

CENTRO UNIVERSITÁRIO FEI

JEAN LOURENÇO

LUCAS FONSECA

LUIS CARVALHO

**DATASET PARA RECONHECIMENTO DE COMPORTAMENTO DO MOTORISTA
BASEADO NO CONTEXTO DO AMBIENTE NO SETOR AUTOMOBILÍSTICO**

São Bernardo do Campo

2021

SUMÁRIO

1	INTRODUÇÃO	5
1.1	Objetivo	6
1.2	Estrutura do Trabalho	6
1.3	material complementar	6
2	Trabalhos Relacionados	7
2.1	detectores de objetos	7
2.2	estimador de posição da cabeça	9
2.3	rastreador ocular	10
2.4	reconhecedor de emoções faciais	11
2.5	Dataset	11
3	Conceitos Fundamentais	15
3.1	Sistema de posicionamento global	15
3.2	acelerômetro	15
3.3	microfone	16
3.4	You Only Look Once (YOLO)	16
3.5	Estimador de posição da cabeça	18
3.6	Rastreador ocular	18
3.7	Detector de emoções faciais	19
3.8	Database	20
3.8.1	ibug 300-W	20
3.8.2	COCO	20
3.8.3	FER-2013	21
3.9	Hardware utilizado	22
3.9.1	Aparelho celular	22
3.9.2	Servidor	23
4	Metodologia Proposta	24
4.1	Sensores de captura de informações	25
4.1.1	Sensores de captura da aceleração	25
4.1.2	Sensores de captura de imagem interna	25
4.1.3	Sensores de captura de som	25
4.1.4	Sensores de captura de imagem externa	25

4.1.5	Sensores de captura de localização	26
4.2	Servidor de Dados	26
4.2.1	Separação do áudio e vídeo	26
4.2.2	Processamento da Imagem externa	27
4.2.3	Processamento da Imagem interna	28
4.3	Criação do dataset	30
5	ESTRATÉGIA EXPERIMENTAL	32
5.1	Ambiente	32
5.2	Teste da identificação de objetos externos	32
5.3	Teste da identificação de objetos internos	33
5.4	Teste de rastreamento ocular parado	33
5.5	Teste de rastreamento ocular em movimento	34
5.6	Teste de estimativa de posição da cabeça parado	34
5.7	Teste de estimativa de posição da cabeça em movimento	34
5.8	Teste de detecção de emoções faciais	35
6	RESULTADOS e DISCUSSÃO	36
6.1	Resultados do experimento 1	36
6.2	Resultados do experimento 2	36
6.3	Resultados do experimento 3	37
6.4	Resultados do experimento 4	38
6.5	Resultados do experimento 5	38
6.6	Resultados do experimento 6	39
6.7	Resultados do experimento 7	40
6.8	Dataset	40
7	Conclusão	41
	REFERÊNCIAS	42

RESUMO

Com o trânsito crescente nas grandes cidades, compreender o comportamento dos motoristas se torna cada vez mais essencial por diversos motivos, como para permitir que os responsáveis possam tomar ações que visam proteger a vida da população. Tendo isso em vista, este trabalho tem como objetivo documentar e executar o ciclo completo de produção de um *dataset* que armazenará o comportamento dos motoristas de acordo com o contexto do ambiente, para isso usaremos um aparelho celular para capturar informações do ambiente, através dos diversos sensores embarcados nele. Além disso serão usados detectores de objetos, tanto para identificação de objetos internos quanto externos, um rastreador ocular, para determinar a direção em que o motorista está olhando, um estimador de posição da cabeça, para determinar a posição da cabeça do motorista, e um reconhecedor de emoções faciais, para determinar por qual emoção o motorista está passando.

O principal resultado é uma proposta de baixo custo que reúne um estimador de posição da cabeça, com precisão de 82,50% com o veículo em movimento, um rastreador ocular, que obteve uma precisão média de 71,25% com o veículo em movimento, dois detectores de objetos, um para imagens externas, que atingiu precisão 85,00%, e um para imagens internas, que atingiu 83,33%, e um reconhecedor de emoções faciais, cuja precisão obtida foi de 82,00%. O trabalho também contribui com um novo *dataset* com vídeos, áudios e dados provenientes de diversos sensores dos celulares para reconhecimento do comportamento do motorista.

Palavras-chave: Classificação de objetos, *Dataset*, Trânsito, YOLO, Rastreador ocular, Estimação de posição da cabeça, Reconhecedor de emoções faciais.

1 INTRODUÇÃO

Os *datasets* vêm se tornando cada vez mais importantes devido aos avanços tecnológicos em diversas áreas, fazendo com que seja necessário compreender o que são os *datasets* e como é realizado o seu processo de produção. Existem vários formatos diferentes que podem se qualificar como um *dataset*, desde um conjunto de imagens que capturam dados, até um conjunto organizado de tabelas. Sendo assim, pode-se definir que um *dataset* é qualquer conjunto de dados que apresente um formato definido e um propósito específico. O processo de produção de um *dataset* pode ser resumido a quatro etapas principais: coleta dos dados a serem armazenados, processamento desses dados para transformá-los no formato desejado, criação do *dataset* e validação do mesmo. Essas etapas servem para melhorar a qualidade dos dados e consequentemente aumentar a eficiência do *dataset*.

A demanda por *datasets* de qualidade aumenta a cada dia. Como demandantes podemos citar alguns exemplos como os setores automobilísticos, de marketing, de comércio, meteorológico e a área da saúde. Neste projeto, focar-se-á no setor automobilístico, onde esses *datasets* vêm se tornando cada vez mais importantes devido ao aumento do trânsito nas grandes cidades nas últimas décadas. Devido a esse aumento, tem se tornado cada vez mais importante compreender como se comportam os motoristas de acordo com o contexto do ambiente. Desde então, diversos trabalhos vêm surgindo propondo a criação de *datasets* que auxiliam nesta compreensão. Esses trabalhos, por mais diferentes que sejam, sempre têm uma ideia em comum: utilizam uma série de sensores dispostos ao redor do veículo. Porém, com a tecnologia atual, seria possível juntar todos esses sensores em um lugar só.

Pensando nisto, este trabalho propõe substituir todos esses sensores por aparelhos celulares. A metodologia deste trabalho utiliza detectores de objetos, tanto para identificação de objetos internos quanto externos, um rastreador ocular, para determinar a direção em que o motorista está olhando, um estimador de posição da cabeça, para determinar a posição da cabeça do motorista, e um reconhecedor de emoções faciais, para determinar por qual emoção o motorista está passando.

Dessa forma, este trabalho procura explorar a área automobilística visando a criação de um *dataset* que armazene o comportamento do motorista de acordo com o contexto do ambiente, utilizando os dados capturados por aparelhos celulares.

1.1 OBJETIVO

O objetivo deste projeto é executar o ciclo completo de produção de um *dataset* que armazena o comportamento de motoristas de acordo com o contexto do ambiente, capturando os dados do exterior e interior do veículo utilizando aparelhos celulares.

1.2 ESTRUTURA DO TRABALHO

A divisão do projeto foi realizada em seis capítulos, sendo o capítulo dois referente aos trabalhos relacionados, onde dissertaremos sobre os artigos que foram usados como base de estudos e sobre o *dataset*.

No capítulo 3, explicaremos sobre os conceitos fundamentais que serão utilizados neste projeto.

No capítulo 4, mostraremos a metodologia que será utilizada e suas etapas devidamente explicadas.

No capítulo 5, apresentaremos a estratégia experimental desenvolvida para testar a metodologia descrita no capítulo 4.

Por fim, no capítulo 6, mostraremos os resultados decorrentes dos experimentos descritos no capítulo 5.

1.3 MATERIAL COMPLEMENTAR

Os algoritmos implementados neste trabalho estão disponíveis em <<https://drive.google.com/drive/folders/1QFnsJGvZSvXvGIItqYNNOD4GB83qnFYxb?usp=sharing>>.

O dataset com os dados coletados nos experimentos pode ser acessado pelo mesmo endereço para futuros trabalhos que queiram avançar na área proposta por este trabalho.

2 TRABALHOS RELACIONADOS

Tendo como objetivo explorar o ciclo de produção de um *dataset*, foi realizado um levantamento de trabalhos relevantes a esta área. Estes foram divididos em cinco Seções: detectores de objetos, estimador de posição da cabeça, rastreador ocular, reconhecedor de emoções faciais e *dataset*. As três primeiras Seções correspondem a processos internos feitos para preparar os dados para a criação do *dataset*, que é abordada na última seção citada.

2.1 DETECTORES DE OBJETOS

Em Liu et al. (2016), os autores apresentam um método de detecção de objetos com o nome de SSD (*Single Shot Detection*) que se utiliza de uma única rede neural profunda. Este método delimita o espaço das *Bounding Boxes* em um conjunto de caixas de diferentes tamanhos e proporções por localização do mapa. No momento da previsão, a rede gera pontuações para a presença de cada categoria de objetos em cada caixa e então faz ajustes para melhor se encaixar ao formato do objeto. Resultados obtidos de experimentos nos seguintes datasets: COCO, Pascal VOC, ILSVRC (Russakovsky et al. (2015)), mostram que o SSD possui uma precisão competitiva a métodos que possuem uma camada adicional de proposta de objetos e é muito mais rápido. Para entradas de 300 x 300 o SSD obteve 74,3% mAP(mean Average Precision) no teste VOC2007 em 59 FPS, e para 512 x 512 obteve um mAP de 76,9% superando em desempenho modelos do estado da arte.

No trabalho de Redmon et al. (2016), os autores propuseram um método de detecção de objetos denominado de YOLO (*You Only Look Once*) que se utiliza de apenas uma única rede neural. Esta rede neural divide a imagem em regiões e prevê as *bounding boxes* e probabilidades para cada região. Essas *bounding boxes* são ponderadas pelas probabilidades previstas. O método proposto analisa a imagem inteira no momento do teste, para que suas previsões sejam informadas pelo contexto global da imagem. De acordo com o último relatório técnico Redmon e Farhadi (2018) os autores obtiveram resultados de mAP-50 em média a 55,3 ficando abaixo do RetinaNet(Lin et al. (2017)), entretanto obteve um tempo de execução 3,8x menor.

Em He et al. (2016), os autores apresentam um *framework* de aprendizado residual que facilita o treinamento de redes substancialmente mais profundas que as usadas anteriormente, de nome ResNet(*Residual Network*). Esse *framework* é usado para abordar o problema de degradação das redes neurais. Para atingir seu objetivo, os autores reformulam as camadas como

funções de aprendizado residual com referência às entradas da camada, ao invés de funções de aprendizado sem referência. No ImageNet *dataset* foram avaliadas redes com profundidades de até 152 camadas, um conjunto dessas redes treinadas usando o método sugerido obtiveram um resultado de 3,57% de erro neste *dataset*. Esse resultado obteve primeiro lugar no ILSVRC 2015 classification task. O método de treinamento também foi testado no COCO *dataset*, onde obteve uma melhora de 6% em relação a métrica padrão. Os autores concluem afirmando que redes residuais são mais fáceis de serem otimizadas, e que quanto maior a profundidade dessas redes mais precisas elas se tornam.

No trabalho de Ren et al. (2015) é apresentado uma RPN (*Region Proposal Network*) que compartilha de características convolucionais de imagem completa com redes de detecção, permitindo assim que a proposta de regiões seja feita quase sem custo. A RPN é treinada de forma ponta a ponta para gerar regiões propostas de alta qualidade, as quais serão usadas por uma R-CNN para detecção de objetos, esse conjunto leva o nome *faster R-CNN*. O método dos autores permite que um sistema de detecção de objetos que se utilize das regiões propostas tenha um tempo de execução de 5-17 fps. A RPN proposta ainda aumenta a qualidade das regiões propostas e consequentemente aumenta a precisão dos detectores de objeto, podemos citar que a *faster R-CNN*, obteve primeiro lugar na ILSVRC 2015 *object detection competition* por uma margem de 8,5%.

Em Dai et al. (2016), os autores apresentam o R-FCN, uma rede neural convolucional para a detecção eficiente e precisa de objetos, baseada em regiões. A estratégia de detecção proposta é dividida em 2 partes: proposta de região e classificação de região. As regiões candidatas são extraídas usando o *region proposal network(RCN)*. Dadas as regiões propostas a arquitetura do R-FCN é designada a classificá-las em categorias de objetos e *background*. Também se utiliza do estado da arte em classificação de imagem, como o ResNet, como *backbone*. O método sugerido obtém 83,6% mAP no *dataset* PASCAL VOC 2007 e 82,0% na versão de 2012, esses resultados foram obtidos com uma velocidade de teste de 170 ms por imagem, que é 2,5 a 20 vezes mais rápido que a *faster R-CNN* mais rápida e que o ResNet. O método atinge precisão competitiva com a *faster R-CNN*, mas é muito mais rápido durante o treinamento.

No trabalho de Wong et al. (2018) é introduzido o Tiny-SSD, uma rede neural profunda de *single-shot detection* para detecção de objetos em tempo real embarcado em aparelhos móveis, que é composta de uma *non-uniform Fire sub-network stack* altamente otimizada e uma pilha auxiliar convolucional de sub-redes não uniformes baseadas em Liu et al. (2016) com o objetivo de minimizar o tamanho do modelo, enquanto mantém a performance do detector de objetos. O Tiny-SSD possui tamanho de apenas 2,3 megabytes, o que é 26 vezes menor que

o Tiny YOLO (<<https://pjreddie.com/darknet/yolo/>>), ainda atingindo um resultado de mAP de 61,3% no VOC 2007, resultado esse que supera o Tiny YOLO em 4,2%. Esses resultados mostram que arquiteturas de redes neurais profundas de pequeno porte podem ser usadas para detecção de objetos em tempo real de forma embarcada.

Em Tan et al. (2019), é proposta uma arquitetura de busca neural automatizada para dispositivos móveis, o MnasNet, que incorpora explicitamente a latência do modelo como um dos objetivos principais, de modo que a busca possa identificar um modelo que alcance um bom equilíbrio entre precisão e latência. A abordagem sugerida pelos autores mede a latência diretamente do mundo real ao executar o modelo em aparelhos móveis. De modo a buscar um equilíbrio ainda maior entre flexibilidade e tamanho do espaço de busca, os autores propõem ainda um novo espaço de pesquisa hierárquico fatorado que estimula a diversidade das camadas em toda rede. Resultados obtidos no COCO *object detection dataset* demonstram que o MnasNet obtém mAP equivalente a métodos como YOLOv2 (Redmon e Farhadi (2017)) e SSD, enquanto tem 10 vezes menos parâmetros de treinamento que o YOLOv2 e 7 vezes menos que o SSD, obtendo também latência menor que o SSDLite.

2.2 ESTIMADOR DE POSIÇÃO DA CABEÇA

Em Ruiz, Chong e Rehg (2018), os autores argumentam que os métodos tradicionais de estimadores de posição da cabeça são frágeis porque dependem inteiramente do desempenho de detecção por pontos de referência. Para solucionar essa fragilidade os autores apresentam o Hopenet, um estimador de posição da cabeça robusto que utiliza uma *multi-loss CNN*, para prever os ângulos de Euler diretamente da matriz de intensidade de uma imagem através de classificação e regressão de *joint binned pose*. Resultados obtidos no BIWI Dataset (Fanelli et al. (2013)) demonstram que a metodologia proposta diminui o erro médio do ângulo de Euler em 1,29 graus quando comparado com Gu et al. (2017), e a metodologia proposta obtém também uma performance melhor que métodos como Dlib(Kazemi e Sullivan (2014)) e FAN(Bulat e Tzimiropoulos (2017)) quando testado no AFLW2000 dataset(Zhu et al. (2016)).

Em Yang et al. (2019), é apresentado o FSA-Net, um estimador de posição da cabeça a partir de uma única imagem, baseada em regressão e agregação de características. A fim de tornar o modelo compacto, os autores utilizam um esquema de regressão *soft stagewise*. É proposto também uma estrutura de mapeamento refinada para agrupar características antes da agregação, essa estrutura fornece informações baseadas em partes e valores agrupados. Os resultados mostram que o FSA-Net supera outros métodos como Dlib, FAN e Hopenet em erro

absoluto, obtendo 5,07, enquanto o Dlib obteve 15.8, o FAN obteve 9.12 e o Hopenet obteve 6.16. O FSA-Net apresenta ainda o modelo mais leve, tendo seu tamanho 100 vezes menor que os métodos citados.

2.3 RASTREADOR OCULAR

Em Kafka et al. (2016), os autores apresentam o GazeCapture, que é o primeiro *dataset* em larga escala para rastreamento ocular contendo data de mais de 1400 pessoas consistindo em aproximadamente 2.5 milhões de dados. Esse *dataset* é usado para treinar o iTracker, que é uma CNN que tem como objetivo o rastreamento ocular. Para isso é utilizado uma imagem da face juntamente com as coordenadas dela em relação a imagem, para conseguir inferir a posição da cabeça em relação a câmera; e a posição dos olhos na imagem, para que o modelo consiga determinar a posição dos olhos em relação a cabeça. Ao combinar essas informações o iTracker consegue inferir a direção que a pessoa está olhando. Os resultados mostram que o iTracker obtém um erro de previsão de 1.71cm e 2.53cm sem calibração, em telefones celulares e tablets, respectivamente. Com calibração esses valores caem para 1.34cm e 2.12cm. Além disso o modelo roda em tempo real, entre 10 e 15 fps.

Em Alemdag e Cagiltay (2018) é apresentada uma revisão sistemática da pesquisa de rastreamento ocular no domínio da aprendizagem multimídia. O objetivo principal da revisão é explorar como os processos cognitivos em aprendizagem multimídia são estudados com variáveis relevantes através da tecnologia de rastreamento ocular. Para atingir esse objetivo 52 artigos foram analisados, esses artigos foram selecionados através das seguintes bases de dados: *The Web of Science, Education Resources Information Center, Education Source, and PsycINFO*. As palavras-chaves usadas para realizar a busca foram multimídia e aprendizado, juntamente das seguintes palavras-chaves acompanhadas do operador "OU": movimento do olho, rastro do olho, movimento do olhar e rastro do olhar. Por fim, a fim de avaliar a qualidade e trabalhos mais recentes a busca foi limitada para artigos em inglês com texto completo e com data de publicação entre 2010 e 2016. Após a análise dos artigos os autores puderam constatar que existe um crescente interesse no uso da tecnologia de rastreamento ocular na pesquisa de aprendizagem multimídia e que os principais fatores que podem influenciar na detecção do movimento do olho são: os princípios do aprendizado multimídia, o conteúdo multimídia, diferenças individuais, metacognição e emoções.

No trabalho de Vicente et al. (2015), é proposto um sistema barato baseado em visão para detectar se o motorista de um veículo não está olhando para a via, este sistema tem o nome

de *Eyes off the Road* (EOR). O sistema possui três componentes principais: o rastreamento robusto das características faciais; postura da cabeça e a estimativa do olhar; e por fim o cálculo geométrico 3-D para detectar se o motorista não está com os olhos na via. O EOR em seus testes obteve uma precisão geral superior a 95% em vias urbanas, já em vias rurais obteve-se uma precisão superior a 90%.

2.4 RECONHECEDOR DE EMOÇÕES FACIAIS

Para o trabalho de Tarnowski et al. (2017), foi apresentado o resultado do reconhecimento de sete estados emocionais (neutro, alegria, tristeza, surpresa, raiva, medo e nojo) com base nas expressões faciais. Para isso foram usados coeficientes que descrevem elementos de expressões faciais como características. Estas características foram calculadas utilizando um modelo de face tridimensional, e então, são classificadas usando um classificador K-NN com $K = 3$, e uma rede neural MLP de duas camadas com 7 neurônios na camada oculta. Foi obtido uma precisão nas classificações das emoções de 96% para o 3-NN e 90% para o MLP.

No trabalho de Minaee e Abdolrashidi (2019), foi apresentado uma abordagem de aprendizagem profunda baseada em rede convolucional atencional, que é capaz de focar em partes importantes do rosto, e consegue uma melhoria significativa em relação aos modelos anteriores, também é utilizado uma técnica de visualização que é capaz de encontrar regiões importantes do rosto para detectar diferentes emoções, baseada na saída do classificador.. Esta abordagem obteve uma taxa de precisão de 70,02%, 98%, 99,3% e 92,8% nos respectivos datasets FER-2013, CK+ (Lucey et al. (2010)), FERG (Aneja et al. (2016)) e JAFFE (Lyons, Kamachi e Gyoba (1998)).

2.5 DATASET

Em Everingham et al. (2015), foram apresentados dois componentes: o primeiro, um *dataset* com imagens disponíveis ao público obtidos pelo site Flickr(<<https://www.flickr.com/>>), acompanhado de anotações de *ground truth* e um software de avaliação padronizado. O segundo, é uma competição anual e uma oficina que ocorrem desde o ano de 2006. Esta competição é dividida em: classificação, detecção, segmentação, classificação de ações e layout da pessoa e tem como público alvo, desenvolvedores e pesquisadores que querem ver qual é o estado da arte, medido pela performance no *dataset* VOC.

No trabalho de Lin et al. (2014), foi apresentado um novo *dataset* denominado de COCO que tem como objetivo avançar o estado da arte em reconhecimento de objetos. Este *dataset* reúne imagens de cenas complexas do cotidiano contendo objetos comuns em seu contexto natural. Os objetos são rotulados utilizando uma estratégia de segmentação por instância para auxiliar na localização precisa do objeto. Contendo fotos de 91 tipos de objetos, com um total de 2,5 milhões de instâncias rotuladas em 328 mil imagens. Comparando o desempenho médio do DPMv5 no pascal VOC e no COCO, mostra que o desempenho médio no MS COCO cai o fator por 2, sugerindo que o MS COCO possui imagens mais difíceis (não icônicas) de objetos parcialmente ocluídos, em meio a desordem e etc.

Em Scharstein et al. (2014), os autores apresentam um sistema de iluminação estruturado para criação de *dataset* estéreo de alta resolução de cenas internas estáticas com disparidade em *ground-truth* de alta precisão. O sistema inclui novas técnicas para busca eficiente de correspondência de subpixel em 2D e autocalibração de câmeras e projetores com modelagem da distorção da lente. Combinando estimativas de disparidade de várias posições do projetor, conseguimos obter uma precisão de disparidade de 0,2 pixels na maioria das superfícies observadas, inclusive em regiões parcialmente obstruídas. Sendo assim contribuindo com 33 novos *datasets* de 6 megapixels obtidos com o sistema e assim mostrando novos desafios para a próxima geração de algoritmos estéreos.

No trabalho de Geiger, Lenz e Urtasun (2012) é apresentado o KITTI, criado com o objetivo de ser um *benchmark* desafiador para ser usado para avaliar métodos de fluxo óptico, odometria visual e detecção de objetos 3D. As imagens presentes no *dataset* foram capturadas dirigindo por áreas rurais e em rodovias, usando duas câmeras estereoscópicas de alta resolução, uma em tons de cinza e outra colorida, um *Velodyne HDL-64E laser scanner* que produz mais de um milhão de pontos 3D por segundo, e um sistema de localização *OXTS RT 3003* que combina *GPS*, *GLONASS*, *an IMU and RTK correction signals*. Esses aparelhos são calibrados e sincronizados, fornecendo informações precisas sobre o mundo real. Resultados obtidos por estado da arte revelam que métodos com classificação alta em *datasets* como Middlebury, tem um desempenho abaixo da média quando levados para o mundo real.

Em Butler et al. (2012) é introduzido o MPI-Sintel, um novo *dataset* para avaliação de fluxo óptico baseado no curta de animação 3D Sintel. O *dataset* sugerido aborda características como sequências longas, grandes movimentos, reflexões especulares, desfoque de movimento e efeitos atmosféricos, características essas que não são abordadas por outros *datasets* como Middlebury. Uma das principais novidades introduzidas pelo MPI-Sintel é que uma mesma cena é renderizada diversas vezes com configurações diferentes, gradualmente aumentando a

complexidade. Os autores selecionam algoritmos de fluxo óptico com classificação alta no Middlebury *dataset*, e comprovam que os mesmos apresentam dificuldade quando testados em *datasets* mais complexos, sugerindo que é necessário mais pesquisas em estimativa de fluxo óptico.

No trabalho de Zhou et al. (2017), foi apresentado o Places Database um *dataset* que contém 10 milhões de fotografias de cenas, rotuladas com categorias semânticas de cenas, compondo assim uma lista grande e diversificada dos tipos de ambientes encontrados no mundo. A montagem do *dataset* é composta por quatro etapas: pesquisa e *download*, identificar as imagens com uma categoria de *ground truth*, expandir o *dataset* usando um classificador e melhorar ainda mais a separação de classes semelhantes. Comparando a diversidade entre o Places, ImageNet e SUN(Xiao et al. (2016)), mostra que o Places é o *dataset* que contém maior diversidade. Permutando o treinamento e teste entre os três *dataset* obtém se que o treinamento e o teste no mesmo *dataset* oferece o melhor desempenho para um número fixo de exemplos de treinamento. Como o Places é muito grande, nele se obtém o melhor desempenho em dois dos conjuntos de teste quando todos os dados de treinamento são usados.

Nome	Data de publicação	Tipo de dados	Etapas de produção	De onde foram obtidos os dados	Sensores/ ferramentas	Finalidade
KITTI	20/03/2012	Pares de imagens no formato png, arquivos de calibração, imagens distintas de cenas, categorias semanticas e de carros.	1°-Calibragão e sincronização dos sensores 2°-Coleta de dados 3°-Processamento dos dados 4°-Criação do dataset 5°-Validação do dataset	De áreas rurais e avenidas de uma cidade de tamanho médio	Gps, Laserscanner, 2 câmeras preto e branco (FL2-14S3M-C) e 2 coloridas (FL2-14S3C-C)	Optical flow, object tracking, object detection, visual odometry,
COCO	21/02/2015	Imagens, ground truth.	1°-Coleta de dados 2°-Classificar os dados em categorias 3°-Criação do dataset 4°-Validar o dataset	Através do uso do Amazon Mechanical Turk	Amazon Mechanical Turk	Object detection, segmentation, and captioning dataset
Places	04/07/2017	Imagens de alta resolução, imagem pequena.	1°-Coleta de dados 2°-Classificar os dados em categorias através do Amazon Mechanical Turk 3°-Classificar os dados restantes através de um classificador baseado em deep-learning AlexNet 4°-Otimiza a separação das categorias similares 5°-Criação do dataset 6°-Validação do dataset	Através de vários mecanismos de busca de imagens	Mecanismos de busca de imagens	Scene context, object recognition, action and event prediction, and theory-of-mind inference
MPI-Sintel	24/08/2012	Imagens, ground truth para optical flow.	1°-Coleta de dados 2°-Modificação na geometria da cena e no movimento da câmera de 2 vídeos 3°-Modificação na renderização através dos seguintes passos: albedo, limpeza, e final 4°-Estimar onde pode ocorrer obstrução nos limites dos movimentos	Do curta de animação Sintel	Short-film gerado por computador, Blender	Optical flow e Object tracking
Middlebury	15/10/2014	Imagens variadas cada uma fotografada com alguns dos três tipos de iluminações escolhida, as resoluções variando entre grande, média e pequena e cada imagem contendo seu respectivo disparity maps.	1°- Coleta dos dados sob diferentes iluminações e posições 2°- Decodificação e Interpolação das imagens obtidas 3°- Para cada pixel de uma imagem é procurado o pixel mais similar em outra imagem 4°- Dado 2 mapas de disparidade é realizado uma filtragem, seguida de um combinação dos mapas 5°- Refinamento da calibragão das camera 6°- Calibragão dos projetos de luz	Ambiente interno	2 cameras Canon DLSR e 2 cameras compactas fixadas em um suporte, projetores de luz	Optical flow, object detection
Pascal VOC	25/06/2014	Imagens, ROI (region of interest) annotated objects.	1°- Coleta de dados 2°-Classificar os dados em categorias 3°-Criação do dataset 4°-Validação do dataset	Do site de compartilhamento de fotos Flickr e de outros datasets	Mecanismos de busca de imagens	Object class recognition, object detection

Tabela 1 – Tabela comparativa dos trabalhos relacionados de *dataset*.

Na Tabela 1 é possível notar que existe uma ordem nas etapas de criação de um *dataset*, embora eles sejam diferentes para todos os trabalhos é possível reduzir a quatro etapas: coleta

de dados, processamento dos dados, criação do *dataset* e validação do *dataset*. Neste trabalho usaremos essas etapas para guiar a nossa metodologia.

3 CONCEITOS FUNDAMENTAIS

A seguir serão apresentados os conceitos fundamentais que serão abordados pela metodologia e pela estratégia experimental. Esses conceitos abrangem os sensores utilizados na metodologia (GPS, acelerômetro e microfone), o detector de objetos (YOLO), estimador de posição da cabeça, rastreador ocular e reconhecedor de emoções faciais utilizados para o processamento dos dados, os *datasets* usados para os treinamentos (COCO, ibug 300-W, fer2013), e os *hardwares* que serão utilizados neste projeto (aparelho celular e servidor).

Na Seção de cada conceito será descrito seu funcionamento, e caso necessário, descrito mais detalhadamente alguma funcionalidade relevante para a metodologia proposta no capítulo 4.

3.1 SISTEMA DE POSICIONAMENTO GLOBAL

O GPS (*Global Positioning System*) funciona tendo como base uma rede de satélites que orbitam o planeta. A sua função é localizar a posição do receptor do sinal, para isso é necessário que o receptor se encontre no campo de visão de pelo menos 3 satélites, quanto maior for os números de satélites que tiverem visão do receptor maior é a precisão. Os satélites e os receptores GPS, possuem um relógio interno, que marca o tempo com uma precisão de nanossegundos. Quando o sinal é emitido, é enviada a hora em que ele “saiu” do satélite. Esse sinal, que viaja à velocidade da luz, é recebido no receptor que calcula quanto tempo que ele demorou a chegar, a partir dessa diferença do tempo de emissão e recepção do sinal e da posição dos satélites, é calculada a posição do receptor.

3.2 ACELERÔMETRO

O acelerômetro é um dispositivo usado para medir a aceleração própria de um sistema. A aceleração própria de um sistema é medida em relação a outro sistema em queda livre, pois está baseado na teoria da relatividade geral desenvolvida pelo físico Albert Einstein em 1915. Desde então o acelerômetro ganhou uma série de aplicações importantes, como monitoramento sísmico, aplicações médicas e uso em aparelhos celulares, desde que o Iphone inovou trazendo esse recurso para os celulares. O acelerômetro no celular tem a função de medir a inclinação e o movimento do mesmo. A medição da inclinação serve para que os dispositivos sempre exibam

as imagens na vertical, já a medição do movimento serve para detectar quando o aparelho está caindo e fazer com que as cabeças de gravação do HD travem em posição, evitando que dados sejam perdidos em eventuais impactos.

3.3 MICROFONE

Um microfone é um dispositivo transdutor eletroacústico que converte ondas de pressão sonoras em um sinal elétrico alternado que pode ser amplificado, gravado ou transmitido por um circuito, e então reproduzido por um alto-falante ou fones de ouvido. Os celulares usam pequenos condensadores de eletreto ou microfones MEMS. Esses microfones requerem muito pouca energia para funcionar, o que é facilmente fornecido pela bateria do celular além de se encaixarem muito bem no circuito do celular.

O microfone de eletreto é um tipo de microfone que acaba com a necessidade de uma fonte de tensão para se tornar polarizado. Essa polarização é adquirida pelo uso de um material quase que permanentemente carregado no dipolo capacitor, o eletreto. Esses microfones em comparação com os condensadores, é que não precisam de uma voltagem de polarização, mas eles possuem um pré-amplificador integrado de baixa potência. Este pré-amplificador além de ser composto por apenas um transistor, é mantido externamente ou por uma pilha comum de 1,5 volts dentro do próprio microfone.

Um microfone MEMS (MicroElectrical-Mechanical System) é um microfone gravado em uma *wafer* de silicone ou "chip". Os microfones MEMS apresentam diafragmas sensíveis à pressão e geralmente são construídos com pré-amplificadores / conversores de impedância integrados. Além disso, eles também incluem conversores de analógico para digital no mesmo chip para uso com celulares digitais.

3.4 YOU ONLY LOOK ONCE (YOLO)

O YOLO é uma abordagem que interpreta a detecção de objetos como um único problema de regressão, desde a obtenção dos pixels da imagem, até as coordenadas geradas pelas *bounding boxes*, e subsequentemente, a classificação do objeto obtida pela probabilidade deste pertencer a determinada classe. (REDMON et al., 2016)

O YOLO possui várias versões atualizadas além de sua versão base, o mais atual é o YOLOv4 lançado em 2020 e quando comparado ao YOLOv3, obtém uma melhora na precisão e no FPS.

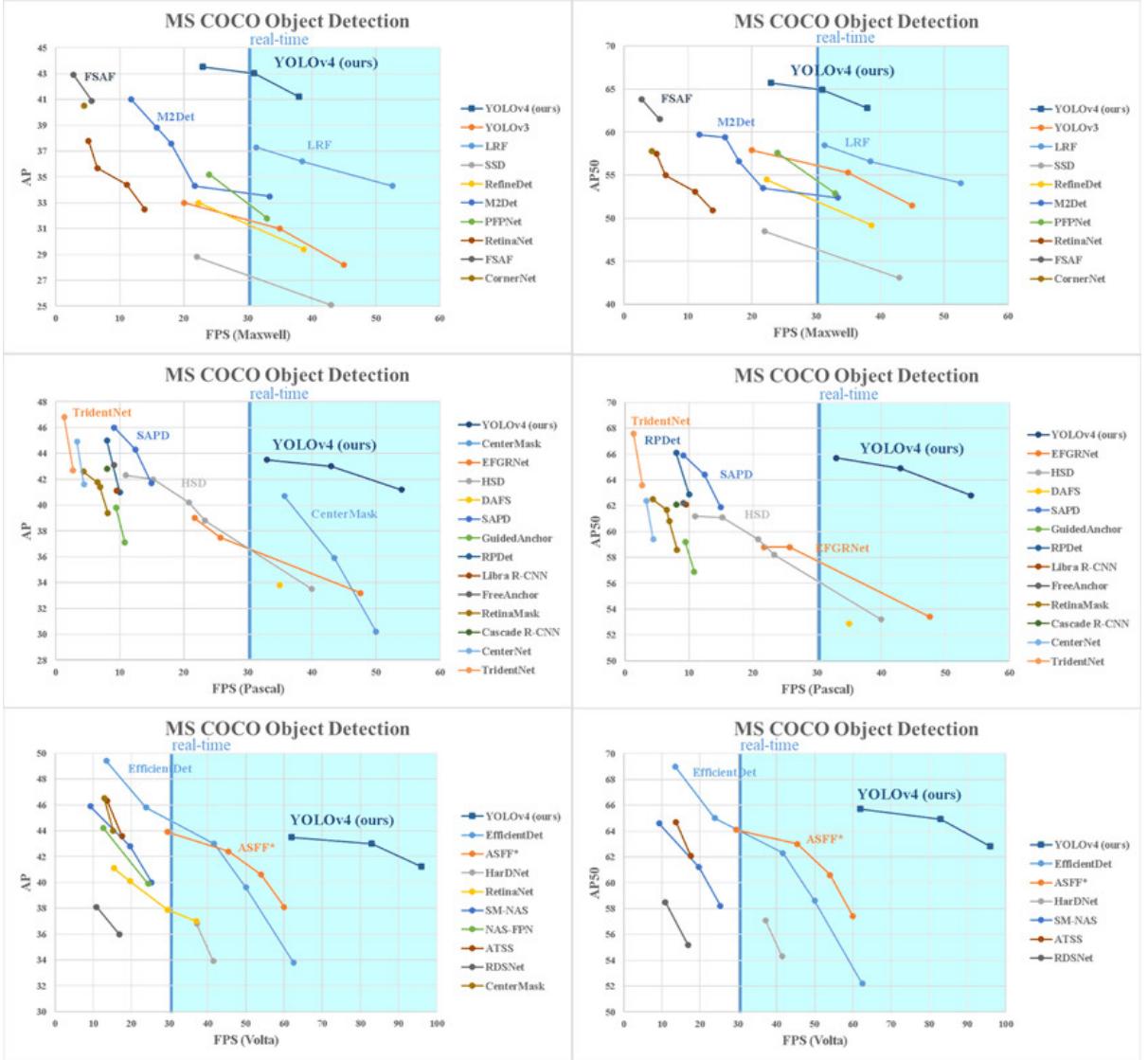


Figura 1 – Gráfico que mostra uma comparação do YOLOv4 com outros estados da arte em detecção de objetos, onde AP é a *average precision* e FPS significa frames por segundo e o Maxwell, Pascal e Volta são as GPUs usadas, imagem obtida em (BOCHKOVSKIY; WANG; LIAO, 2020)

Mesmo tendo diversas versões, a forma pela qual o objeto é detectado continua praticamente a mesma, para fazer as *bounding boxes* o YOLO faz uma divisão da imagem de entrada em diferentes regiões, onde, a célula em que o centro do objeto está é a responsável por detectá-lo. Para cada *bounding box*, uma pontuação é calculada baseando-se em quanto o sistema acredita que ali há algum objeto para o qual ele foi treinado. Além disso, cada célula da grade que possuir um objeto, recebe uma pontuação que corresponde a probabilidade daquele objeto pertencer a uma determinada classe para a qual foi treinada, o que resulta em um conjunto de probabilidades indicando o que possivelmente aquela imagem representa.

3.5 ESTIMADOR DE POSIÇÃO DA CABEÇA

A estimação da posição da cabeça possui várias aplicações na área de visão computacional, como em aplicações em realidade virtual, aplicações com sistemas de controles orientados a gestos, detecção da atenção de motoristas e estimar a direção que uma pessoa está olhando.

O estimador de posição da cabeça usado nesse projeto foca em resolver o *Perspective-n-Point problem (PnP problem)*, que é definido pela seguinte equação.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1)$$

O lado esquerdo da equação $s[u v 1]$ denota a imagem 2D obtida pela câmera. O lado direito da equação, a primeira porção que se parece com uma matriz triangular superior é a matriz da câmera onde $f(x, y)$ são as distâncias focais, γ é o parâmetro de inclinação que foi como 1 no estimador usado, (u, v) são o centro da imagem. A porção do meio, r e t representa a rotação e translação, a porção final denota o modelo 3D da face que for detectada na imagem.

Portanto para estimar a posição da cabeça é necessário detectar a face e calcular as matrizes da câmera e a matriz 2D de referência, para assim conseguir solucionar o *Pnp problem* e adquirir a rotação e translação da face. Para tal são usados seis pontos de referência do rosto, o queixo, a ponta do nariz, o canto esquerdo do olho esquerdo, o canto direito do olho direito, e os dois cantos da boca. Com esses dados, o estimador resolve o *Pnp problem* utilizando a API *solvePnP* do OpenCV e tem como resposta um vetor de rotação e um vetor de translação.

Por fim, os dados são usados para calcular os ângulos de Euler através da API *RQDecomp3x3* do OpenCV, para isso é necessário primeiro transformar o vetor de rotação em uma matriz de rotação, e para tal é usada a API Rodrigues, que faz essa conversão. E ao final esses ângulos calculados são usados para determinar a posição da cabeça da pessoa.

3.6 RASTREADOR OCULAR

O rastreamento ocular é um processo para mensurar a direção para qual uma pessoa está olhando ou o movimento do olho em relação a cabeça. Os principais usos de rastreadores oculares estão ligados a pesquisas na área do sistema visual, na psicologia, marketing e como dispositivo de entrada para interação humano-máquina. Os rastreadores também vem

sendo usados para reabilitação e aplicações de ajuda, como para controlar cadeiras de roda para pessoas com paralisia.

Assim como a estimativa da posição da cabeça, no rastreamento ocular, também é necessário detectar a face e os principais pontos do rosto, após essa detecção é completamente isolado um olho das outras das outras partes do rosto para obter um frame com apenas o olho, e então são realizados cálculos de calibração para cada frame a fim de obter o melhor valor dos limites, e com esses valores limites (*Threshold*) binarizar o frame contendo apenas o olho, para obter assim um frame contendo o elemento representado a íris. Na íris é calculado a porcentagem que ela ocupa dentro do olho e sua detecção e estimativa é realizada através do cálculo do centróide. Após essas etapas é gerado um valor de 0 a 1 onde ambos são extremos, e indicam a direção vertical e horizontal da pupila, e através desses valores o algoritmo consegue estimar se a pessoa está olhando para direita ou esquerda.

3.7 DETECTOR DE EMOÇÕES FACIAIS

A detecção de emoções faciais é um processo que identifica expressões de pessoas e as classifica entre as emoções humanas. Algumas das aplicações desses detectores são na área da saúde, permitindo que o médico consiga monitorar a saúde emocional de forma não invasiva, na área do marketing, permitindo medir a reação em tempo real dos espectadores ao assistir um anúncio, e na área de visão computacional, em aplicações de interações humano-máquina.

O detector usado recebe como entrada o frame atual da câmera, e executa um *Haar Cascades*, que é um classificador usado para detectar objetos em uma imagem. No caso deste detector é usado a API CascadeClassifier do OpenCV, que executa um *Haar Cascades*, e em conjunto da API cvtColor, que deixa a imagem da câmera em tons de cinza, e detectMultiScale, que detecta esses diferentes tons de cinza e é utilizado para detectar o rosto da pessoa na imagem. Depois de detectado, o detector recorta o rosto e redimensiona o mesmo para 48x48 pixels, e na sequência coloca os pixels da imagem em um array. Esse array é inserido em um modelo pré-treinado, onde ocorre a predição da emoção do rosto detectado baseado nas emoções pré-treinadas no modelo.

Por fim, o detector retorna uma porcentagem para cada emoção, sendo a que tiver maior porcentagem, considerada a emoção detectada no rosto.

3.8 DATABASE

As bases de dados têm uma grande importância na área da visão computacional por fornecer imagens e vídeos com o intuito de auxiliar no treinamento da metodologia envolvida. Esses dados são disponibilizados de forma rotulada e divididas em categorias distintas.

A seguir serão descritos brevemente as bases de dados que foram usadas no treinamento do reconhedor de objetos, rastreador ocular, estimador de posição de cabeça e do detector de emoções faciais.

3.8.1 ibug 300-W

O ibug 300-w foi uma competição realizada em 2013 no qual os participantes tinham seus algoritmos de detecção facial por pontos de referência testados utilizando um dataset que contém 600 imagens ao total divididos em duas categorias: indoor e outdoor. O dataset 300-W tem como objetivo testar a capacidade dos sistemas atuais de lidar com assuntos invisíveis, independentemente de variações na pose, expressão, iluminação, fundo, oclusão e qualidade da imagem.



Figura 2 – A imagem a esquerda está presente na categoria outdoor enquanto a da direita está na indoor, presentes no ibug 300-w

Para ter acesse ao dataset que foi utilizado na competição ou para saber mais sobre ela acesse o link a seguir <<https://ibug.doc.ic.ac.uk/resources/300-W/>>

3.8.2 COCO

O COCO Dataset(Lin et al. (2014)) é um banco de dados que possui imagens divididas em 91 categorias. O COCO conta também com um conjunto de imagens de ações humanas, por

isso ele pode ser usado para o treinamento de uma vasta gama de redes neurais. Esse *database* conta com um total de 2,5 milhões de instâncias rotuladas em 328 mil imagens segmentadas.



Figura 3 – Uma imagem que está segmentada entre pessoas, veículos e semáforos que está presente no COCO.

É possível saber mais sobre o COCO, desde como fazer o *download* gratuito da base de dados ou até participar de desafios fornecidos pelos criadores, acessando o link <<http://cocodataset.org/#home>>

3.8.3 FER-2013

O FER-2013 (*Facial Expression Recognition 2013*) foi uma competição realizada em 2013 com o intuito de avaliar a precisão de algoritmos que detectam a expressão facial de pessoas. Para isso foi utilizado um dataset que consistem em imagens de rostos em tons de cinza de 48x48 pixels. Os rostos foram registrados automaticamente para que fiquem mais centralizados possíveis e ocupem aproximadamente a mesma quantidade de espaço em cada imagem. O conjunto de treinamento consiste em 28.709 exemplos e o conjunto de teste público consiste em 3.589 exemplos. As imagens são classificadas em: *happy, neutral, fear, sad, disgust, angry* e *surprise*.

Para saber mais sobre a competição realizada acesse <<https://bit.ly/2IuUt4k>> e para ter acesso ao dataset utilizado na competição <<https://www.kaggle.com/msambare/fer2013>>

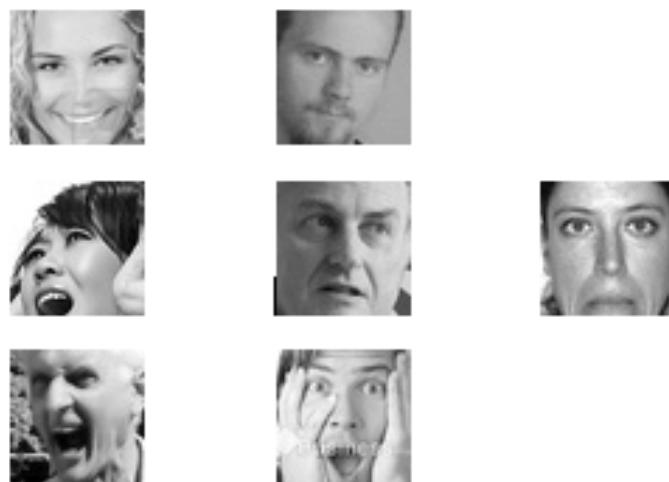


Figura 4 – Imagens retirada do dataset FER que foram utilizadas na competição

3.9 HARDWARE UTILIZADO

Nessa seção detalharemos quais aparelhos celulares foram utilizados e quais os requisitos mínimos que o celular deve ter, como também descreveremos as características do nosso servidor.

3.9.1 Aparelho celular

Para esse trabalho tivemos a nossa disposição o galaxy S8+ e o galaxy A7 da Samsung, porém para a reprodução desse projeto é possível usar qualquer aparelho que tenha o sistema operacional Android e que também possua os seguintes sensores:

- Acelerômetro;
- GPS;
- Microfone;
- Câmera traseira e frontal;

Sendo que a qualidade da câmera influencia diretamente a precisão do sistema.

3.9.2 Servidor

Criamos um webserver utilizando como base o Node JS e com o Nginx sendo utilizado para proxy re-verso. Com as seguintes configurações da máquina:

- Processador AMD Ryzen 5 1600 Sinx-Core 3.2GHZ;
- Memória ram: 16gb ddr4;
- Sistema operacional: Windows 10 pro;

4 METODOLOGIA PROPOSTA

A metodologia proposta deste projeto é composta de sete etapas, como pode ser visto na Figura 5. A partir dos sensores de dois celulares posicionados no para-brisa do veículo, vários tipos de informações serão capturadas, e enviadas ao nosso servidor para processamento e análise, onde posteriormente serão utilizadas para a criação de um *dataset* sobre o comportamento do motorista no trânsito. A seguir, cada uma dessas etapas mostradas na Figura 5 será descrita detalhadamente.

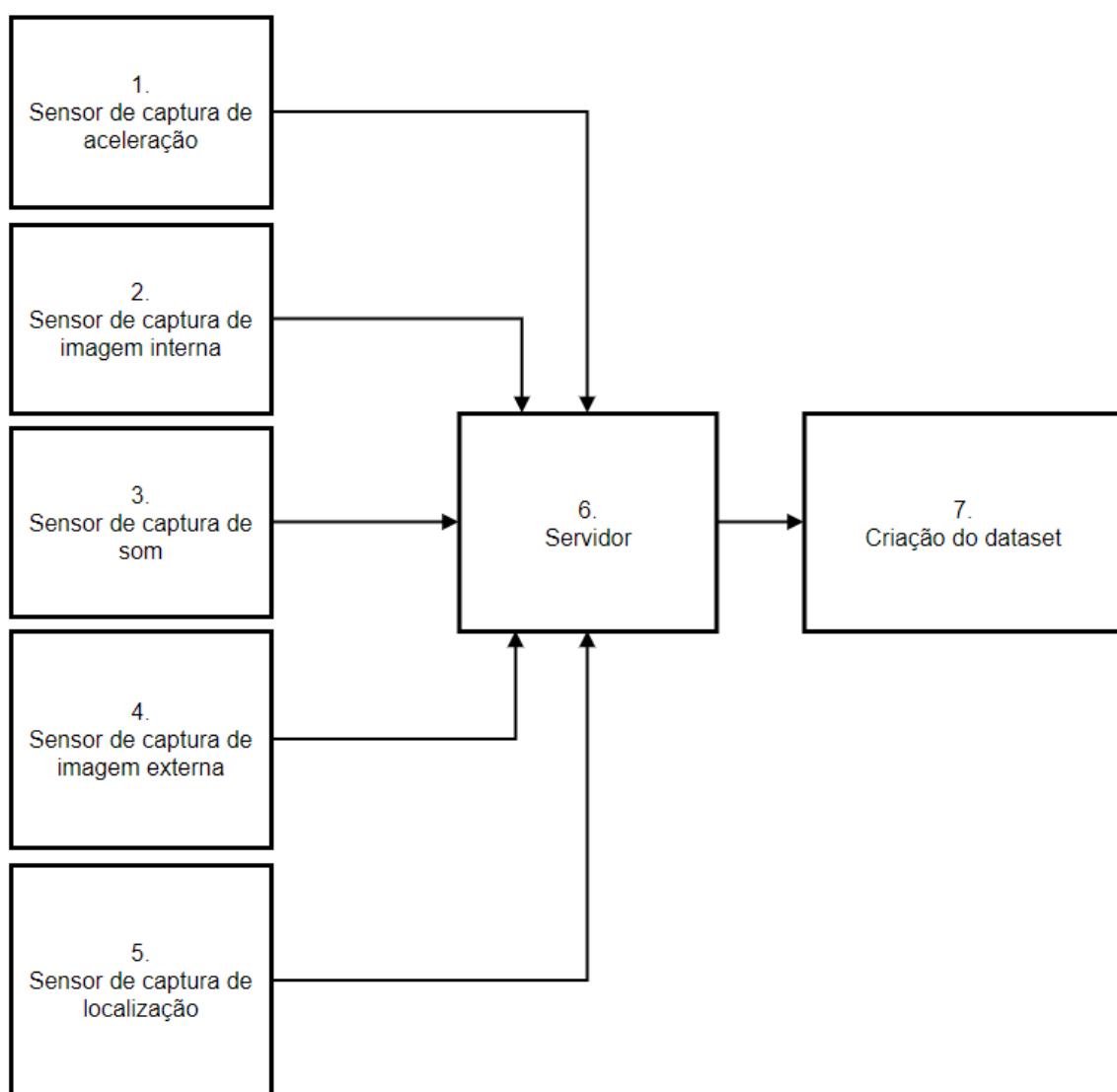


Figura 5 – Diagrama esquemático da metodologia proposta dividida em 7 etapas.

4.1 SENsores de CAPTURA DE INFORMAÇÕES

Nessa seção serão abordadas as etapas 1 a 5 da Figura 5, que descrevem como as informações são capturadas. Para capturar os dados 2 celulares serão usados, o primeiro com visão frontal do motorista, e o segundo com visão da via. A seguir será descrito quais sensores do aparelho celular serão usados para capturar cada tipo de informação.

4.1.1 Sensores de captura da aceleração

Na etapa 1 da Figura 5, a aceleração do veículo, com e sem gravidade, será capturada através do acelerômetro do primeiro celular e enviada para a etapa 6 da Figura 5. Para mais informações sobre o acelerômetro ver a Seção 3.2.

4.1.2 Sensores de captura de imagem interna

Na etapa 2 da Figura 5, as imagens internas serão capturadas utilizando a câmera do primeiro celular, sendo responsável por fornecer o vídeo para o processamento da imagem interna. Por fim, esse vídeo será enviado para a etapa 6 da Figura 5.

4.1.3 Sensores de captura de som

Na etapa 3 da Figura 5, o som ambiente será capturado através do microfone de ambos celulares juntamente dos vídeos, e enviado para a etapa 6 da Figura 5. A forma pela qual o microfone captura os sons do ambiente está descrito na Seção 3.3.

4.1.4 Sensores de captura de imagem externa

Na etapa 4 da Figura 5, as imagens externas serão capturadas através da câmera do segundo celular, sendo responsável por fornecer o vídeo para o processamento da imagem externa. Por fim, esse vídeo será enviado para a etapa 6 da Figura 5.

4.1.5 Sensores de captura de localização

Na etapa 5 da Figura 5, a localização do veículo será captada através do GPS presente no segundo celular e enviada para a etapa 6 da Figura 5. A forma pela qual o GPS funciona está descrita na Seção 3.1.

4.2 SERVIDOR DE DADOS

Nessa seção descrevemos a etapa 7 da Figura 5, onde serão realizados processamentos na imagem interna e externa, além da separação do áudio e vídeo. Os caminhos para os dados serão armazenados no MongoDB. A Figura 6 mostra como serão realizados esses processamentos. A seguir será descrito como serão realizados a separação e os processamentos internos e externos, detalhando como eles ocorrerão.

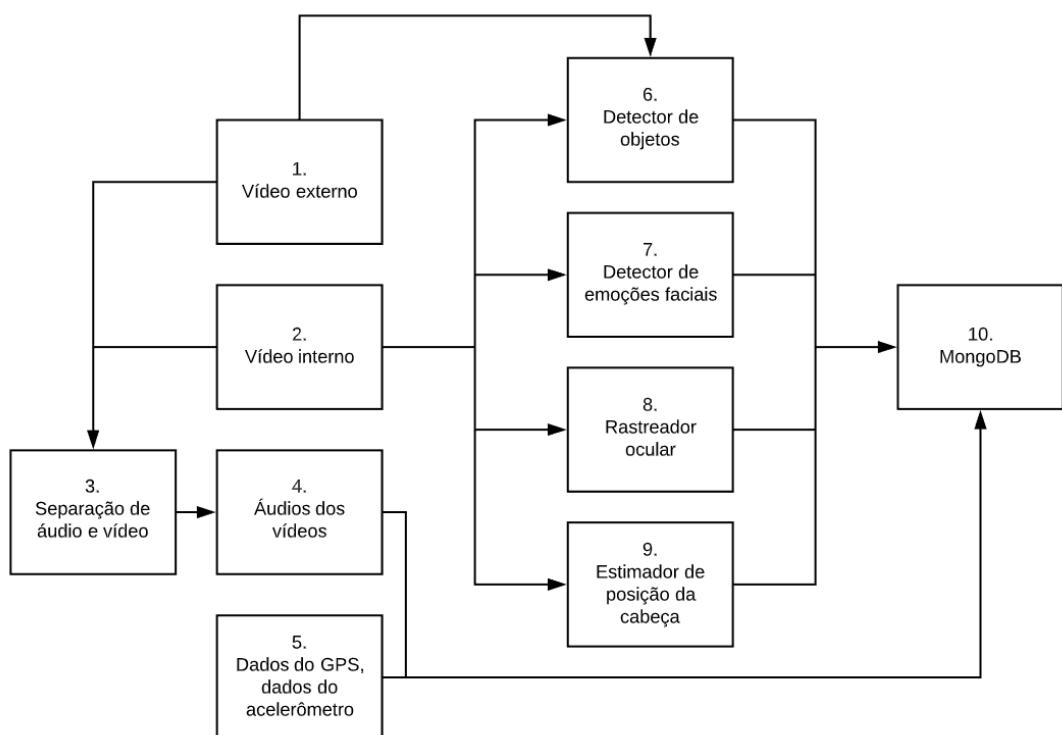


Figura 6 – Diagrama esquemático dos processamentos realizados no servidor.

4.2.1 Separação do áudio e vídeo

Quando os dados chegam ao servidor, o vídeo interno e externo conterá os áudios do ambiente, portanto é necessário separar o áudio que será inserido no *dataset*, como mostrado

na etapa 3 da Figura 6. Para isso iremos programar o node js através da biblioteca *child process* afim de executar um comando cmd, com o propósito de utilizar o *software ffmpeg*, que receberá o caminho para um vídeo no formato mp4, e irá salvar o áudio no formato mp3 em um caminho pré-definido no comando. Ao final teremos dois arquivos mp3 contendo o áudio do ambiente de diferentes posições no veículo e os vídeos internos e externos inalterados.

4.2.2 Processamento da Imagem externa

Depois de separar os áudios, o vídeo externo passará por uma detecção de objetos, como mostrado na etapa 6 da Figura 6. Antes de executar a detecção, é necessário preprocessar o vídeo, para isso realizaremos um escalonamento dos frames, diminuindo a imagem para 416 pixels de altura e largura. Essa detecção será realizada por uma rede neural profunda (dnn) em *openCV* que utiliza um modelo e os pesos do YOLOv4 pré-treinados utilizando o COCO *dataset*. Essa dnn irá receber como entrada os frames devidamente pré-processados provenientes do vídeo externo e irá gerar *bounding boxes* ao redor de objetos de trânsito, veículos e pessoas. Todo o procedimento de geração das *bounding boxes* é realizado da mesma forma como o do YOLOv4 e está explicado na Seção 3.4.

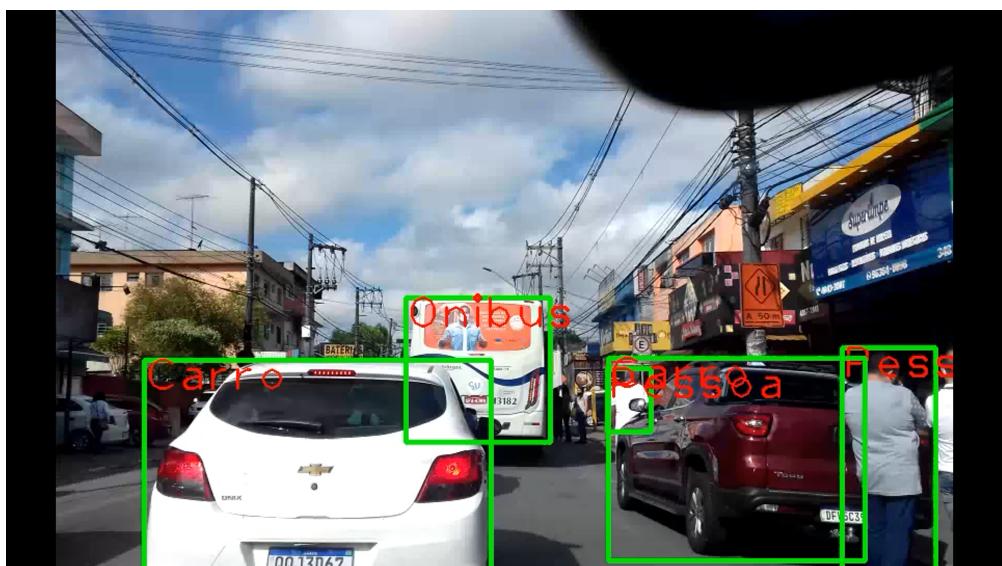


Figura 7 – A imagem mostra os bounding boxes do detector de objetos externos.

4.2.3 Processamento da Imagem interna

Para realizar o processamento do vídeo interno iremos utilizar um detector de objetos, um rastreador ocular, um estimador de posição da cabeça e um detector de emoções faciais, que irão receber como entrada os frames provenientes do vídeo interno. O detector de objetos usado será o mesmo utilizado para o processamento do vídeo externo.

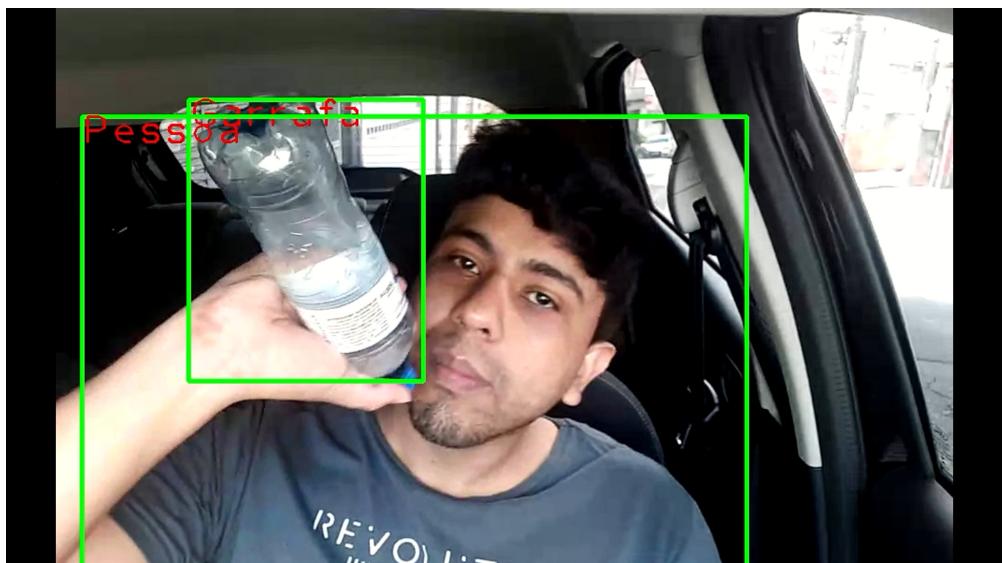


Figura 8 – A imagem mostra o bounding box do detector de objetos internos.

O rastreador ocular e o estimador de posição da cabeça tem como objetivo determinar se o motorista está olhando para direita, esquerda ou para frente, e ambos utilizam modelos pré-treinados no *ibug 300-W dataset*. Todo o processo de determinar essa direção está explicado na Seção 3.6 e 3.5, respectivamente.





Figura 9 – A imagem superior mostra o resultado do rastreamento ocular. Já a imagem inferior mostra o resultado do estimador de posição da cabeça.

O detector de emoções tem como objetivo determinar qual emoção o motorista está demonstrando, e utiliza um modelo pré-treinado no *FER-2013 dataset*. Todo o processo de como ele realiza essa determinação pode ser encontrado na Seção 3.7.

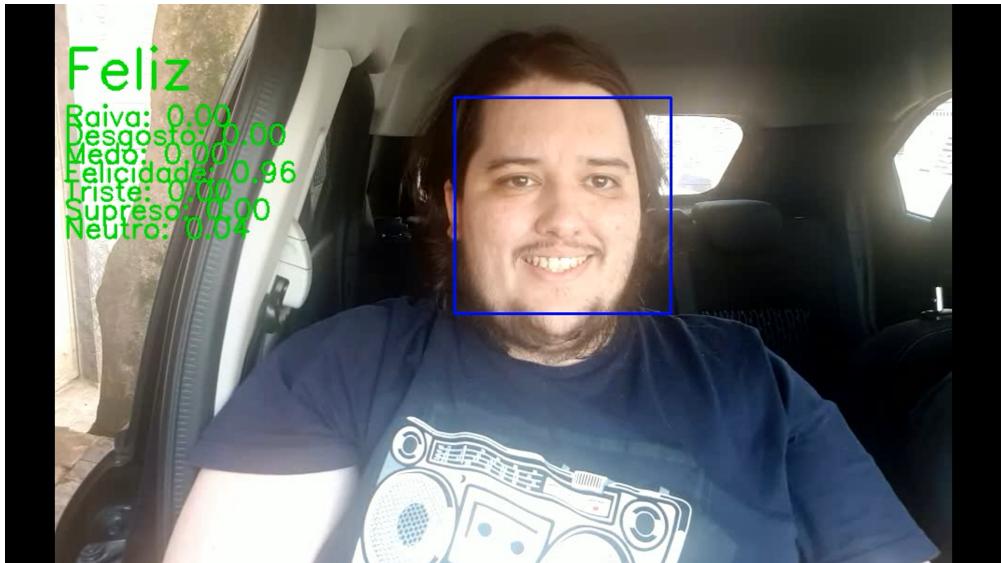
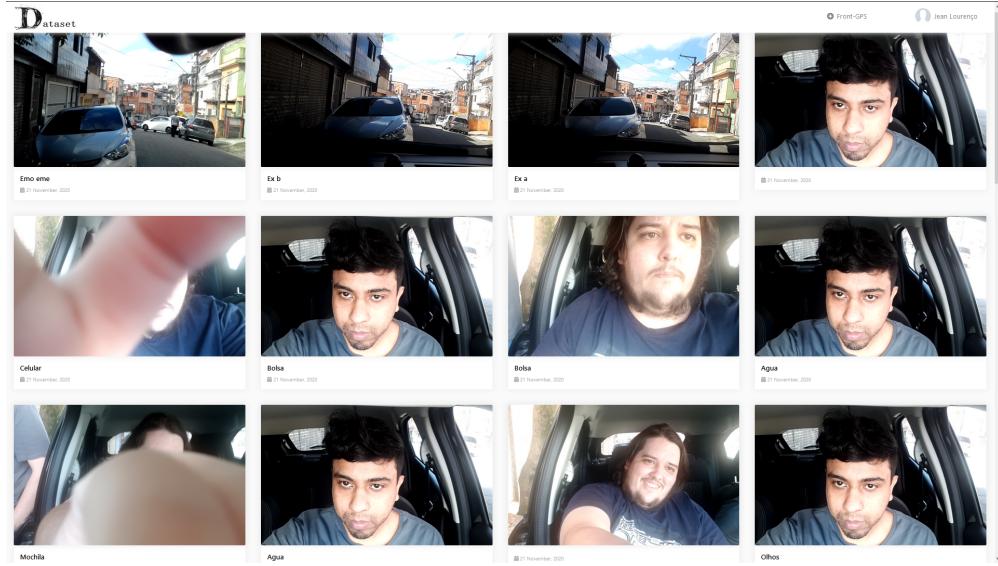


Figura 10 – A imagem mostra a detecção da emoção pelo detector, bem como a porcentagem de probabilidade de ser cada emoção.

4.3 CRIAÇÃO DO DATASET

A criação do *dataset* corresponde a última etapa dessa metodologia. Os dados provenientes da etapa 7 da Figura 5, como aceleração sem gravidade (em x, y e z), aceleração com gravidade (em x, y e z), dados do GPS (latitude, longitude, altitude e velocidade), vídeos processados (internos e externos) e vídeos sem o processamento (internos e externos), serão mostrados no formato de links para download, em uma interface juntamente com dados do usuário (idade e sexo).



Dataset

Sexo: M
Idade: 22
Published em 28 November, 2020

Dados do Video:

- Video Original [Reconhecimentos](#)
- Audio (mp3) [Audio](#)
- Gaze Grading (mp4) [Reconhecimentos](#)
- Detectões Gaze Tracking (Frame a frame) [Detecção Gaze](#)
- Facial emotion recognition (mp4) [Reconhecimentos](#)
- Detectões FER (Frame a frame) [Detecção FER](#)
- Head pose estimation (mp4) [Reconhecimentos](#)
- Detectões Head Pose (Frame a frame) [Detecção Head Pose](#)
- Object recognition (mp4) [Reconhecimentos](#)
- Detectões Object Recognition (Frame a frame) [Detecção Obj](#)
- Aceleração em X (txt) [Acelerômetro](#)
- Aceleração em Y (txt) [Acelerômetro](#)
- Aceleração em Z (txt) [Acelerômetro](#)
- Aceleração em X com gravidade (txt) [Acelerômetro G](#)
- Aceleração em Y com gravidade (txt) [Acelerômetro G](#)
- Aceleração em Z com gravidade (txt) [Acelerômetro G](#)
- Preciso (txt) [GPS](#)
- Altitude (txt) [GPS](#)
- Preciso da altitude (txt) [GPS](#)
- Heading [GPS](#)
- Latitude [GPS](#)
- Longitude (txt) [GPS](#)
- Velocidade (txt) [GPS](#)

Figura 11 – A imagem superior mostra uma lista de todos os videos presentes no dataset. A imagem inferior mostra um video selecionado e uma lista com todos seus respectivos dados para download (no caso do video não possuir algum dado, o link é nulo).

5 ESTRATÉGIA EXPERIMENTAL

Neste capítulo são apresentados os testes que foram realizados a fim de testar a metodologia proposta e validar os dados do dataset. A seguir será explicado detalhadamente o ambiente e os testes realizados.

5.1 AMBIENTE

O ambiente onde foram realizados os testes consiste em ruas e avenidas do município de São Bernardo do Campo, utilizando um carro de modelo Ford KA 2020 com o primeiro celular preso em um suporte no para-brisa com visão frontal para o motorista, e o segundo celular preso em um suporte no para-brisa com visão para a via. Para alguns testes, o ambiente foi a parte interna do veículo. Para mais detalhes sobre celular ver a Seção 3.9.1.

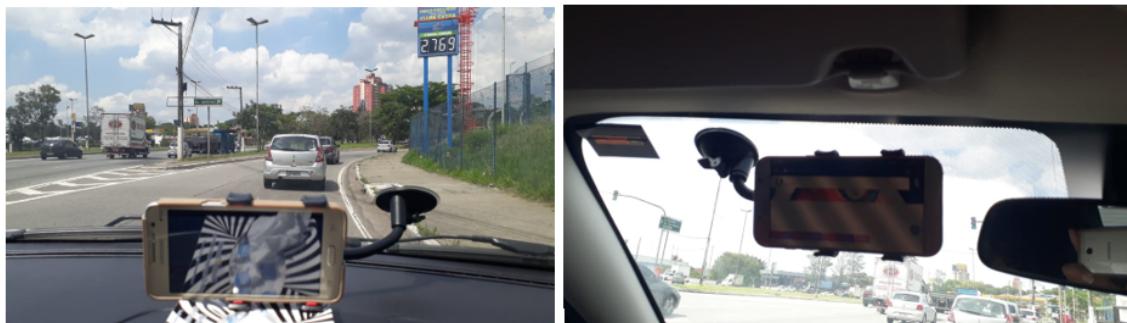


Figura 12 – A imagem da esquerda mostra a câmera com visão para a via, enquanto a da direita mostra a câmera com visão para o motorista.

5.2 TESTE DA IDENTIFICAÇÃO DE OBJETOS EXTERNOS

O primeiro teste teve como objetivo verificar a precisão da dnn responsável em detectar os objetos de fora do veículo. Este teste foi realizado com o veículo em movimento pelas ruas da cidade.

O teste foi realizado sessenta vezes para cada classe de objeto contido na Figura 13, a fim de verificar a porcentagem de acerto para cada classe, além de uma porcentagem geral de acerto da dnn de objetos externos como um todo.

Classes de objetos externos
Carro
Moto
Caminhão
Ônibus
Pessoa
Semáforo

Figura 13 – Classes dos objetos externos.

5.3 TESTE DA IDENTIFICAÇÃO DE OBJETOS INTERNOS

O segundo teste teve como objetivo verificar a precisão da dnn responsável por detectar os objetos de dentro do veículo. Este teste foi realizado utilizando a parte interna do veículo, com um dos autores deste projeto sentado no banco do motorista do veículo.

O teste foi realizado cento e cinquenta vezes para cada classe de objeto contido na Figura 14, a fim de verificar a porcentagem de acerto para cada classe, além de uma porcentagem geral de acerto da dnn interna como um todo.

Classes de objetos internos
Garrafa
Celular

Figura 14 – Classes dos objetos internos.

5.4 TESTE DE RASTREAMENTO OCULAR PARADO

O terceiro teste teve como objetivo verificar a precisão do rastreador ocular com o veículo parado. Este teste foi realizado utilizando a parte interna do veículo, com um dos autores deste projeto sentado no banco do motorista do veículo.

O teste foi realizado oitenta vezes para cada direção prevista na Figura 15, a fim de verificar a porcentagem de acerto para cada direção, além de uma porcentagem geral de acerto do rastreador ocular como um todo.

Direções do rastreamento ocular
Direita
Esquerda
Frente

Figura 15 – Direções detectadas pelo rastreador ocular.

5.5 TESTE DE RASTREAMENTO OCULAR EM MOVIMENTO

O quarto teste teve como objetivo verificar a precisão do rastreador ocular com o veículo em movimento. Este teste foi realizado utilizando a parte interna do veículo da mesma forma como descrita no teste anterior.

O teste foi realizado oitenta vezes para cada direção prevista na Figura 15, a fim de verificar a porcentagem de acerto para cada direção, além de uma porcentagem geral de acerto do rastreador ocular como um todo.

5.6 TESTE DE ESTIMAÇÃO DE POSIÇÃO DA CABEÇA PARADO

O quinto teste teve como objetivo verificar a precisão do estimador de posição da cabeça com o veículo parado. Este teste foi realizado utilizando a parte interna do veículo, com um dos autores deste projeto sentado no banco do motorista do veículo.

O teste foi realizado oitenta vezes para cada direção prevista na Figura 16, a fim de verificar a porcentagem de acerto para cada direção, além de uma porcentagem geral de acerto do estimador de posição da cabeça como um todo.

5.7 TESTE DE ESTIMAÇÃO DE POSIÇÃO DA CABEÇA EM MOVIMENTO

O sexto teste teve como objetivo verificar a precisão do estimador de posição da cabeça com o veículo em movimento. Este teste foi realizado utilizando a parte interna do veículo, com um dos autores deste projeto sentado no banco do motorista do veículo.

Direções da estimação de posição da cabeça
Direita
Esquerda
Frente

Figura 16 – Direções detectadas pelo estimador de posição da cabeça.

O teste foi realizado oitenta vezes para cada direção prevista na Figura 16, a fim de verificar a porcentagem de acerto para cada direção, além de uma porcentagem geral de acerto do estimador de posição da cabeça como um todo.

5.8 TESTE DE DETECÇÃO DE EMOÇÕES FACIAIS

O último teste teve como objetivo verificar a precisão do detector de emoções faciais. Este teste foi realizado utilizando a parte interna do veículo, com um dos autores deste projeto sentado no banco do motorista do veículo.

O teste foi realizado cinquenta vezes para cada emoção prevista na Figura 17, a fim de verificar a porcentagem de acerto para cada emoção, além de uma porcentagem geral de acerto do detector de emoções faciais como um todo.

Classes de emoções
Raiva
Medo
Desgosto
Felicidade
Surpresa
Tristeza
Neutro

Figura 17 – Classes detectadas pelo reconhecedor de emoções faciais.

6 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentados os resultados obtidos por meio dos experimentos descritos no capítulo 5, onde foram apresentados sete experimentos para os processamentos realizados. Os resultados deles são descritos nas seções abaixo.

6.1 RESULTADOS DO EXPERIMENTO 1

O Experimento 1, que teve como objetivo medir a precisão do detector de objetos externos, alcançou uma precisão média de 85%.

A Tabela 2 mostra a precisão obtida para cada classe de objetos externos. Das seis classes, apenas uma obteve uma precisão inferior a 80%, sendo ela "Moto". O principal motivo para o desempenho inferior dessa classe deve-se a obstrução por parte de outros veículos e pelo próprio motorista, pois por ser um veículo que se movimenta pelo corredor entre as faixas de trânsito, ele acaba sendo frequentemente obstruído por outros veículos, além dos próprios passageiros da moto, e por ser um veículo de pequeno porte, a obstrução acaba tendo um impacto maior do que em outros veículos, como carros, ônibus e caminhões. Apesar dessa interferência, a precisão para a classe "Moto" permaneceu acima de 75%, configurando uma boa assertividade.

Objeto	Qtd. Testes	Acertos	Precisão
Carro	60	57	95,00%
Pessoa	60	54	90,00%
Moto	60	47	78,33%
Caminhão	60	48	80,00%
Ônibus	60	48	80,00%
Semáforo	60	52	86,67%
Média		51	85,00%

Tabela 2 – Resultados do experimento 1. Fonte: Autores

6.2 RESULTADOS DO EXPERIMENTO 2

O Experimento 2, que teve como objetivo medir a precisão do detector de objetos internos, atingiu uma precisão média de 83,33%.

A Tabela 3 mostra a precisão obtida para as duas classes de objetos externos. A classe "Garrafa" obteve uma precisão elevada mesmo considerando os diferentes tamanhos e formatos do objeto, mostrando que o modelo usado foi treinado usando uma base de dados bastante diversificada para essa classe. Já a classe "Celular" obteve uma precisão abaixo de 80%, por conta da obstrução do objeto pelas mãos do motorista. Além disso, o objeto apresentou dificuldades em ser reconhecido quando aparecia lateralmente, problema que pode ser resolvido com a adição de mais imagens laterais do celular na base de dados usada no treinamento do modelo, aumentando assim a precisão para essa classe.

Objeto	Qtd. Testes	Acertos	Precisão
Garrafa	150	132	88,00%
Celular	150	118	78,67%
Média		125	83,33%

Tabela 3 – Resultados do experimento 2. Fonte: Autores.

6.3 RESULTADOS DO EXPERIMENTO 3

O Experimento 3, que teve como objetivo medir a precisão do rastreador ocular com o veículo parado, obteve uma precisão média de 88,75%.

A Tabela 4 mostra a precisão obtida para cada direção do rastreador ocular com o veículo parado. A direção "Direita" foi a única a ficar com menos de 80% de precisão. O motivo para o desempenho inferior dessa direção é a luminosidade inferior do lado direito do rosto do motorista, isso ocorreu devido a iluminação no momento do experimento vir da janela que fica do lado esquerdo do motorista, o que explica também a alta precisão da classe "Esquerda", pois com maior iluminação o rastreador tem mais facilidade em localizar o olho do motorista.

Direção	Qtd. Testes	Acertos	Precisão
Direita	80	62	77,50%
Esquerda	80	77	96,25%
Frente	80	74	92,50%
Média		71	88,75%

Tabela 4 – Resultados do experimento 3. Fonte: Autores.

6.4 RESULTADOS DO EXPERIMENTO 4

O Experimento 4, que teve como objetivo medir a precisão do rastreador ocular com o veículo em movimento, obteve uma precisão média de 71,25%.

A Tabela 5 mostra a precisão obtida para cada direção do rastreador ocular com o veículo em movimento. A precisão para todas as direções diminui quando comparadas com as precisões do veículo parado. Isso ocorre porque a câmera acaba balançando quando o veículo está em movimento, fazendo com que as imagens captadas por ela fiquem desfocadas, o que acaba dificultando a localização do olho pelo rastreador, impactando em todas as precisões.

Direção	Qtd. Testes	Acertos	Precisão
Direita	80	53	66,25%
Esquerda	80	60	75,00%
Frente	80	58	72,50%
Média		57	71,25%

Tabela 5 – Resultados do experimento 4. Fonte: Autores.

6.5 RESULTADOS DO EXPERIMENTO 5

O Experimento 5, que teve como objetivo medir a precisão do estimador de posição da cabeça com o veículo parado, obteve uma precisão média de 80%.

A Tabela 6 mostra a precisão obtida para cada direção do estimador de posição da cabeça com o veículo parado. As três direções obtiveram precisões próximas a 80%, isso pode ser explicado pelo estimador ter dificuldade em determinar a direção da cabeça quando o motorista está olhando de forma sutil para a esquerda ou direita, pois acaba ficando na divisa do que ele define como frente, direita e esquerda, como é possível ver na Figura 18.

Direção	Qtd. Testes	Acertos	Precisão
Direita	80	64	80,00%
Esquerda	80	65	81,25%
Frente	80	63	78,75%
Média		64	80,00%

Tabela 6 – Resultados do experimento 5. Fonte: Autores.

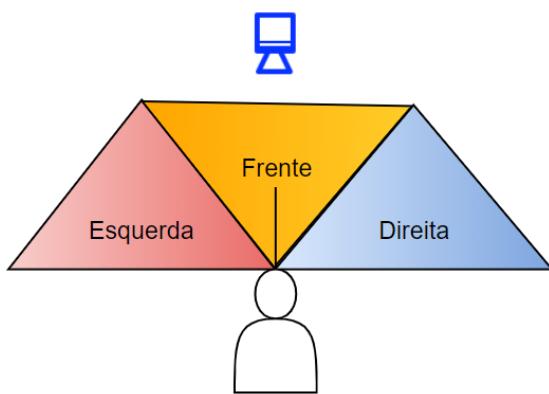


Figura 18 – Imagem ilustrativa da divisa entre as direções. Fonte: Autores.

6.6 RESULTADOS DO EXPERIMENTO 6

O Experimento 6, que teve como objetivo medir a precisão do estimador de posição da cabeça com o veículo em movimento, obteve uma precisão média de 82,50%.

A Tabela 7 mostra a precisão obtida para cada direção do estimador de posição da cabeça com o veículo em movimento. As precisões para as direções "Direita" e "Esquerda" aumentam em relação a precisão das mesmas com o veículo parado, enquanto a precisão da direção "Frente" diminui. Isso ocorre por conta da divisa entre as direções e a imprecisão advinda do desfoco das imagens, pois para o estimador a área considerada como frente é menor que as áreas consideradas como direita e esquerda, logo quando o motorista olha de forma sutil para o lado esquerdo ou direito, a imprecisão faz com que o estimador acerte a direção que antes não acertava, mas por consequência acaba errando quando o motorista está olhando para frente, um pouco antes do divisor.

Direção	Qty. Testes	Acertos	Precisão
Direita	80	70	87,50%
Esquerda	80	69	86,25%
Frente	80	59	73,75%
Média		66	82,50%

Tabela 7 – Resultados do experimento 6. Fonte: Autores.

6.7 RESULTADOS DO EXPERIMENTO 7

O último experimento da metodologia, que teve como objetivo medir a precisão do detector de emoções faciais, obteve uma precisão média de 82%.

A Tabela 8 mostra a precisão obtida para cada classe do detector de emoções faciais. Das sete emoções apenas duas obteram um valor abaixo de 80% de precisão, sendo elas "Triste" e "Surpreso". O principal motivo para o desempenho inferior dessas emoções deve-se as diversas variações de expressões faciais relacionadas a essas emoções, o detector está limitado na detecção de expressões específicas, como, por exemplo, no nosso detector a emoção de "Tristeza" está associada aos cantos dos lábios para baixo, enquanto a emoção de "Surpresa" está associada a boca aberta verticalmente e olhos arregalados.

Emoção	Qtd. Testes	Acertos	Precisão
Feliz	50	47	94,00%
Triste	50	37	74,00%
Desgosto	50	43	86,00%
Medo	50	41	82,00%
Surpreso	50	36	72,00%
Raiva	50	40	80,00%
Neutro	50	43	86,00%
Média		41	82,00%

Tabela 8 – Resultados do experimento 7. Fonte: Autores.

A adição de uma maior variedade de expressões faciais para as emoções "Triste" e "Surpreso" na base de dados usada no treinamento do detector, acarretaria em uma melhoria na precisão dessas emoções.

6.8 DATASET

Uma parte dos dados coletados durante os experimentos constituem um *dataset*, que pode ser acessado em: <<https://drive.google.com/drive/folders/1QFnsJGvZSvXvGIrqYNNOD4GB83qnFYusp=sharing>>.

Esse *dataset* possui 289 dados divididos em 3 categorias:

- 202 textos;
- 59 vídeos;
- 28 áudios;

7 CONCLUSÃO

Este trabalho apresentou uma proposta de execução de um ciclo completo de produção de um *dataset* que armazena o comportamento de motoristas de acordo com o contexto do ambiente, desde a coleta dos dados (Seção 4.1) até a validação do *dataset*, através dos experimentos definidos no capítulo 5.

Este projeto traz consigo três contribuições: uma forma de capturar dados para a criação de um *dataset* através de aparelhos celulares; um sistema para a criação de um *dataset* que se utiliza de detectores de objetos para classificação tanto de objetos externos quanto internos ao veículo, com precisões médias de 85% e 83,33%, respectivamente, de um detector de emoções faciais para determinar a emoção que o motorista está sentindo com precisão média de 82%, de um rastreador ocular, com precisão média de 88,75% com veículo parado e 71,25% com veículo em movimento, e de um estimador de posição da cabeça, com precisão média de 80% com veículo parado e 82,50% com veículo em movimento, para determinar a direção que o motorista está olhando e para onde a cabeça dele está virada, respectivamente; por fim, contribui com um novo *dataset* com vídeos, áudios e dados provenientes de diversos sensores dos celulares para reconhecimento do comportamento do motorista.

REFERÊNCIAS

- ALEMDAG, E.; CAGILTAY, K. A systematic review of eye tracking research on multimedia learning. **Comput. Educ.**, v. 125, p. 413–428, 2018.
- ANEJA, D. et al. Modeling stylized character expressions via deep learning. In: SPRINGER. **Asian Conference on Computer Vision**. [S.l.], 2016. p. 136–153.
- BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. **arXiv preprint arXiv:2004.10934**, 2020.
- BULAT, A.; TZIMIROPOULOS, G. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230, 000 3d facial landmarks). **CoRR**, abs/1703.07332, 2017. Disponível em: <<http://arxiv.org/abs/1703.07332>>.
- BUTLER, D. J. et al. A naturalistic open source movie for optical flow evaluation. In: SPRINGER. **European conference on computer vision**. [S.l.], 2012. p. 611–625.
- DAI, J. et al. R-fcn: Object detection via region-based fully convolutional networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2016. p. 379–387.
- EVERINGHAM, M. et al. The pascal visual object classes challenge: A retrospective. **International journal of computer vision**, Springer, v. 111, n. 1, p. 98–136, 2015.
- FANELLI, G. et al. Random forests for real time 3d face analysis. **International journal of computer vision**, Springer, v. 101, n. 3, p. 437–458, 2013.
- GEIGER, A.; LENZ, P.; URTASUN, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In: IEEE. **2012 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.], 2012. p. 3354–3361.
- GU, J. et al. Dynamic facial analysis: From bayesian filtering to recurrent neural network. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 1548–1557.
- HE, K. et al. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.
- KAZEMI, V.; SULLIVAN, J. One millisecond face alignment with an ensemble of regression trees. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2014. p. 1867–1874.
- KRAFKA, K. et al. Eye tracking for everyone. **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, p. 2176–2184, 2016.

- LIN, T.-Y. et al. Focal loss for dense object detection. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2017. p. 2980–2988.
- _____. Microsoft coco: Common objects in context. In: SPRINGER. **European conference on computer vision**. [S.l.], 2014. p. 740–755.
- LIU, W. et al. Ssd: Single shot multibox detector. In: SPRINGER. **European conference on computer vision**. [S.l.], 2016. p. 21–37.
- Lucey, P. et al. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: **2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops**. [S.l.: s.n.], 2010. p. 94–101.
- LYONS, M.; KAMACHI, M.; GYOBA, J. The Japanese Female Facial Expression (JAFFE) Dataset. Zenodo, abr. 1998. The images are provided at no cost for non-commercial scientific research only. If you agree to the conditions listed below, you may request access to download. Disponível em: <<https://doi.org/10.5281/zenodo.3451524>>.
- MINAEE, S.; ABDOLRASHIDI, A. Deep-emotion: Facial expression recognition using attentional convolutional network. **arXiv preprint arXiv:1902.01019**, 2019.
- REDMON, J. et al. You only look once: Unified, real-time object detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 779–788.
- REDMON, J.; FARHADI, A. Yolo9000: better, faster, stronger. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 7263–7271.
- _____. Yolov3: An incremental improvement. **arXiv**, 2018.
- REN, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2015. p. 91–99.
- RUIZ, N.; CHONG, E.; REHG, J. M. Fine-grained head pose estimation without keypoints. **2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)**, p. 2155–215509, 2018.
- RUSSAKOVSKY, O. et al. Imagenet large scale visual recognition challenge. **International journal of computer vision**, Springer, v. 115, n. 3, p. 211–252, 2015.
- SCHARSTEIN, D. et al. High-resolution stereo datasets with subpixel-accurate ground truth. In: SPRINGER. **German conference on pattern recognition**. [S.l.], 2014. p. 31–42.
- TAN, M. et al. Mnasnet: Platform-aware neural architecture search for mobile. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p.

2820–2828.

TARNOWSKI, P. et al. Emotion recognition using facial expressions. In: **ICCS**. [S.l.: s.n.], 2017. p. 1175–1184.

Vicente, F. et al. Driver gaze tracking and eyes off the road detection system. **IEEE Transactions on Intelligent Transportation Systems**, v. 16, n. 4, p. 2014–2027, 2015.

WOMG, A. et al. Tiny ssd: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. In: IEEE. **2018 15th Conference on Computer and Robot Vision (CRV)**. [S.l.], 2018. p. 95–101.

XIAO, J. et al. Sun database: Exploring a large collection of scene categories. **International Journal of Computer Vision**, Springer, v. 119, n. 1, p. 3–22, 2016.

YANG, T.-Y. et al. Fsa-net: Learning fine-grained structure aggregation for head pose estimation from a single image. **2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, p. 1087–1096, 2019.

ZHOU, B. et al. Places: A 10 million image database for scene recognition. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 40, n. 6, p. 1452–1464, 2017.

ZHU, X. et al. Face alignment across large poses: A 3d solution. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 146–155.