiv.org/abs/1504.08083>.

GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN 9780131687288 013168728X 9780135052679 013505267X.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

GUO, H. et al. Image-based seat belt detection. In: *Proceedings of 2011 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2011*. [S.l.: s.n.], 2011. p. 161–164. ISBN 9781457705762.

HOWARD, J.; GUGGER, S. Fastai: A layered api for deep learning. *Information*, MDPI AG, v. 11, n. 2, p. 108, Feb 2020. ISSN 2078-2489. Disponível em: <http://dx.doi.org/10.3390/info11020108>.

Kim, H. et al. On-road object detection using deep neural network. In: *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. [S.l.: s.n.], 2016. p. 1–4.

KOTCHERGENKO, R. M.; LOPES, M. D.; COMUNELLO, E. *Pesquisa e desenvolvimento de tecnologias de visão computacional para o reconhecimento de passageiro frontal com cinto de segurança*. [S.l.], 2015. v. 0, n. 0, 358–367 p. Disponível em: <https://siaiap32.univali.br/seer/index.php/acotb/article/view/7051>.

LE, T. H. N. et al. Deepsafedrive: A grammar-aware driver parsing approach to driver behavioral situational awareness (db-saw). *Pattern Recognition*, v. 66, p. 229 – 238, 2017. ISSN 0031-3203. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320316303867>.

LECUN, Y.; BENGIO, Y.; HINTON, G. *Deep learning*. Nature Publishing Group, 2015. 436–444 p. Disponível em: <https://www.nature.com/articles/nature14539>.

LECUN, Y. et al. *Gradient-Based Learning Applied to Document Recognition*.

LI, W. et al. Seatbelt detection based on cascade Adaboost classifier. In: *Proceedings of the 2013 6th International Congress on Image and Signal Processing, CISP 2013*. [S.l.: s.n.], 2013. v. 2, p. 783–787. ISBN 9781479927647.

MANDACARU, P. M. P. et al. Qualifying information on deaths and serious injuries caused by road traffic in five brazilian capitals using record linkage. *Accident Analysis & Prevention*, v. 106, p. 392 – 398, 2017. ISSN 0001-4575. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0001457517302324>.

QIN, X. H. et al. Efficient seat belt detection in a vehicle surveillance application. In: *Proceedings of the 2014 9th IEEE Conference on Industrial Electronics and Applications, ICIEA 2014*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2014. p. 1247–1250. ISBN 9781479943166.

REDMON, J. et al. *You Only Look Once: Unified, Real-Time Object Detection*. [S.l.], 2015. Disponível em: <http://pjreddie.com/yolo/>.

REDMON, J. et al. *You Only Look Once: Unified, Real-Time Object Detection*. 2016.

REDMON, J.; FARHADI, A. *YOLOv3: An Incremental Improvement*. 2018.

Ren, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 39, n. 6, p. 1137–1149, 2017.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. *Understanding Machine Learning, From Theory to Algorithms*. [S.l.]: Cambridge University Press, 2014. <http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning>.

SZEGEDY, C. et al. Going deeper with convolutions. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2015. v. 07-12-June-2015, p. 1–9. ISBN 9781467369640. ISSN 10636919. Disponível em: <https://arxiv.org/abs/1409.4842v1>.

THEOFILATOS, A. Incorporating real-time traffic and weather data to explore road accident likelihood and severity in urban arterials. *Journal of Safety Research*, v. 61, p. 9 – 21, 2017. ISSN 0022-4375. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022437517301378>.

TIANSHU, W. et al. Detection and Implementation of Driver's Seatbelt Based on FPGA. *Journal of Physics: Conference Series*, IOP Publishing, v. 1229, p. 12075, 5 2019. Disponível em: <https://iopscience.iop.org/article/10.1088/1742-6596/1229/1/012075>.

TZUTA, L. *labelImg: LabelImg is a graphical image annotation tool and label object bouding boxes in images*. 2015. Acessado em Março 15, 2020. Disponível em: <https://github.com/tzutalin/labelImg>.

URI. *VISION-BASED SEAT BELT DETECTION SYSTEM*. 2007.

WANG, L. et al. Cluster-wise unsupervised hashing for cross-modal similarity search. *Pattern Recognition*, p. 107732, 2020. ISSN 0031-3203. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320320305355>.

WANG, Q.; WAN, J.; YUAN, Y. Locality constraint distance metric learning for traffic congestion detection. *Pattern Recognition*, v. 75, p. 272 – 281, 2018. ISSN 0031-3203. Distance Metric Learning for Pattern Recognition. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320317301401>.

WHO. Global status report on road safety 2018. *WHO*, World Health Organization, 2020. Acessado em Junho 23, 2020. Disponível em: <http://www.who.int/violence_injury_prevention/road_safety_status/2018/en/>.

XU, B. et al. A machine learning approach to vehicle occupancy detection. In: *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2014. p. 1232–1237. ISBN 9781479960781.

YONGQUAN, J. et al. GPU acceleration design method for driver's seatbelt detection. In: *2019 14th IEEE International Conference on Electronic Measurement and Instruments, ICEMI 2019*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2019. p. 949–953. ISBN 9781728105093.

YU, D.; ZHENG, H.; LIU, C. Driver's seat belt detection in crossroad based on gradient orientation. In: *Proceedings - 2013 International Conference on Information Science and Cloud Computing Companion, ISCC-C 2013*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2013. p. 618–622. ISBN 9781479952458.

ZHOU, B. et al. Learning-based seat belt detection in image using salient gradient. In: *Proceedings of the 2017 12th IEEE Conference on Industrial Electronics and Applications, ICIEA 2017*. [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2018. v. 2018-February, p. 547–550. ISBN 9781538621035.