



**INSTITUTO
FEDERAL**
Brasília

Instituto Federal de Educação, Ciência e Tecnologia de Brasília
Campus Taguatinga

**DESENVOLVIMENTO DE PROTÓTIPO PARA REPRESENTAR UMA CADEIRA DE
RODAS MOTORIZADA COM SISTEMA DE SEGMENTAÇÃO DE CAMINHOS**

Por

FILIPE CAVALCANTE DOS SANTOS E RAFAEL WAITI GITIRANA UMETSU

Trabalho de Graduação

BRASÍLIA/2023

Filipe Cavalcante dos Santos e Rafael Waiti Gitirana Umetsu

**DESENVOLVIMENTO DE PROTÓTIPO PARA REPRESENTAR UMA
CADEIRA DE RODAS MOTORIZADA COM SISTEMA DE
SEGMENTAÇÃO DE CAMINHOS**

*Trabalho apresentado ao Curso de Ciência da Computação
do Instituto Federal de Educação, Ciência e Tecnologia de
Brasília como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação.*

Orientador: Dr. Fabiano Cavalcanti Fernandes

Co-Orientador: Dr. Lucas Paes Moreira

BRASÍLIA

2023

Ficha de identificação da obra elaborada pelo bibliotecário
Daniel Cerqueira Costa (CRB1-3256)

Santos, Filipe Cavalcante dos

S237c

Desenvolvimento de protótipo para representar uma cadeira de rodas motorizada com sistema de segmentação de caminhos / Filipe Cavalcante dos Santos e Rafael Waiti Gitirana Umetsu. Brasília-DF, 2023.

55 f. : il.

Monografia (Bacharel em Ciência da Computação) – Instituto Federal de Brasília – Campus Taguatinga, 2023.

Orientador: Dr. Fabiano Cavalcanti Fernandes.

Co-Orientador: Dr. Lucas Paes Moreira.

Inclui referências.

1. Software - Desenvolvimento. 2. Pessoas com deficiência - Aspectos sociais. 3. Inclusão social. 4. Correntes contínuas. I. Título. II. Umetsu, Rafael Waiti Gitirana. III. Fernandes, Fabiano Cavalcanti. IV. Moreira, Lucas Paes. V. Instituto Federal de Educação, Ciência e Tecnologia de Brasília.

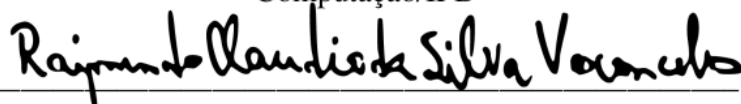
CDU 004-056.26

Trabalho de Graduação apresentado por **Filipe Cavalcante dos Santos e Rafael Waiti Gitirana Umetsu** ao curso de Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia de Brasília sob o título **Desenvolvimento de Protótipo para Representar uma Cadeira de Rodas Motorizada com Sistema de Segmentação de Caminhos**, orientado pelo **Prof. Dr. Fabiano Cavalcanti Fernandes** e aprovado pela banca examinadora formada pelos professores:



Prof. Fabiano Cavalcanti Fernandes

Computação/IFB



Prof. Raimundo Cláudio da Silva Vasconcelos

Computação/IFB



Prof. Thiago Batista Amorim

Computação/IFB

BRASÍLIA

2023

Resumo

O presente trabalho tem como objetivo abordar a necessidade de um sistema de detecção de perigo para as pessoas com deficiência (PCDs) que utilizam cadeira de rodas motorizada. Segundo o último censo demográfico, o Brasil tem aproximadamente 13 milhões de PCDs que dependem de cadeiras de rodas para a locomoção. A segurança e autonomia das pessoas com deficiência são elementos essenciais para a sua qualidade de vida e inclusão na sociedade, no entanto, esses indivíduos podem enfrentar diversos desafios e riscos no seu cotidiano, principalmente quando se trata da identificação e prevenção de perigos iminentes. Por meio deste estudo, foi desenvolvido um protótipo para representar uma cadeira de rodas motorizada com um sistema de detecção de perigo para PCDs que utilizam cadeira de rodas motorizada. Foram utilizadas técnicas e tecnologias, como câmeras, computador embarcado, motores de corrente contínua (CC), técnicas de transmissão, e algoritmos de processamento de imagem como modelo de aprendizado de máquina (ML), para se locomover, identificar e alertar sobre situações de perigo nas proximidades dessas pessoas, como obstáculos, desníveis e tráfego intenso. Os desafios encontrados incluem a integração adequada dos componentes, o cálculo mais apurado das tensões e correntes necessárias para alimentação dos sistemas, melhoria no peso da base onde se encontra a configuração elétrica e mecânica do protótipo e também melhorias na implementação dos algoritmos para aperfeiçoar a resposta entre o controle humano e a resposta do código, como também um possível aprimoramento de performance em quadros por segundo (qps). O presente estudo fornece *insights* e diretrizes para futuras pesquisas e desenvolvimentos na área de sistemas de detecção de perigo para cadeiras de rodas motorizadas, com o objetivo de melhorar a segurança e a qualidade de vida das PCDs. O projeto foi finalizado com a criação de um protótipo que simula uma cadeira de rodas motorizada, incorporando uma câmera. Esse protótipo utiliza um modelo de segmentação capaz de diferenciar entre caminhos e não caminhos em ruas. O modelo de segmentação alcançou uma acurácia notável na área de atenção, registrando 0.9287 ± 0.03656 no conjunto de teste, além disso, o desempenho médio do sistema atingiu 6 qps em um Raspberry Pi 3B+, demonstrando uma velocidade de processamento eficiente.

Palavras-chave: Pessoas com Deficiência, Modelo de Aprendizado de Máquina, Cadeira de Rodas Motorizada, Sistema de Detecção de Perigo

Abstract

The present work aims to approach the need of a danger detection system for *people with disabilities* (PWD) who use motorized wheelchairs. According to the last Brazilian demographic census, there are approximately 13 million PWD in Brazil who depend on wheelchairs for their locomotion. Safety and autonomy of PWD are essential for their quality of life and inclusion in society, however, these individuals might face several challenges and risks in their day-to-day lives, especially when it comes to identifying and preventing imminent dangers. Through this study, a prototype was developed to represent a motorized wheelchair with a danger detection system for PWD who use motorized wheelchairs. Techniques and technologies were used for that, such as cameras, an embedded computer, direct current engines, transmission techniques and image processing algorithms such as ML to move, identify and alert about dangerous situations in the individual's surroundings such as obstacles, unevenness and intense traffic. The challenges found include adequate components integration, sharper calculations of tensions and currents needed to feed the systems, improving the base weight where the prototype's electric and mechanical settings are positioned and improving the implementation of algorithm to better the response between human control and the code, as well as a possible performance improvement in *frames per second* (fps). This study provides insights and guidelines for future research and developments in the danger detection systems for motorized wheelchairs, with the goal of improving safety and quality of life for PWD. The project was concluded with the creation of a prototype which simulates a motorized wheelchair, incorporating a camera. This prototype uses a segmentation model able to tell pathways and non-pathways apart in the street. The segmentation model reached a notable accuracy in the attention zone, marking 0.9287 ± 0.03656 in the test set; besides, the average system performance reached up to 6 fps in a Raspberry Pi 3B+, demonstrating an efficient processing speed.

Keywords: People with disabilities, Machine Learning Model, Wheelchair, Danger Detection System

Lista de Figuras

2.1	Um dos primeiros protótipos de cadeira de rodas patenteados com <i>joystick</i> utilizando transmissão por corrente	15
2.2	Sistema motorreductor direto nas rodas de tração	15
2.3	Sistema motorreductor direto nas rodas de tração com melhorias estéticas, ergonômicas e aumento do número de rodas de estabilidade	16
2.4	Cadeira transmissão flexível, (a) - <i>joystick</i> , (b) - motor, (c) - corrente ou correia para transmitir o movimento do motor para as rodas traseiras	16
2.5	Transmissão por engrenagem no eixo, (a) - <i>joystick</i> , (b) - Motor com engrenagem, (c) - Engrenagem acoplada diretamente ao eixo da roda para transmitir o movimento do motor	17
2.6	Transmissão por atrito, (a) - <i>joystick</i> , (b) - Motor realizando atrito com as rodas traseiras	17
2.7	Motor acoplado no eixo na roda, (a) - <i>joystick</i> , (b) - Motor acoplado no eixo roda	18
2.8	Disposição dos eletrodos no projeto SIAMO	19
2.9	Protótipo VAHM	19
2.10	Partes de um Motor CC	20
2.11	Diagrama de um circuito ponte H	21
2.12	Abordagem clássica (a) versus <i>Deep Learning</i> (b)	23
2.13	Estágios do Modelo de ML	23
2.14	Características extraídas com a profundidade das camadas do modelo	24
2.15	Exemplo de segmentação	25
2.16	Segmentação de caminho - DeepLabV3 + MobileNetV2 no dataset Cityscapes .	26
2.17	Estrutura da arquitetura DeepLabV3+	27
2.18	Diferença de convolução com dilatação = 1 e dilatação = 2	27
2.19	Diferenças de arquiteturas	28
2.20	Hold out	29
2.21	Validação cruzada	29
2.22	Interseção sobre União exemplo	30
3.1	Motor CC MY1016	32
3.2	Engrenagem para corrente com diâmetro de 180 mm	32
3.3	Engrenagem para corrente com diâmetro de 63.5 mm	33
3.4	Módulo Driver Ponte H Dupla - BTS7960	33
3.5	Montagem do Driver	34
3.6	Montagem de toda a configuração fora da cadeira de rodas para teste	35
3.7	Redutor de tensão	35

3.8	Raspberry Pi 3 Model B+	36
3.9	DeepLabV3+ - modelo A	37
3.10	DeepLabV3+ - modelo B	39
3.11	Validação cruzada	40
3.12	Área de atenção. verde caminho, vermelho não caminho e azul ignorado no aprendizado	41
4.1	Módulos ponte H ligados em paralelo com fios de 2.5milímetro (mm) ²	42
4.2	Polia para correia	43
4.3	Roda dentada para correia	43
4.4	Adaptação do motor	44
4.5	Adaptação do motor	44
4.6	Protótipo - parte frontal	44
4.7	Protótipo - parte superior	44
4.8	Protótipo - parte inferior	44
4.9	Área predita verde caminho, vermelho não caminho e azul ignorado no aprendizado	45
4.10	Distribuição dos resultados nas 5 partições	47
4.11	Área de atenção predita verde caminho, vermelho não caminho e azul ignorado no aprendizado	49
4.12	Comparação entre o modelo A e o modelo B em uma rua	50
4.13	Comparação entre o modelo A e o modelo B em um corredor	51

Lista de Tabelas

3.1	Estimativa de custo da instrumentação de uma cadeira de rodas motorizada . . .	41
4.1	Resultado dos modelos de validação cruzada	48
4.2	Resultado dos modelos de busca de hiperparâmetros	48
4.3	Resultado do modelo final	49
4.4	Resultado do modelo final na área de atenção	49

Lista de Acrônimos

A	ampere	32
CC	corrente contínua	19
fps	<i>frames per second</i>	46
IBGE	Instituto Brasileiro de Geografia e Estatística	11
IC	interface de Controle	33
IA	Inteligência Artificial	12
IdC	internet das coisas	22
mIOU	interseção sobre união	46
m	metros	31
mA	miliampere	33
mm	milímetro	32
ML	modelo de aprendizado de máquina	13
PWM	modulação por largura de pulso	20
PCD	peessoa com deficiência	11
PWD	<i>people with disabilities</i>	4
qps	quadros por segundo	3
RPM	rotação por minuto	32
SSH	<i>secure shell</i>	36
SUS	Sistema Único de Saúde	11
Vdc	tensão de corrente contínua	33
V	<i>volt</i>	20
W	<i>watt</i>	32

Sumário

1	Introdução	11
1.1	Formulação do problema	11
1.2	Motivação	12
1.3	Objetivo geral	13
1.4	Objetivos específicos	13
1.5	Descrição dos capítulos	14
2	Referencial Teórico	15
2.1	Transmissão	16
2.2	Interfaces de controle	18
2.3	Motor de Corrente contínua	19
2.3.1	Funcionamento de um Motor CC	20
2.4	Modulação por largura de pulso	20
2.5	Ponte H	21
2.6	Sensores	22
2.7	Sistemas embarcados	22
2.8	<i>Deep Learning</i> - Aprendizagem Profunda	22
2.8.1	Abordagem clássica versus <i>Deep Learning</i>	23
2.8.1.1	Abordagem clássica - ML	23
2.8.1.2	Abordagem <i>end-to-end</i> - <i>Deep Learning</i>	24
2.8.2	Visão computacional	25
2.8.2.1	Classificação de Imagem	25
2.8.2.2	Detecção de objetos	25
2.8.2.3	Segmentação de imagem	25
2.8.3	DeepLabV3+	26
2.8.4	Ajuste de hiperparâmetros	28
2.8.5	Divisão do conjunto de dados	28
2.8.5.1	Hold out	28
2.8.5.2	Validação cruzada	29
2.8.5.3	Interseção sobre União (IoU)	30
3	Material e Métodos	31
3.1	Sistema mecânico	31
3.2	Sistema eletro-eletrônico	33
3.3	Sistema Computacional	36
3.4	Área de atenção	40
3.5	Custo do projeto	41

4	Resultados e Discussão	42
4.1	Sistema eletro mecânico	42
4.2	Sistema Computacional	45
4.3	Integração dos sistemas	51
5	Conclusão e trabalhos futuros	52
	Referências	53

1

Introdução

1.1 Formulação do problema

As cadeiras de rodas desempenham um papel essencial na mobilidade das pessoas com deficiência (PCDs) que estão paralisadas totalmente ou da cintura para baixo. De acordo com o censo demográfico de 2010, realizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE), o número de PCDs no Brasil aumentou significativamente chegando a cerca de 45 milhões de indivíduos. Entre eles, aproximadamente 13 milhões são deficientes motores, representando cerca de 28% do total de PCDs do país (IBGE, 2012).

Esses números destacam a importância das cadeiras de rodas como meio de locomoção para pessoas com limitações físicas. Além de proporcionar autonomia e independência, esses dispositivos contribuem para a inclusão social e a melhoria da qualidade de vida das PCDs. Reconhecer a necessidade de acessibilidade e garantir o acesso adequado a cadeiras de rodas e outros equipamentos assistivos é fundamental para promover a igualdade de oportunidades e a participação plena dessas pessoas na sociedade.

No Brasil, por meio do Sistema Único de Saúde (SUS), todas as pessoas têm o direito de receber uma cadeira de rodas manual ou motorizada independentemente da renda familiar. Esse benefício destina-se aqueles que realmente possuem alguma deficiência física, desde que haja comprovação por meio de laudo médico. Esse direito é garantido pela Lei nº 8.080, de 16/09/90, decretada na Constituição Federal e na Lei Orgânica de Saúde que reconhecem o atendimento integral à saúde como um direito de cidadania. Esse atendimento abrange a atenção primária, secundária e terciária, e inclui o fornecimento de equipamentos necessários para promoção, prevenção, assistência e reabilitação (FREEDOM, 2020).

No entanto, é importante ressaltar que as cadeiras de rodas motorizadas são disponibilizadas apenas após uma avaliação multidisciplinar que considera aspectos físicos, mentais e ambientais do indivíduo. Portanto, essas cadeiras são prescritas principalmente para pessoas que não têm condições de conduzir manualmente o equipamento (FREEDOM, 2020).

Assim, o SUS busca garantir o acesso equitativo às cadeiras de rodas, considerando as necessidades individuais e promovendo a inclusão e a mobilidade das pessoas com deficiência física.

No caso de indivíduos que não foram contemplados com o apoio governamental, mas desejam adquirir uma cadeira de rodas motorizada para melhorar sua mobilidade, é necessário desembolsar uma quantia significativa. No mercado brasileiro, atualmente existem diversas marcas que comercializam cadeiras de rodas motorizadas, porém os preços situam-se na faixa de R\$7 mil a R\$18 mil. Para famílias de baixa renda esse valor torna-se inviável para aquisição.

Apesar de existirem melhorias na mobilidade para pessoas com deficiência motora, é importante ressaltar que para aqueles que conseguem se beneficiar do apoio governamental ou possuem recursos para adquirir cadeiras de rodas motorizadas, tais equipamentos não são capazes de prevenir acidentes, especialmente em locais onde a acessibilidade para cadeirantes ainda não tenha sido contemplada com soluções adequadas. Segundo um estudo feito por SILVEIRA (2019), constatou-se que 79% dos participantes relataram ter sofrido acidentes envolvendo cadeiras de rodas. Dentre os acidentes mais frequentes destacam-se a quebra das rodas, colisões com obstáculos e o aprisionamento das rodas em buracos.

Em um estudo adicional realizado por UMMAT; KIRBY (1994) nos Estados Unidos, abrangendo o período de 1986 a 1990, verificou-se uma média anual de 36.559 acidentes não fatais envolvendo cadeiras de rodas. Dessas ocorrências, 41,4% foram atribuídas a causas secundárias, com 13,6% relacionadas a móveis, 5,1% a escadas e 5,5% a rampas. Além disso, de 156 acidentes envolvendo pessoas e cadeiras de rodas, 25% ocorreram devido a colisões com a cadeira, 17,9% foram atropelamentos causados por cadeiras de rodas e 26,7% ocorreram devido a tropeços na cadeira de rodas.

1.2 Motivação

Com os avanços tecnológicos atuais no campo da Inteligência Artificial (IA), torna-se indispensável a criação de soluções que visem aprimorar a acessibilidade e a segurança das PCDs. Entre essas soluções, destacam-se cadeiras de rodas autônomas e cadeiras com mecanismos de controle por gestos, como exemplificado pela cadeira de rodas utilizada pelo renomado cientista Stephen Hawking ou mesmo cadeiras controladas pelo movimento dos olhos. Essas inovações representam uma oportunidade significativa para melhorar a qualidade de vida e a independência das PCDs promovendo uma inclusão mais ampla e igualitária em diversos aspectos do dia-a-dia.

“A deficiência está na condição humana, ao longo da vida qualquer um poderá ter alguma limitação, seja ela permanente ou temporária, principalmente na velhice, onde as limitações influenciam na funcionalidade do corpo, e dificultam as atividades cotidianas” (SILVEIRA, 2019).

A vida cotidiana de uma pessoa exige uma imensa quantidade de conhecimento sobre o mundo. Grande parte desse conhecimento é subjetivo e intuitivo, e, portanto, difícil de articular de maneira formal. Computadores precisam capturar esse mesmo conhecimento para se comportarem de maneira inteligente. Um dos principais desafios na inteligência artificial é como inserir esse conhecimento informal em um computador (GOODFELLOW; BENGIO; COURVILLE, 2016).

As dificuldades enfrentadas por sistemas que dependem de conhecimento pré-programado sugerem que os sistemas de IA precisam ter a capacidade de adquirir seu próprio conhecimento, extraindo padrões a partir de dados brutos. Essa capacidade é conhecida como modelo de aprendizado de máquina (ML). A introdução do ML permitiu que os computadores abordassem problemas envolvendo conhecimento do mundo real e tomassem decisões que parecem subjetivas (GOODFELLOW; BENGIO; COURVILLE, 2016).

Com a inserção de ML nas tecnologias que envolvem a construção de soluções para PCDs, pode haver um grande avanço na qualidade de vida e segurança dessas pessoas. Com o paradigma de ML, tem-se a capacidade de criar tecnologias que aprendem a resolver problemas em relação ao mundo e as necessidades humanas. Encontram-se exemplos em diversas áreas, como sistemas de recomendação de filmes, produtos, cursos, detectando fraudes em cartões e até diferenciando plantas.

Em âmbito nacional, a disponibilidade de soluções de baixo custo para cadeiras de rodas motorizadas equipadas com sistemas de detecção de perigos é limitada. Isso cria uma área de pesquisa e desenvolvimento extremamente promissora. Diante dessa realidade, torna-se imperativo explorar essa demanda e atender a essa parcela da sociedade, proporcionando soluções adequadas e acessíveis para as PCDs que possuam habilidades manuais nas mãos para a condução da cadeira de rodas através de um *joystick*.

1.3 Objetivo geral

O objetivo principal deste projeto é desenvolver um modelo em escala reduzida que representa uma cadeira de rodas motorizada, o qual será integrado a um ML para a prevenção de incidentes relacionados à presença de obstáculos no trajeto da cadeira de rodas.

Esse modelo em escala reduzida será utilizado como base para a implementação de algoritmos de ML, com o propósito de identificar potenciais perigos e riscos de segurança durante a utilização de uma cadeira de rodas motorizada.

Ao integrar o modelo de ML à cadeira de rodas, espera-se obter um sistema capaz de detectar e alertar sobre situações de risco, proporcionando maior segurança e autonomia aos usuários. Dessa forma, pretende-se explorar o potencial das tecnologias de aprendizado de máquina para aprimorar a segurança e a eficiência no uso de cadeiras de rodas motorizadas.

1.4 Objetivos específicos

- Realizar o projeto e desenvolvimento do sistema mecânico e elétrico.
- Analisar a eficácia e desempenho do modelo de aprendizado de máquina no problema em questão e documentar os resultados obtidos.

1.5 Descrição dos capítulos

1. Introdução

- Breve introdução ao trabalho e explicação da relevância do tema escolhido.
- Justificativa para o estudo proposto.
- Apresentação dos objetivos que foram seguidos.

2. Referencial Teórico

- Apresentação dos principais conceitos.

3. Material e Métodos

- Descrição detalhada dos métodos e técnicas utilizados na construção dos sistemas mecânico, eletro-eletrônico e computacional.

4. Resultados e Discussão

- Análise dos resultados, considerando suas implicações teóricas e práticas.
- Apresentação e interpretação dos resultados obtidos durante a pesquisa.
- Exploração de possíveis limitações do estudo.

5. Conclusão e trabalhos futuros

- Resumo dos principais resultados e conclusões obtidos.
- Sugestões para pesquisas futuras.

6. Referências

- Lista completa das fontes citadas ao longo do trabalho.

2

Referencial Teórico

A evolução das cadeiras de rodas motorizadas pode ser observada através do avanço técnico das patentes relacionadas. Um exemplo inicial desse progresso pode ser visto na Figura 2.1, que ilustra o esquema de uma das primeiras patentes de cadeira motorizada com *joystick*, com a patente US pat.3807520 de 1971. Essa cadeira apresentava um motor elétrico transmitindo o movimento por corrente (FILHO et al., 2010).

Figura 2.1: Um dos primeiros protótipos de cadeira de rodas patenteados com *joystick* utilizando transmissão por corrente



Fonte: FILHO et al. (2010)

Uma evolução subsequente pode ser observada na Figura 2.2, correspondente à patente US pat.D287836 de 1987. Nessa versão, foi empregada uma cadeira com chassis monobloco, sistema motorreductor diretamente nas rodas de tração e com bancos e *joystick* ergonômicos (FILHO et al., 2010).

Figura 2.2: Sistema motorreductor direto nas rodas de tração



Fonte: FILHO et al. (2010)

Os modelos atuais mais comuns encontrados no mercado internacional são baseados na patente US pat.D397645 de 1996. Esses modelos apresentam melhorias estéticas e ergonômicas,

além de um aumento no número de rodas de estabilidade, ilustrado na Figura 2.3 (FILHO et al., 2010).

Figura 2.3: Sistema motorreductor direto nas rodas de tração com melhorias estéticas, ergonômicas e aumento do número de rodas de estabilidade

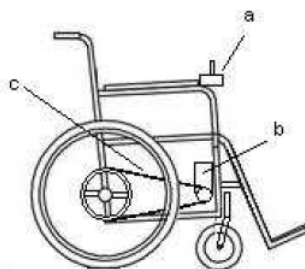


Fonte: FILHO et al. (2010)

2.1 Transmissão

Desde meados de 1981, têm sido realizados estudos em âmbito global para aprimorar a funcionalidade e mobilidade das cadeiras convencionais por meio de adaptações. Ao longo desse período, foram desenvolvidas diversas técnicas para a motorização dessas cadeiras. Uma das abordagens utilizadas consiste no emprego de correias ou correntes (c) para transmitir o movimento do motor (b) a uma polia conectada ao eixo de cada roda traseira, conforme ilustrado na Figura 2.4 (FILHO et al., 2010).

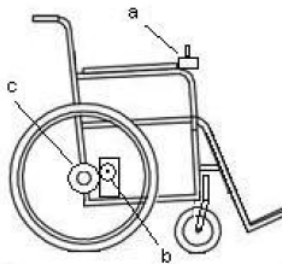
Figura 2.4: Cadeira transmissão flexível, (a) - joystick, (b) - motor, (c) - corrente ou correia para transmitir o movimento do motor para as rodas traseiras



Fonte: FILHO et al. (2010)

Uma forma adicional de motorização é a utilização de engrenagens, conforme ilustrado na Figura 2.5. Nesse método, engrenagens são aplicadas no eixo da roda (c), sendo acionadas por engrenagens no eixo do motor (b). Essa ideia específica pode ser encontrada no documento da patente US Pat 5394066, que foi solicitada em 1990 (FILHO et al., 2010).

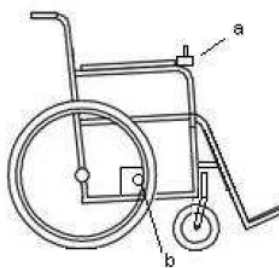
Figura 2.5: Transmissão por engrenagem no eixo, (a) - *joystick*, (b) - Motor com engrenagem, (c) - Engrenagem acoplada diretamente ao eixo da roda para transmitir o movimento do motor



Fonte: FILHO et al. (2010)

Além das técnicas mencionadas anteriormente, a transmissão de movimento por atrito também tem sido observada em aplicações de motorização de cadeiras manuais, como exemplificado na Figura 2.6. Nesse método, o movimento é transmitido através do atrito entre as superfícies permitindo a motorização da cadeira (FILHO et al., 2010).

Figura 2.6: Transmissão por atrito, (a) - *joystick*, (b) - Motor realizando atrito com as rodas traseiras



Fonte: FILHO et al. (2010)

Outra abordagem que tem sido observada é o uso de motores montados diretamente nas rodas, como ilustrado na Figura 2.7. Essa técnica foi documentada na patente US Pat. 5350032. Nesse sistema, a adaptação seria relativamente simples, envolvendo a substituição dos mancais da cadeira por uma base para fixar o eixo do motor. Quando o motor é acionado o eixo não gira, mas sim o motor solidário à roda que é responsável pelo movimento (FILHO et al., 2010).

Essa solução apresenta benefícios, pois simplifica o projeto de motorização e elimina a necessidade de transmissões adicionais. Além disso, essa configuração direta proporciona um controle mais preciso do movimento, uma vez que cada roda pode ser controlada individualmente (FILHO et al., 2010).

Figura 2.7: Motor acoplado no eixo na roda, (a) - *joystick*, (b) - Motor acoplado no eixo roda



Fonte: FILHO et al. (2010)

2.2 Interfaces de controle

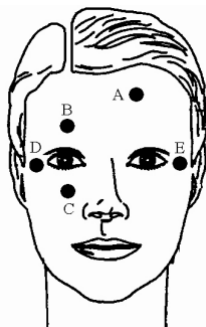
O *joystick* é uma das interfaces mais comumente utilizadas no controle de cadeiras de rodas motorizadas. No entanto, essa forma de controle pode excluir uma parcela significativa de PCDs que não possuem movimentos nos membros superiores, que têm incapacidades cerebrais ou não possuem destreza nas mãos.

Com o intuito de auxiliar esses usuários no controle das cadeiras de rodas, diversos estudos vêm sendo conduzidos para desenvolver interfaces alternativas que possam ser operadas por outras partes do corpo, como cotovelo, boca, língua ou queixo (IVO, 2016).

Um exemplo interessante é um projeto que consistiu no desenvolvimento de um *joystick* acionado por um tubo movido pela boca. Além dos movimentos *xy* convencionais, esse *joystick* permitia comandos adicionais através de sopro ou sucção (FILHO et al., 2010).

No âmbito dos sistemas de acionamento, existem outros projetos importantes que se baseiam no uso de sensores eletrônicos para captar os comandos do usuário. Por exemplo, um projeto desenvolvido por JAFFE (1982) utiliza sensores conectados à cabeça do usuário e a um microcomputador de bordo para controlar a cadeira de rodas. Dessa forma, a PCD poderia direcionar o movimento da cadeira com a cabeça. Para mover a cadeira motorizada para frente, o usuário precisava posicionar a cabeça para frente, e assim sucessivamente para trás, esquerda e direita. A intensidade do movimento controlava a velocidade dos motores.

Outro exemplo é o projeto SIAMO, que é controlado por sensores baseados no processo de eletro-oculografia. Nesse projeto, são utilizados cinco eletrodos ao redor dos olhos, conforme ilustrado na Figura 2.8. No entanto, um problema dessa solução é a disposição dos eletrodos, que pode causar desconforto ao usuário (BAREA et al., 2002).

Figura 2.8: Disposição dos eletrodos no projeto SIAMO

Fonte: BAREA et al. (2002)

Outro projeto interessante é o protótipo VAHM (Figura 2.9), desenvolvido levando em consideração a grande diversidade de tipos de usuários possíveis. Esse protótipo pode ser utilizado em três modos diferentes. O primeiro modo é o modo autônomo, no qual o sistema executa uma trajetória em um ambiente mapeado. O segundo modo é o modo manual assistido, no qual o sistema responde aos comandos do usuário, podendo seguir paredes e evitar obstáculos. O último modo é o modo manual que permite a utilização do dispositivo como uma cadeira de rodas convencional.

Esse protótipo utiliza 14 transdutores ultrassônicos, duas redes de transmissores e receptores infravermelhos juntamente com um processador dedicado para o tratamento dos dados recebidos por cada sensor (BOURHIS et al., 2001).

Figura 2.9: Protótipo VAHM

Fonte: BOURHIS et al. (2001)

2.3 Motor de Corrente contínua

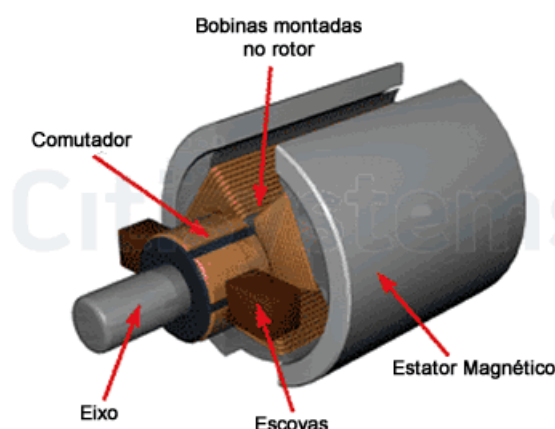
Um Motor corrente contínua (CC) é alimentado por uma fonte de corrente contínua, como uma bateria ou outra fonte de alimentação CC. A comutação desse motor, ou seja, a troca de energia entre o rotor e o estator, pode ser realizada usando escovas (motor escovado) ou sem

escovas (*brushless*). A velocidade do motor pode ser variada simplesmente alterando o nível de tensão aplicado a ele. Por esse motivo, o motor CC é adequado para equipamentos alimentados por níveis de tensão de 24 *volt* (V) ou 12 V, como é o caso de automóveis, bem como em aplicações industriais que exigem um controle preciso de velocidade (SILVEIRA, 2016).

2.3.1 Funcionamento de um Motor CC

Nesse projeto, foi utilizado um motor escovado de corrente contínua. Um motor CC escovado é composto por diversas partes, conforme ilustrado na Figura 2.10. O motor possui um eixo acoplado ao rotor, que é a parte girante do motor. O estator é composto por ímãs, enquanto o motor é representado por uma bobina que é alimentada pelo comutador. O comutador desempenha o papel de ser o elo entre a fonte de alimentação e o rotor, sendo composto por escovas condutoras que fazem contato com o eixo girante do motor. A função do comutador é transferir a energia da fonte de alimentação para o rotor (SILVEIRA, 2016).

Figura 2.10: Partes de um Motor CC



Fonte: SILVEIRA (2016)

2.4 Modulação por largura de pulso

A modulação por largura de pulso (PWM) é uma técnica utilizada para permitir o controle da energia fornecida para equipamentos elétricos, como servomotores, motores CC e dispositivos de iluminação.

Com o uso do PWM é possível controlar a tensão e a corrente fornecidas a uma carga ao ligar e desligar (chavear) o fornecimento de energia entre a fonte e a carga em uma taxa muito rápida. Quanto mais tempo a alimentação permanecer ligada em comparação com o tempo desligado, maior será a quantidade total de potência fornecida à carga. Conseqüentemente, se houver a intenção de ajustar, por exemplo, a velocidade de um motor para 75% de seu

desempenho máximo, essa meta pode ser atingida por intermédio do aproveitamento do conceito de PWM. (REIS, 2017a).

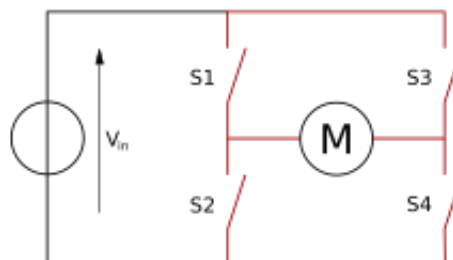
2.5 Ponte H

Uma Ponte H é um circuito especial que permite realizar a inversão da direção (polaridade) da corrente que flui através de uma carga, como um motor CC. É amplamente utilizada em aplicações de controle de motores para possibilitar a mudança de direção de rotação do motor (REIS, 2017b).

Para o correto funcionamento da Ponte H, é essencial que haja um trajeto que conduza a corrente ao motor em uma direção específica e um outro trajeto que permita o fluxo da corrente no sentido oposto. Adicionalmente, o circuito deve ser capaz de estabelecer ou interromper o fornecimento da corrente que alimenta o motor REIS (2017b).

A configuração básica da Ponte H consiste em quatro interruptores eletrônicos (transistores ou relés), que podem ser controlados independentemente. Estes interruptores são acionados de forma alternada para permitir que a corrente flua em direções opostas através da carga. A Figura 2.11 ilustra a disposição dessas quatro chaves (S1 a S4) (REIS, 2017b).

Figura 2.11: Diagrama de um circuito ponte H



Fonte: REIS (2017b)

Para que o motor funcione, é necessário saber em qual estado a chave está, podendo elas serem combinadas.

- S1 e S4 fechados, S2 e S3 abertas: A corrente passa pelo motor em um sentido, fazendo-o girar em uma direção.
- S1 e S4 abertas, S2 e S3 fechadas: A corrente passa pelo motor no sentido inverso, fazendo-o girar na direção contrária.
- S1 e S3 abertas: O motor não gira, pois não há fluxo de corrente por ele.

No geral, as pontes H são construídas com mosfets ou com transistores bipolares e é muito comum encontrá-las na forma de um circuito integrado, o que torna muito mais fácil de usá-las (REIS, 2017b).

2.6 Sensores

Antes de se falar de sensores, é preciso introduzir a definição de transdutores. Segundo RAMÍREZ; JIMÉNEZ; CARREÑO, um transdutor é um dispositivo capaz de converter uma variável física em outra de outro tipo. Ou, de maneira mais geral, um transdutor transforma uma variável física em um sinal elétrico.

A definição de sensor está intimamente relacionada com o conceito de transdutor, pois um sensor sempre fará uso de um transdutor. A principal diferença entre os dois é que o sensor, além de converter uma variável física que foi medida, irá retornar um dado que será útil para a medida de algo. Portanto, assim como RAMÍREZ; JIMÉNEZ; CARREÑO definem, um sensor é um dispositivo que recebe uma entrada e devolve uma saída manipulada através da variável física medida.

2.7 Sistemas embarcados

De acordo com CHASE (2007), um sistema é embarcado quando ele é dedicado a uma única tarefa e interage com o ambiente a sua volta através de sensores e atuadores. A denominação "embarcado" vem devido que estes sistemas são projetados normalmente para serem independentes de fontes energéticas fixas como uma tomada ou um gerador. Tendo como principal característica sua capacidade computacional e a sua independência de operação. Um exemplo de sistema embarcado são os *Smartphones*.

Com o avanço das tecnologias, como a internet das coisas (IdC), houve a criação de placas com capacidades de criar novos sistemas embarcados para diversas aplicações. É o caso do Raspberry pi e dos Arduinos.

2.8 Deep Learning - Aprendizagem Profunda

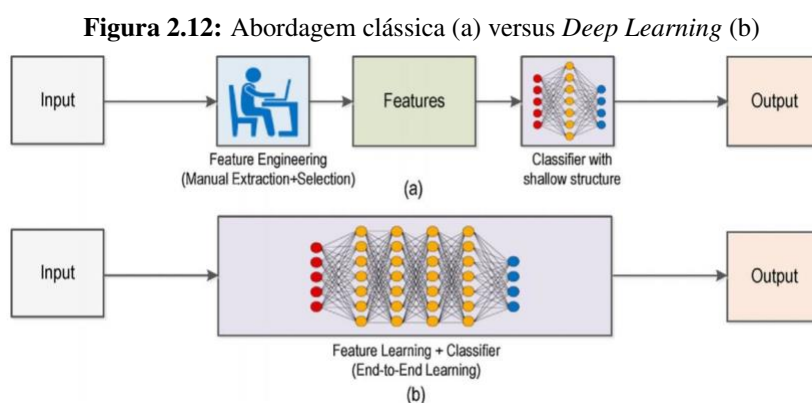
A dificuldade encontrada por sistemas que dependem de conhecimento em codificação rígida sugere que sistemas de IA necessitam da habilidade de adquirir o próprio conhecimento extraíndo padrões de dados brutos. Essa capacidade é conhecida como modelo de aprendizado de máquina. A introdução de ML permitiu computadores atacarem problemas envolvendo conhecimento do mundo real e fazerem decisões que parecem ser subjetivas (GOODFELLOW; BENGIO; COURVILLE, 2016).

Deep Learning permite modelos computacionais, compostos de múltiplas camadas de processamento, a aprenderem representações de dados com vários níveis de abstração. Esses métodos mudaram drasticamente o estado da arte em reconhecimento de fala, e visual de objetos, detecção de objetos e em diversas áreas como em descoberta de remédios e na área de genomas. *Deep Learning* descobre estruturas complexas em grande quantidade de dados, usando algoritmos de retropropagação, para indicar como uma máquina deve mudar seus parâmetros internos, os quais são usados para computar a representação em cada camada, pela representação da camada

anterior. Redes Convolucionais Profundas (*Deep*) trouxeram avanços em processamentos de imagens, vídeos, fala e áudios, enquanto Redes Recorrentes iluminaram dados sequenciais como textos e fala (LECUN; BENGIO; HINTON, 2015).

2.8.1 Abordagem clássica versus *Deep Learning*

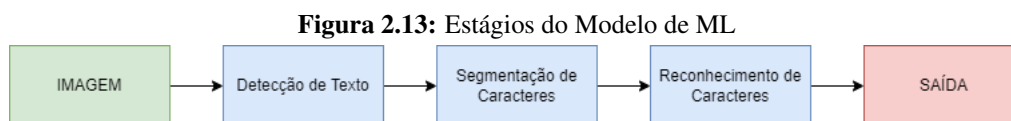
Na Figura 2.12 é demonstrado a distinção de abordagens produzidas pela abordagem clássica de machine learning e a abordagem de *deep learning*.



Fonte: REAL; DORADO; DURÁN (2020)

2.8.1.1 Abordagem clássica - ML

A abordagem clássica de modelo de aprendizado de máquina requer um amplo conhecimento e planejamento tem como base a interferência humana para extração e criação de características dos dados, como, por exemplo, um algoritmo que extrai textos de uma foto (OCR), desde outdoors à capas de revistas, o que ocorre em diversas etapas (Figura 2.13).



Fonte: Elaborado pelos autores

1. A entrada - Foto
2. Um algoritmo para detectar o texto na foto.
3. Recorte do texto.
4. Um algoritmo, que é alimentado com o texto recortado, que reconheça as divisões entres os caracteres e marque eles.

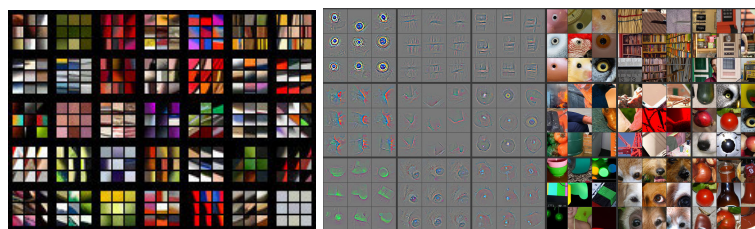
5. Separação dos caracteres.
6. Um algoritmo para classificação dos caracteres encontrados.
7. A saída - O texto resultante.

2.8.1.2 Abordagem *end-to-end* - *Deep Learning*

Em *Deep Learning*, a abordagem, também chamada de *end-to-end* (Figura 2.12), gera um resultado com extração e classificação de ponta a ponta, que permite a extração de características pela profundidade do algoritmo, ou seja, pelo exemplo anterior de OCR, pode-se ensinar o algoritmo a encontrar o texto na foto e ter a forma digitalizada como saída.

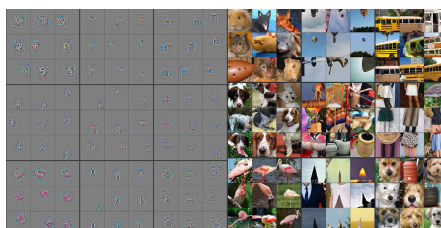
Como apresentado na Figura 2.14, com o aumento da profundidade do modelo, as características mapeadas tornam-se mais complexas, percebe-se que na camada mais rasa (a) simples agrupamentos de cores e linhas, (b) nas camadas médias padrões de cores e características e (c) nas camadas mais profundas, onde já existe o agrupamento de fotos similares.

Figura 2.14: Características extraídas com a profundidade das camadas do modelo



(a) Característica de baixo nível

(b) Característica de médio nível



(c) Característica de alto nível

Fonte: ZEILER; FERGUS (2013).

Dentro da abordagem *end-to-end* existem diversas arquiteturas. Como parte delas, a Rede Neural Residual (ResNet), que apresentou atalhos em arquiteturas, servindo de solução para o desaparecimento de gradientes, algo comum em redes cada vez mais profundas. Como também MobileNet, com suas convoluções separáveis em profundidade, que foi a arquitetura que diminuiu o gasto computacional, com uma leve penalidade em precisão, auxiliando a entrada de diversas soluções de modelo profundos em sistemas móveis.

2.8.2 Visão computacional

Visão computacional é a área de IA, que tem como foco o entendimento visual do mundo real por uma máquina. E tem como funcionamento a entrada de uma imagem e a saída de informação, diferente de processamento de imagem, que tem como foco a transformação de imagem com a entrada e a saída de imagem. Normalmente dividida em 3 grandes áreas: classificação de imagem, detecção de objetos e segmentação de imagem, como representados na Figura 2.15. Sendo importante apenas a área de segmentação.



Fonte: Figura de cachorros e gatos retirados de domínios públicos e editada pelos autores

2.8.2.1 Classificação de Imagem

Classifica a qual classe a imagem pertence, uma possível classificação poderia retornar que a foto em questão é a de um gato, ou de um cachorro. A Figura 2.15 possui um exemplo classificando um cachorro. Um ponto negativo da classificação da imagem é que apenas classifica como um todo, o que faz as classificações retornarem apenas um resultado, onde em casos de objetos, não se possui informação adicional como tamanho e posição.

2.8.2.2 Detecção de objetos

Diferente da classificação, a detecção busca cada objeto das classes escolhidas, como também suas posições na imagem. A Figura 2.15 possui um exemplo de foto detectando gatos e cachorros.

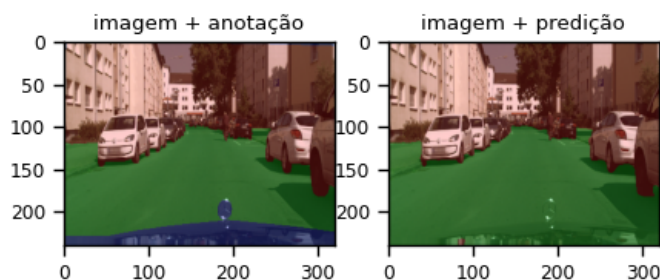
2.8.2.3 Segmentação de imagem

A segmentação de imagem é a separação de porções da imagem, em classes que podem ser objetos ou áreas. A criação de um conjunto de dados para segmentação funciona através de uma imagem, chamada de máscara, a máscara possui valores iguais de *pixels* para classes de mesmo tipo.

Para exemplificar uma máscara, a Figura 2.16 é um exemplo de uma segmentação produzida neste trabalho. Foi feita uma segmentação de caminhos, a primeira imagem representa

a máscara do conjunto de dados, onde existem 3 classes diferentes, verde para caminhos que podem ser percorridos, vermelho para áreas que não podem ser percorridos e azul para áreas ignoradas no treino.

Figura 2.16: Segmentação de caminho - DeepLabV3 + MobileNetV2 no dataset Cityscapes



Fonte: Imagem retirada e editada de Cityscapes(CORDTS et al., 2016)

Em ML, a área de segmentação é formada por segmentação semântica, segmentação de instâncias e segmentação panóptica.

Semântica é o estudo do significado, assim, tem-se que a segmentação semântica divide imagens com base em significados, podendo ser de acordo com tipos de objetos, ou com um objetivo específico, como segmentar caminho de tudo que seja considerado um obstáculo ou, mesmo que não seja, pode ser indesejável para o percurso. Ela não leva como importante a quantidade de objetos, apenas o tipo.

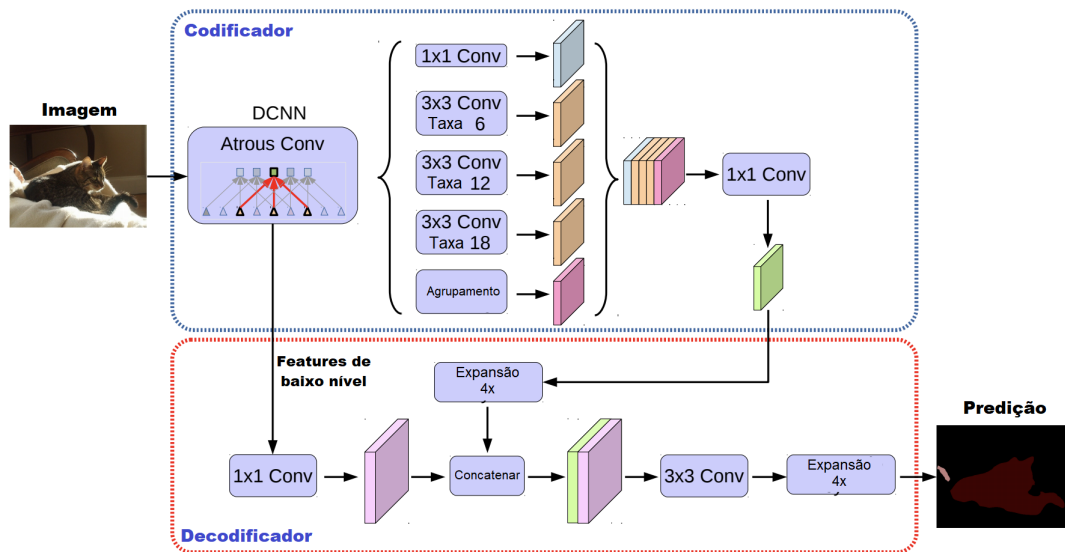
Diferente da segmentação semântica, que não tem consideração com o número de objetos, a segmentação de instâncias foca na segmentação individual de cada objeto, sem considerar a que classes os objetos fazem parte.

Como a segmentação semântica apenas foca no tipo do objeto e a segmentação de instâncias apenas foca em separar cada objeto individualmente, projetos que precisavam segmentar pelas classes e saber especificamente o local de cada objeto precisavam de um novo caminho. A segmentação panóptica funciona segmentando cada classe para o mesmo conjunto de cores, mas cada uma sendo exclusiva. Então, para um exemplo de alto nível, se existir uma classe A, cada objeto dessa classe será segmentado com algum tom de vermelho, e, para uma classe B, algum tom de verde.

2.8.3 DeepLabV3+

A estrutura da arquitetura DeepLabV3+ (Figura 2.17) pode ser explicada em duas partes principais, o *Backbone*, que é a estrutura/arquitetura por trás da codificação, e a decodificação, que é a parte apresentada pela DeepLabV3+, mesmo que a ideia da arquitetura DeepLabV3+ seja o todo.

Figura 2.17: Estrutura da arquitetura DeepLabV3+

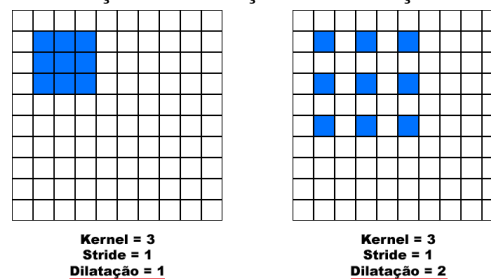


Fonte: CHEN et al. (2018)

O *Backbone*, a arquitetura principal por trás da codificação, é a direta utilização de uma arquitetura que tem como entrada uma imagem, como possível arquitetura existe a possibilidade de utilizar arquiteturas conhecidas para classificação de imagens. Ou formar seu próprio *Backbone* pela montagem de uma estrutura de camadas, que possam receber essa entrada de imagem e codificar as informações.

Nessa arquitetura, como apresentado na Figura 2.17, a saída da rede *backbone* é utilizada como entrada para a estrutura chamada de *atrous spatial pyramid pooling* na codificação, que como toda rede tem objetivo de aprender características, mas com algumas delas sendo relações esparsas na imagem, pois, utiliza-se de convoluções com dilatações, apresentado na Figura 2.18. A dilatação em convoluções captura informações em multi-escalas.

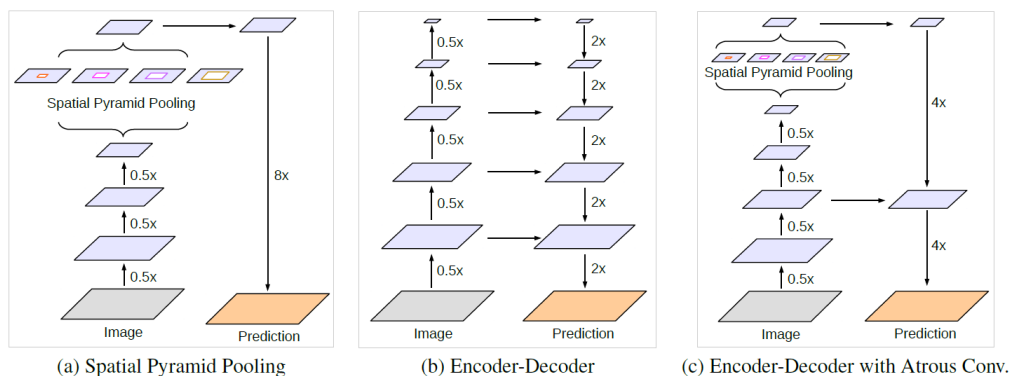
Figura 2.18: Diferença de convolução com dilatação = 1 e dilatação = 2



Fonte: Elaborado pelos autores

Observando a Figura 2.19, DeepLabV3+ propõe a) a utilização da segmentação com *spatial pyramid pooling*, b) com a ideia normalmente utilizada em uma segmentação com codificação e decodificação, c) na estrutura de codificação com informação espaça e decodificação.

Figura 2.19: Diferenças de arquiteturas



Fonte: CHEN et al. (2018)

2.8.4 Ajuste de hiperparâmetros

Hiperparâmetros são valores usados para treinar um modelo, cuja mudança pode alterar o comportamento do treinamento, o ajuste desses parâmetros é feito com o objetivo de obter melhores resultados. Eles podem ser a taxa de treinamento, o valor utilizado para controlar o tamanho do passo do aprendizado, a regularização de pesos, que penaliza o aprendizado para evitar sobre-ajuste aos dados, o alfa de uma *MobileNet*, que altera a quantidade de filtros no modelo, e a profundidade do modelo, que é quantidade de camadas no modelo.

Normalmente, o ajuste de hiperparâmetros é feito em uma escolha aleatória de parâmetros, assim, o espaço de busca representa uma grande área de busca, ocasionando no treinamento de diversos modelos, nos quais os parâmetros que geram o melhor modelo são usados para um próximo treino. Se diversos modelos possuírem resultados aproximados, se escolheria o com o menor custo computacional.

2.8.5 Divisão do conjunto de dados

Uma das partes mais importante do treinamento de modelos de machine learning é a forma que o modelo irá ser treinado, pois existem formas diferentes de como calcular as métricas, treinar e dividir os dados, com base em buscar em gerar um melhor resultado para o problema.

2.8.5.1 Hold out

O modo mais simples de divisão de conjuntos de dados é chamado de *hold out* (Figura 2.20), que consiste em pegar o conjunto e dividir-lo em duas partes, onde uma maior parte é usada para o treinamento e a outra para validar o modelo, normalmente se separa entre 90% e 75% para o treino e o restante para o teste.

A vantagem é o custo em tempo e computação comparado a outros métodos. Utilizado em grandes conjuntos de dados ou com modelos que usam uma grande quantidade de processamento computacional para ser treinado, o que muitas vezes torna inviável múltiplos treinos.

Figura 2.20: Hold out

Conjunto de dados



Fonte: Elaborado pelos autores

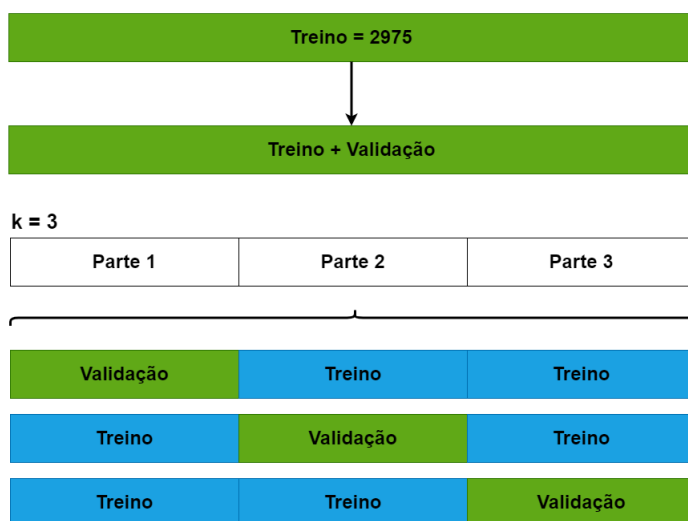
O **hold out** tem desvantagens em relação ao problema de distribuição dos dados, no qual o treino e o teste podem não representar um ao outro, também por existir uma menor validação do modelo, pois, quando se treina em uma divisão treino e teste, se treina em um e validasse em outro, o que não traz o conhecimento de como o modelo vai performar em dados nunca vistos ou outras distribuições de dados.

2.8.5.2 Validação cruzada

Validação cruzada (Figura 2.21) é um método de divisão de conjuntos, o método funciona com a escolha de um k , esse k será a quantidade de partições em que o conjunto de dados será dividido, tendo agora k partições uma partição é usada para o teste, assim, $k-1$ partições será usado para o treino, pode se repetir isso com divisões iguais, ou de tamanhos variados, cada resultado em um modelo para ser treinado.

Figura 2.21: Validação cruzada

Conjunto de dados



Fonte: Elaborado pelos autores

A vantagem de se usar validação cruzada, é não só mitigar o problema da distribuição dos dados de treino e teste, pois, mesmo que se tenha um modelo que performe muito bem, que pode ser baseado em os dados do teste serem mais simples que os dados do treino, pode-se perceber tal comportamento com a performance dos outros modelos. Além de ajudar a mitigar o

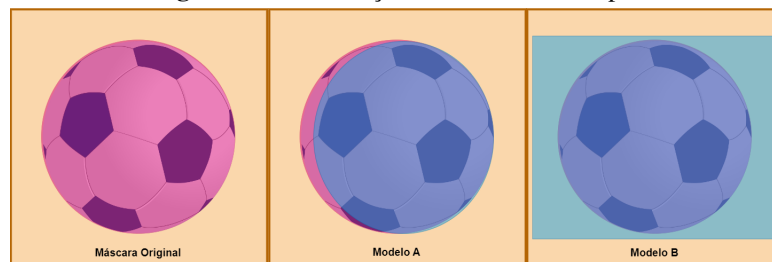
problema de desbalanceamento de classes, bem como pode ser usado para encontrar partições de treino e teste que melhor representem o problema apresentado.

Uma desvantagem é o custo em tempo e computação, já que existe um grande número de modelos sendo treinados.

2.8.5.3 Interseção sobre União (IoU)

Interseção sobre união é uma métrica comumente usada em detecção de objetos e segmentação de imagem, a métrica quantifica a interseção dos dois objetos ou classes comparadas e penaliza em relação a área excedida e faltante. Como na Figura 2.22, o modelo A tem uma segmentação quase total do objeto, mas perde resultado pela área que falta e o modelo B segmenta todo o objeto, mas tem uma grande área excedente, assim se calculado a métrica de interseção sobre a união do modelo A será maior, que do modelo B.

Figura 2.22: Interseção sobre União exemplo



Fonte: pixabay.com

3

Material e Métodos

O presente projeto tem como objetivo a construção de um protótipo que servirá como modelo para uma cadeira de rodas convencional motorizada capaz de prevenção de prevenir incidentes relacionados à presença de obstáculos no trajeto da cadeira de rodas

Considerando que se trata de um projeto mecatrônico caracterizado por sua natureza multidisciplinar, devido à integração de diferentes áreas que nem sempre apresentam intersecções, a metodologia do projeto foi dividida em subprojetos distintos:

- Sistema mecânico
- Sistema eletroeletrônico
- Sistema computacional

3.1 Sistema mecânico

No sistema mecânico da cadeira de rodas motorizada, foram empregadas engrenagens e correntes para transmitir a força dos motores para as rodas. Essa escolha foi baseada no fato de que o sistema estará exposto sujeito a desgaste e impactos. Nesse contexto, as correntes mostram-se mais adequadas em termos de durabilidade.

As correntes são conhecidas por sua resistência e capacidade de suportar condições adversas, como poeira, sujeira e atrito. Além disso, elas oferecem maior capacidade de carga em comparação com as correias. Isso significa que podem transmitir uma quantidade maior de força dos motores para as rodas, o que é essencial para uma cadeira de rodas motorizada.

As transmissões por corrente são comumente utilizadas em situações em que é necessário ter uma distância entre eixos inferior a 3 metros (m), oferece uma relação de transmissão precisa e constante, sendo adequadas para velocidades baixas (NAVARRO, 2018).

No que diz respeito ao rendimento, as transmissões por corrente apresentam uma eficiência que varia entre 92% e 98%. Em relação a custo, são geralmente mais acessíveis em comparação com as transmissões por engrenagens. No contexto deste projeto, elas também são mais econômicas em relação às transmissões por correias, pois seria necessário realizar a troca dos eixos das rodas (NAVARRO, 2018).

É importante ressaltar que as correntes possuem uma vida útil menor em comparação com as engrenagens, devido ao desgaste das articulações. No entanto, sua vida útil é maior quando comparada às correias (NAVARRO, 2018).

Para a motorização da cadeira de rodas, foram utilizados motores CC escovados (Figura 3.1), com uma especificação de 24 V de tensão, 17 ampere (A) de corrente elétrica, potência de 300 *watt* (W) e 2750 de rotações por minuto (RPM). Esses motores foram instalados sobre uma base de madeira fixada na parte inferior da cadeira de rodas.

Figura 3.1: Motor CC MY1016



Fonte: ALIEXPRESS (2022a)

Para transferir a força dos motores para as rodas, foi fixada uma engrenagem adaptada para motorização de bicicletas (Figura 3.2), comumente chamada de coroa, como polia movida. Essa coroa possui um diâmetro de 180 milímetros (mm) e foi inserida no eixo das rodas.

Figura 3.2: Engrenagem para corrente com diâmetro de 180 mm



Fonte: Elaborado pelos autores

Quanto aos motores, foram substituídas as ponteiras originais por engrenagens específicas para correntes (Figura 3.3). Essas engrenagens possuem um diâmetro de 63.5 mm.

Figura 3.3: Engrenagem para corrente com diâmetro de 63.5 mm

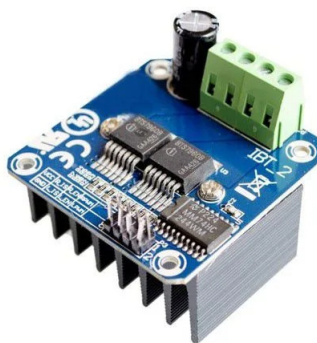


Fonte: Elaborado pelos autores

3.2 Sistema eletro-eletrônico

Devido à configuração dos motores, foram utilizadas duas baterias de nobreak de 12 V, ligadas em série para poder atingir a tensão necessária de 24 V, e de dois módulos ponte H BTS7960 (Figura 3.4), que é um módulo ponte H de alta potência totalmente incorporado para aplicações de acionamento de motores. A interface de Controle (IC) com um microcontrolador é facilitada pela interface incluída do módulo ponte H, que possui entradas de nível lógico, diagnóstico com sentido de corrente, com regulagem da taxa de giro, geração de tempo morto e proteção contra superaquecimento, sobretensão, subtensão, sobrecorrente e curto-circuito.

Figura 3.4: Módulo Driver Ponte H Dupla - BTS7960



Fonte: ELETROGATE (2020)

Especificações dos módulos ponte H BTS7960:

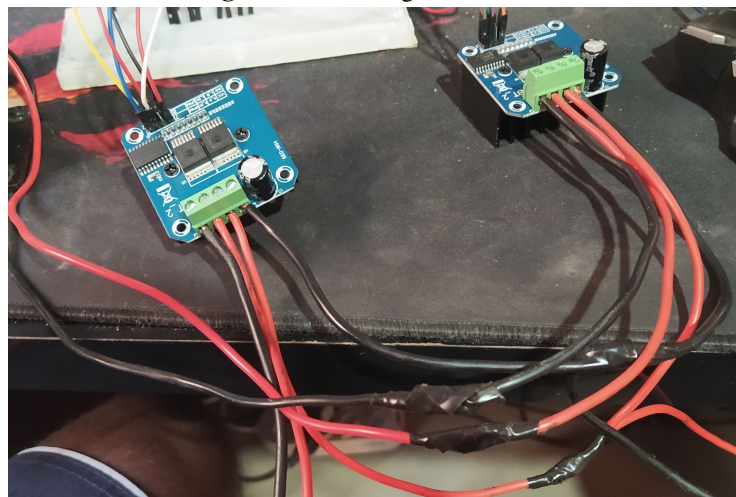
- Tensão de alimentação: 5 - 45 tensão de corrente contínua (Vdc);
- Tensão Lógica: 3,3 - 5,3 V;
- Corrente de Alimentação (módulo): 3 miliampere (mA);

- Máxima potência dissipada: 25 W para 75°C;
- Máxima corrente contínua: 43,0 A;
- Máxima corrente de pico: 60,0 A;
- Proteção: Térmica, sobre tensão, sub tensão, sobre corrente;
- Possui pino para indicar falhas (proteção);
- Possui dissipador de calor;
- dimensão: 50 x 50 x 48 mm;
- Peso: 85 gramas;

No sistema eletroeletrônico, considerando que foram utilizados dois motores, foi necessário aplicar dois módulos ponte H para controlar cada motor individualmente. Para garantir a distribuição adequada da energia das baterias entre os motores, os módulos ponte H foram ligados em paralelo.

Na Figura 3.5 é possível observar as ligações dos módulos ponte H em paralelo. Essa configuração permite que cada módulo ponte H receba a energia necessária para controlar seu respectivo motor, assegurando o funcionamento correto e sincronizado dos motores da cadeira de rodas motorizada.

Figura 3.5: Montagem do Driver



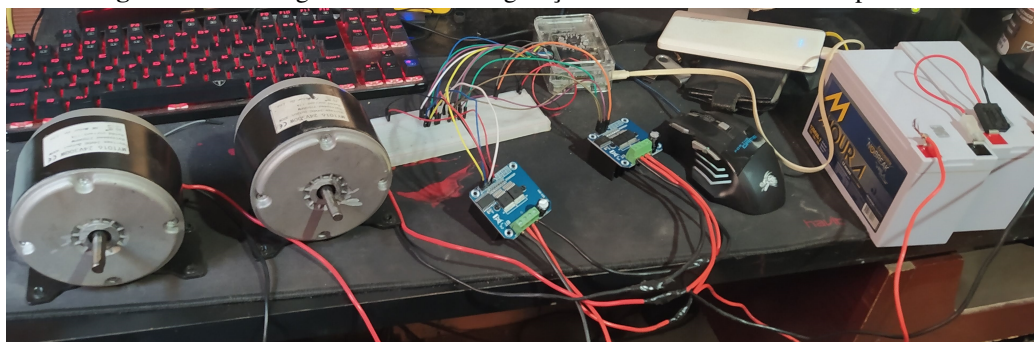
Fonte: Elaborado pelos autores

Para gerenciar todo sistema mecânico e eletro-eletrônico, foi utilizado um Raspberry Pi como método embarcado com o intuito de acionar as controladoras dos motores e para realizar a análise dos dados.

Para analisar a condução da cadeira motorizada para ser adotado, foi realizada a montagem de todo o controle fora da cadeira e o Raspberry Pi foi programado para controlar os

motores a partir de comandos de um teclado sem fio (Figura 3.6). Com isso, foi possível movimentar o motor para frente e para trás e, como as rodas são fixas e não é possível virar, foi necessário fazer com que um lado se movimente mais rápido que o outro. Dessa forma, para que seja possível virar para a direita, a velocidade do motor na roda direita será mantida menor que o lado esquerdo, como também para se virar para a esquerda, se mantém a velocidade do motor esquerdo menor que o lado direito. A demonstração da análise pode ser vista no link <https://youtu.be/VOWFLjAWY5A>.

Figura 3.6: Montagem de toda a configuração fora da cadeira de rodas para teste

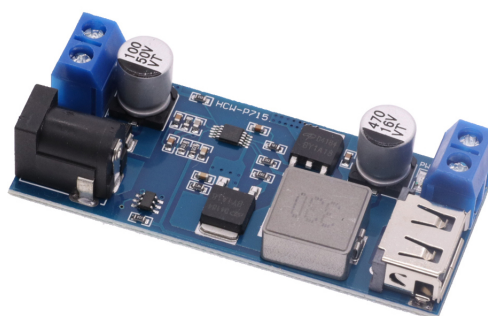


Fonte: Elaborado pelos autores

Para utilizar a corrente das baterias para alimentar o Raspberry Pi, foi inserido um redutor de tensão para diminuir a tensão das baterias a um nível adequado para o funcionamento do Raspberry Pi (Figura 3.7).

Nesse caso, o redutor de tensão está ligado em paralelo com os demais componentes do sistema. Ele recebe uma entrada de 24V e converte essa tensão para uma saída de 5V, que é o necessário para alimentar o Raspberry Pi. Dessa forma, é possível fornecer a energia adequada para o funcionamento do computador embarcado utilizando a mesma corrente proveniente das baterias utilizadas para alimentar os motores da cadeira de rodas motorizada, garantindo o suprimento de energia adequado ao Raspberry Pi sem interferir no funcionamento dos motores ou de outros dispositivos do sistema.

Figura 3.7: Redutor de tensão



Fonte: ALIEXPRESS (2022b)

3.3 Sistema Computacional

Para o sistema computacional, foi utilizado um Raspberry Pi 3B+ (Figura 3.8) equipado com 2GB de RAM e executando o sistema operacional Ubuntu Server 20.04.4 LTS 64-bit. O microcomputador é acessado através do *secure shell* (SSH) com a interface gráfica desativada a fim de obter um desempenho otimizado. É usado o valor 42 para todos os valores de escolha aleatória que possuem o parâmetro SEED com o objetivo de manter a reprodutibilidade.

Figura 3.8: Raspberry Pi 3 Model B+



Fonte: RASPBERRY PI (2014)

Para o funcionamento da detecção de perigo foi escolhido a segmentação de imagem com base na arquitetura DeepLabV3Plus (CHEN et al., 2018), pois é uma arquitetura de segmentação mais leve, e conseguida uma liberdade para a escolha da arquitetura que codifica a informação de entrada, chamado também de backbone; assim, podem ser uma arquitetura com boa performance e acurácia em sistemas embarcados. O backbone utilizado foi o MobileNetV2 (SANDLER et al., 2018) presente na biblioteca Tensor Flow no módulo Keras, com pesos pré treinados no conjunto de dados ImageNet. Modelo é encontrado no link <https://github.com/RWaiti/Keras-DeeplabV3Plus-MobilenetV2/blob/main/models.py>. O código 3.1 é uma modificação do código disponível no site oficial do keras e possui uma modificação no final da rede, esse modelo será chamado de modelo A (Figura 3.9).

Bloco de Código 3.1: Exemplo de utilização do modelo

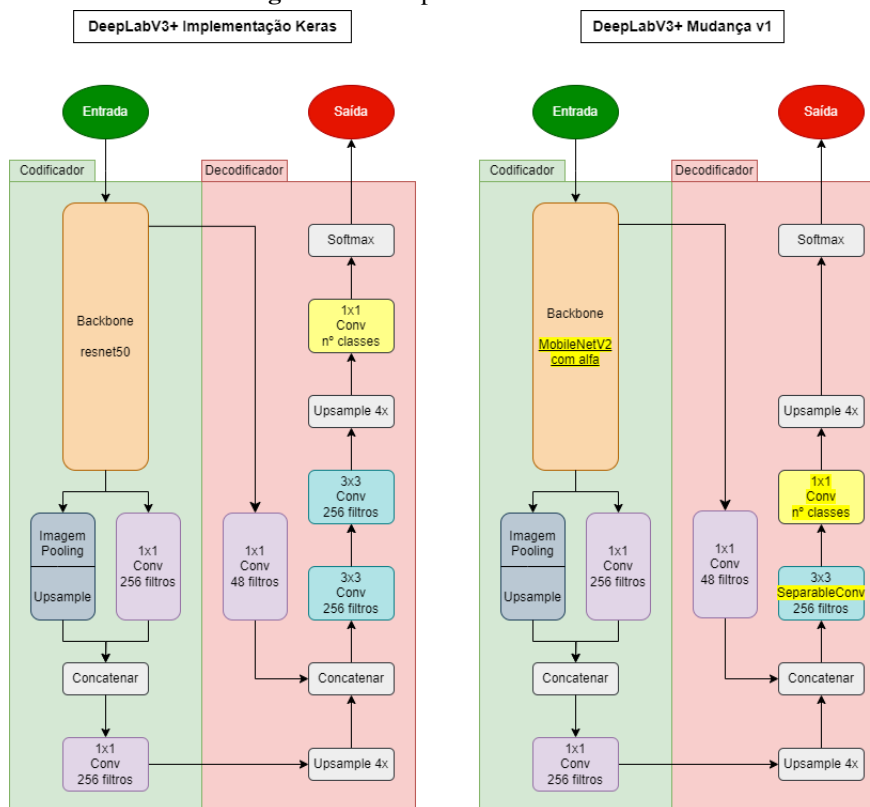
```
1 import models
2
3 img_size = (256, 256)
4 n_classes = 2
5 alpha = 1.0
6 l1 = l2 = 1e-5
7
8 mobileLayers = {"shallowLayer": "block_2_project_BN",
9                 "deepLayer": f"block_12_project_BN"}
```

```

10
11 model = models.deeplabV3(
12     imageSize=img_size, nClasses=n_classes,
13     alpha=alpha, mobileLayers=mobileLayers,
14     kernel_regularizer=
15         tf.keras.regularizers.L1L2(l1=l1, l2=l2))

```

Figura 3.9: DeepLabV3+ - modelo A



Fonte: Elaborado pelos autores

O conjunto de dados escolhido para o trabalho foi o Cityscapes (CORDTS et al., 2016), voltado para pesquisa na área de visão computacional, que possui anotações de segmentação de ruas para veículos autônomos. Na anotação precisa, há 2975 imagens para treino, 500 para validação, que serão usadas de teste, e 1525 para teste, que infelizmente não poderão ser usadas, pois o conjunto de teste não possui anotação e a saída do modelo precisa ser enviada para o site oficial, para que os resultados sejam avaliados. Na anotação 'não precisa', há 20 mil dados de treino, mas eles não serão usados neste trabalho pelo aumento no tempo de treino. Há 19 classes representadas na tabela encontrada no link <https://github.com/mcordts/cityscapesScripts/blob/a7ac7b4062d1a80ed5e22d2ea2179c886801c77d/cityscapesScripts/helpers/labels.py#L62>.

O conjunto também possui dados editados para aumento dos dados, possuindo nos dados de anotação precisa as versões com chuva e com neblina.

Será feita uma mudança de tamanho agressiva na imagem, para se ter um modelo que utilize menos computação com uma entrada menor, as imagens no conjunto de dados possuem um tamanho de 2048 x 1024 e, nos primeiros modelos, será usado o tamanho 256 x 256, mudado em escolhas futuras.

Para tal treino, as máscaras foram remapeadas para categoria que representa caminhos, como pista, meio fio, estacionamento e trilhos de trem, a parte atravessável, igual a 1, e todas as outras categorias iguais a 0, com exceção das partes vazias e não anotadas, que foram colocadas como 2, e serão ignoradas no cálculo de erro do treinamento e no cálculo das métricas. Por ter sido utilizada uma segmentação mais simples, isso resultou na possibilidade de cortar a arquitetura, mudando a última camada a ser usada para o treino da camada 16 para a camada 12, o que resultou na diminuição de 2.229.406 para 826.210 parâmetros, um corte de aproximadamente 63% nos parâmetros, no modelo A, no código disponível no link <https://github.com/RWaiti/Keras-DeepLabV3Plus-MobileNetV2>. Ao qual, pode ser observado a mudança no código 3.2.

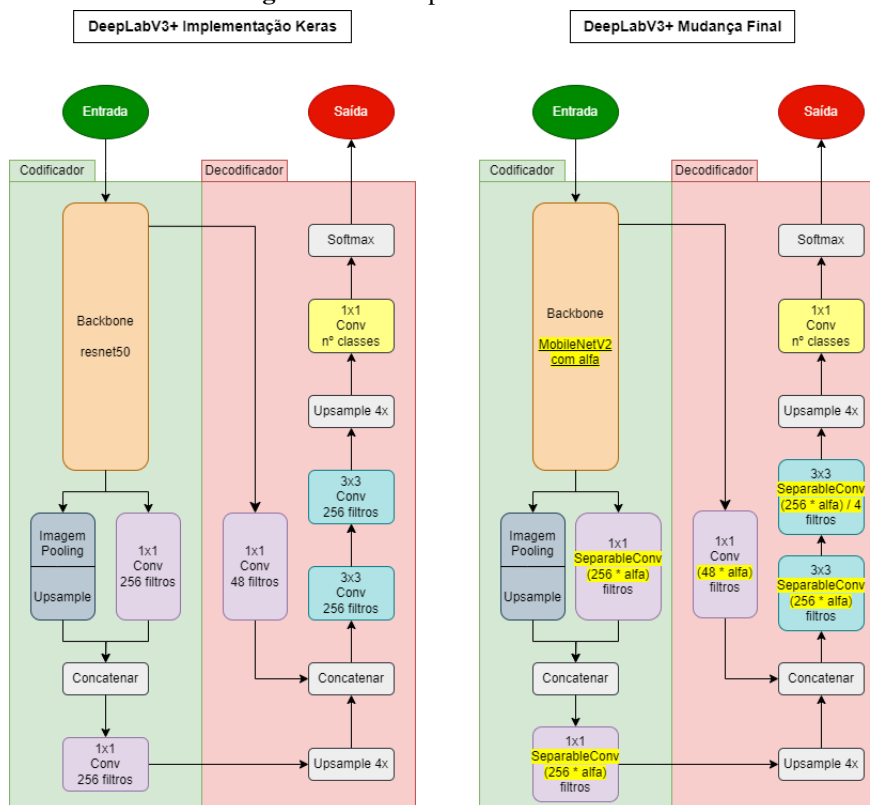
Bloco de Código 3.2: Diminuição de camadas na arquitetura

```
1 # antes
2 mobileLayers = {"shallowLayer": "block_2_project_BN", "deepLayer": ←
   f"block_16_project_BN"}
3
4 # depois
5 mobileLayers = {"shallowLayer": "block_2_project_BN", "deepLayer": ←
   f"block_12_project_BN"}
```

A família de arquiteturas MobileNet são pensadas para hardwares mais simples e, por isso, possuem um parâmetro chamado alfa, que controla a quantidade de filtros nas camadas do modelo - quanto mais filtros, mais parâmetros, e mais poderoso o modelo é, mas, como o objetivo é embarcar o modelo em um hardware mais leve, dois alfas serão usados 1.0, considerado o padrão, e 0.5, que terá metade dos filtros do padrão, o que resulta em outra diminuição, onde o alfa 1.0 possui 826.210 parâmetros e o alfa 0.5 possui 402.258, no modelo A.

Como uma última modificação, que será discutida nos 4, será voltado o final da rede original do Keras, mas com a adição do alfa não só para controlar os filtros da rede MobileNetV2, mas para controlar também a rede de segmentação. Modelo que será referenciado como modelo B (Figura 3.10).

Figura 3.10: DeepLabV3+ - modelo B

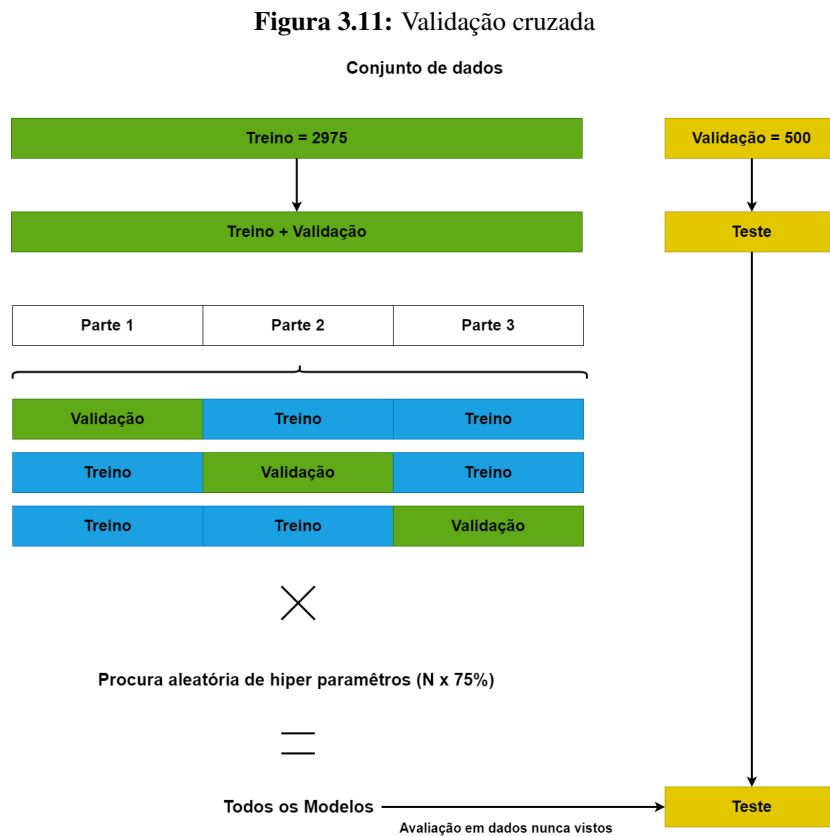


Fonte: Elaborado pelos autores

Pensando na distribuição dos dados no treinamento e estabilidade do aprendizado do modelo, como procurar o melhor modelo possível, será feita a procura de hiper parâmetros e validação cruzada.

A validação cruzada acontecerá com um hold out de treino e teste, no qual, dentro do conjunto de treino, será feita a validação cruzada, com k partições. O conjunto de teste será usado para testar o resultado dos modelos em dados nunca vistos (Figura 3.11).

Assim, existirá n (parâmetros) * k (partições) = x (modelos) a serem treinados. Modelos serão usados para entender o comportamento e resultados nos dados, como também para escolher os parâmetros para o treinamento do último modelo, com todos os dados juntos, de treino e validação, junto com aumento de dados com névoa, inversão da imagem e sendo validado no conjunto de teste.



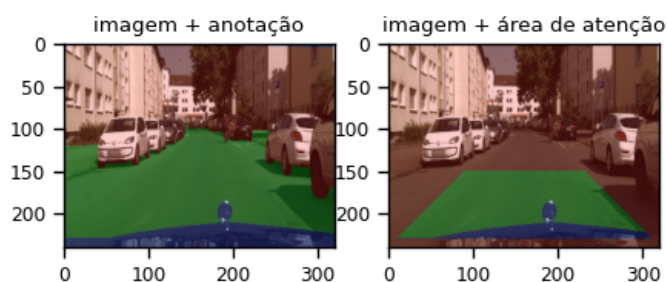
Fonte: Elaborado pelos autores

3.4 Área de atenção

WANG et al. (2018) trazem um modelo de segmentação de trilhos, segmentação que é finalizada com um tratamento para correção de bordas, levando em conta o formato que os trilhos possuem, assim gerando uma melhor segmentação para o problema.

Esse projeto traz uma proposta para finalizar a análise da área percorrável ou não, já que é um projeto de visão computacional com visão monocular (possui apenas um ponto de visão), algo importante para essa análise de obstáculos, pois será criada uma área para focar a atenção do perigo a frente, área essa consistida de um polígono formado pela perspectiva linear da imagem, gerada pela visão monocular, um polígono, um triângulo que acaba no horizonte, e que será finalizado com cortes no topo, objetos mais longe, e um pequeno corte na base, para jogar o foco mais a frente, assim formando uma área de atenção com formato trapezoidal. Como na Figura 3.12

Figura 3.12: Área de atenção. verde caminho, vermelho não caminho e azul ignorado no aprendizado



Fonte: Imagem retirada e editada de Cityscapes(CORDTS et al., 2016)

3.5 Custo do projeto

A tabela 3.1 apresenta uma estimativa de quanto custaria para realizar a construção de uma cadeira motorizada. Foi considerado apenas o valor dos materiais principais que foram utilizados até o momento.

Tabela 3.1: Estimativa de custo da instrumentação de uma cadeira de rodas motorizada

Componentes	Quantidade	Valor unitário	Valor total
Driver BTS7960	2	R\$ 49,28	R\$ 98,56
Redutor de tensão	1	R\$ 9,97	R\$ 9,97
Motor CC Escovado	2	R\$ 240,00	R\$ 480,00
Coroa de 44d com kit fixação da coroa	2	R\$ 28,21	R\$ 56,42
Engrenagem para corrente	2	R\$ 37,50	R\$ 75,00
Cadeira de rodas ortomix	1	R\$ 2.506,65	R\$ 2.506,65
Bateria 12V para nobreak	2	R\$ 66,21	R\$ 132,42
Módulo Joystick	1	R\$ 13,99	R\$ 13,99
Raspberry Pi 3B+	1	R\$ 485,00	R\$ 485,00
Webcam	1	R\$ 30,12	R\$ 30,12
TOTAL			R\$ 3.888,03

Fonte: Elaborado pelos autores

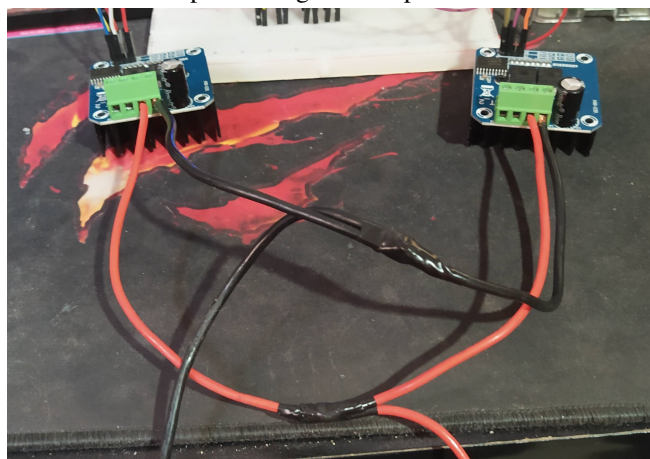
4

Resultados e Discussão

4.1 Sistema eletro mecânico

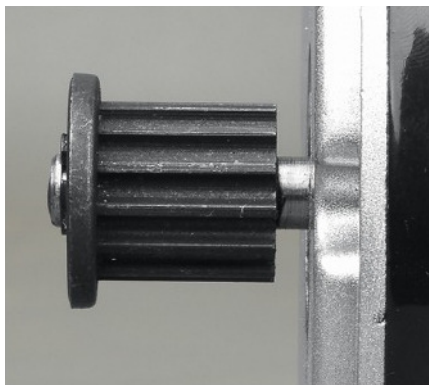
Durante a construção do sistema eletromecânico, houve várias dificuldades. O primeiro problema ocorreu ao lidar com o controle dos motores utilizando os módulos ponte H. Devido à seção dos fios utilizados, que tinham uma espessura de 0.5 mm^2 , dois dos módulos queimaram devido à incapacidade dos fios de suportar a corrente das baterias em série. Como solução, esses fios foram substituídos por outros com uma seção de 2.5 mm^2 que foi capaz de lidar com a conexão em série necessária para alimentar os dois módulos que estavam conectados em paralelo (Figura 4.1).

Figura 4.1: Módulos ponte H ligados em paralelo com fios de 2.5 mm^2



Fonte: Elaborado pelos autores

Em seguida, houve dificuldades relacionadas aos motores escolhidos. As polias que vieram com os motores eram polias dentadas projetadas para serem utilizadas com correias (Figura 4.2). No entanto, para incorporar o uso de correias no projeto, seria necessário substituir os eixos das rodas da cadeira de rodas e adquirir novos eixos específicos para serem adaptados a uma roda dentada para correias (Figura 4.3). Isso resultaria em um aumento no custo do projeto devido às adaptações necessárias.

Figura 4.2: Polia para correia

Fonte: ALIEXPRESS (2022a)

Figura 4.3: Roda dentada para correia

Fonte: ALIEXPRESS (2022a)

Com isso, foi necessário obter uma nova polia, uma que seja adaptada para utilizar com correntes que são as engrenagens para correntes, demonstrada na Figura 3.3. Mas as engrenagens deveriam se encaixar com o eixo do motor, pois o encaixe do eixo tem um achatamento fazendo com que fosse mais difícil de se encontrar a peça. Após obter as engrenagens, foi percebido que o espaçamento entre os dentes é menor que os espaçamentos da coroa (Figura 3.2).

Devido a essa situação, quando os motores eram acionados para girar as rodas, a corrente deslizava entre os dentes da engrenagem impossibilitando que as rodas fossem giradas pela força dos motores. Então, foi solicitado um novo par de engrenagens, porém, a largura do furo na engrenagem para a entrada do eixo do motor era exatamente do mesmo tamanho, impossibilitando a conexão, com isso, foi desbastado o eixo para que a engrenagem pudesse entrar, ocasionando um novo problema. O motor não conseguia girar de forma linear, fazendo com que a corrente ficasse caindo da engrenagem.

Então, de modo que fosse possível demonstrar o software de detecção de perigo, foi construído um protótipo que representa uma cadeira de rodas com a mesma configuração, porém mudando o modo de transmissão.

Foram adaptadas as rodas dianteiras da cadeira de rodas para que fossem inseridos em seus respectivos eixos os motores, para que, com isso, fosse possível mover a base construída em madeira contendo a configuração do projeto, que consiste na câmera, nos dois módulos ponte H, no redutor de tensão, nas baterias e no Raspberry pi.

Para adaptar as rodas, foram utilizados dois mandris de furadeira, de modo que, no lado onde deve ser inserida a broca, foi inserido o eixo do motor e, no outro lado que contém uma rosca de parafuso, foi inserido um parafuso que realizava a prensa da roda entre a cabeça do parafuso com uma ruela e a entrada do mandril. Assim, foi acoplada a roda ao eixo, resultando no movimento do protótipo demonstrado na Figura 4.4 e Figura 4.5.

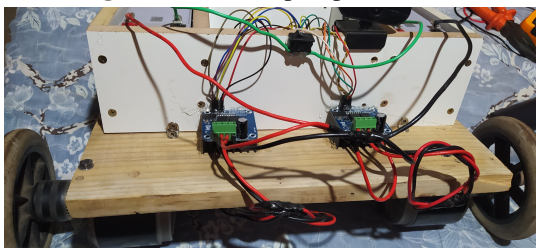
Figura 4.4: Adaptação do motor

Fonte: Elaborado pelos autores

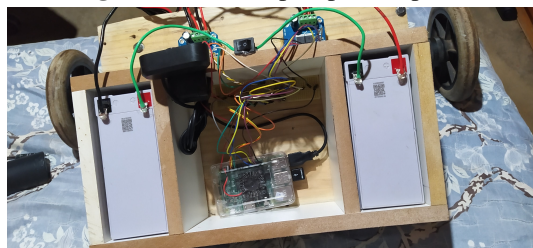
Figura 4.5: Adaptação do motor

Fonte: Elaborado pelos autores

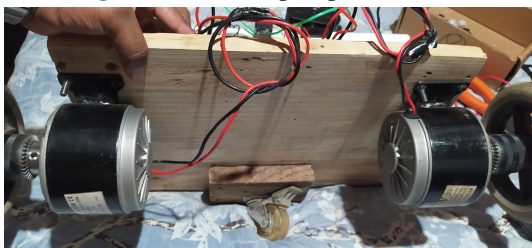
Após a conclusão da construção, foi identificado um novo problema relacionado à direção dos motores. Os motores estavam girando em direções opostas devido ao fato de que os motores possuíam um lado específico de funcionamento, ou seja, o motor do lado esquerdo estava girando para frente, enquanto o motor do lado direito estava girando para trás. Para resolver essa questão, foi necessário alterar a polaridade do motor direito diretamente no módulo ponte H, pois fazer essa modificação via código não surtiu efeito. Ao tentar fazer a alteração no código, resultava em incompatibilidade entre os módulos utilizando os comandos invertidos, ocasionando na falta de movimentação em ambos os motores. Demonstração da construção do protótipo na Figura 4.6, Figura 4.7 e Figura 4.8.

Figura 4.6: Protótipo - parte frontal

Fonte: Elaborado pelos autores

Figura 4.7: Protótipo - parte superior

Fonte: Elaborado pelos autores

Figura 4.8: Protótipo - parte inferior

Fonte: Elaborado pelos autores

Apesar de todas as adaptações realizadas, foi constatado que as baterias não conseguem fornecer corrente suficiente para movimentar os dois motores simultaneamente, considerando o peso da base e o atrito com o solo. Resultando em um acionamento alternado, que, mesmo

quando aplicada aceleração para avançar e ambos os motores deveriam estar em movimento, somente um deles era ativado, enquanto o outro permanecia imóvel. Similarmente, quando um dos motores que estava previamente inativo conseguia entrar em movimento o outro permanecia em repouso.

Essa situação deriva da demanda dos motores por uma corrente de 17A, ao passo que as baterias empregadas fornecem somente 7A. Isto ocorre devido à natureza da ligação em série, que incrementa a tensão, mas mantém inalterada a corrente elétrica. Assim sendo, seria imprescindível utilizar no mínimo 3 baterias interconectadas em paralelo para garantir um fornecimento de corrente de 21A, teoricamente suficiente para acionar os motores com a potência requerida para movimentar o modelo de maneira efetiva. Não obstante, uma análise mais minuciosa se faz necessária para determinar a capacidade essencial para uma locomoção eficaz do modelo, com uma duração significativa do deslocamento.

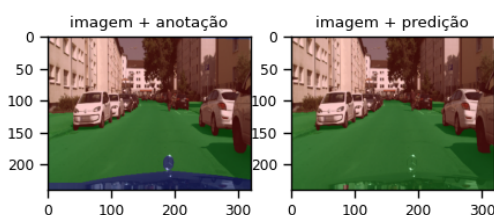
4.2 Sistema Computacional

Para o treinamento do modelo, inicialmente a escolha tinha sido treinar o modelo em algum conjunto de dados maior e retreinar nos dados coletados no campus, o que foi descartado pelo custo de tempo.

O conjunto de dados não possuía muitos dados de anotação precisa, o que causa uma dificuldade em generalizar as 19 classes em um modelo menor, e imagens que perderam informação pela grande mudança de tamanho. Com essa dificuldade em mente, a escolha foi partir para uma abordagem mais simples dos dados e tirar o melhor resultado, com uma melhor performance. Primeiro, foi decidido remapear as 19 classes para as 7 categorias, (*vazio, terreno, construção, objeto, natureza, céu, humano, veículo*).

O primeiro modelo treinado tinha como objetivo testar o aprendizado do modelo e o código, foi usado o maior modelo com a camada profunda 16, alfa 1.0, o remapeamento das 19 classes para as 7 categorias, para combater o desbalanceamento de classe, o treinamento foi alimentado com o pesos nas classes, pesos criados pela porcentagem daquela classe na imagem. Como divisão do conjunto, foi usada a abordagem *hold out* de divisão de dados, usando o split do próprio conjunto de dados de anotação precisa, com alguns dados do treinamento aumentados em neblina e dando zoom em partes aleatórias da imagem (Figura 4.9).

Figura 4.9: Área predita verde caminho, vermelho não caminho e azul ignorado no aprendizado



Fonte: Imagem retirada e editada de Cityscapes(CORDTS et al., 2016)

O modelo A obteve um resultado de interseção sobre união (mIOU) nas imagens do conjunto de teste de 62%, treinado então tinha como performance aproximada de 3.33 *frames per second* (fps) médios, no Raspberry PI, com a parte gráfica desativada e controlado por SSH.

Para buscar uma melhor performance e precisão, foi decidido simplificar o problema mais ainda, focar apenas na segmentação do caminho percorrável e da área não percorrável, formado de obstáculos ou terrenos não propícios, uma segmentação binária que tem como resultado a área que a cadeira pode andar e todo o resto como área perigosa, relevando qual o tipo do obstáculo e se está acima ou abaixo do nível do chão.

O começo foi criar um modelo com busca de hiperparâmetros, com 2 valores para os parâmetros de regularização l1 e l2, 2 valores para coeficiente de aprendizado, 2 para o alfa; assim, existiam $2 \times 2 \times 2 \times 2 = 16$ combinações sem repetições, para diminuir a quantidade de modelos, foi usada a escolha aleatória de hiperparâmetros em apenas 75% dos modelos, o que equivale a 12 dos 16 modelos, com $k = 3$, com isso, tinha-se 12 combinações \times 3 partições, ou seja, 36 modelos para treinar. Infelizmente, após semanas de treino foi percebido um vazamento de memória nos dados, pois os dados de validação cruzada possuíam dados das imagens e mais as mesmas imagens com névoa artificial, então, nas divisões dos partições podia-se ter no treino a mesma imagem no conjunto de treino e validação, uma sem e outra com a mudança de névoa artificial. Mesmo com a perda dos modelos, foi convertido um modelo para o Raspberry Pi para documentar a performance, que foi um resultado médio de 8 fps no modelo tipo A.

Por terem sido perdidas semanas treinando os 36 modelos, houve mudanças nos passos do treino do modelo.

Para diminuir o tempo de treino, foi escolhido separar a validação cruzada da busca de hiper metros, retirando, assim, a multiplicação entre espaço de busca e validação cruzada. Na validação cruzada também foi mudado o k de 3 para 5, pois, foi treinado uma vez o modelo em 5 partições, com o objetivo de determinar melhor a estabilidade do modelo e a distribuição dos dados. Na busca, foi escolhido diminuir o espaço de busca usando uma busca apenas com a regularização l2 com 3 parâmetros e qual função de erro seria usada. Também foi removido o aumento de dados, como inverter, cortar a imagem e também os dados de névoa artificial, apenas para os treinos de validação cruzada e busca de hiperparâmetros.

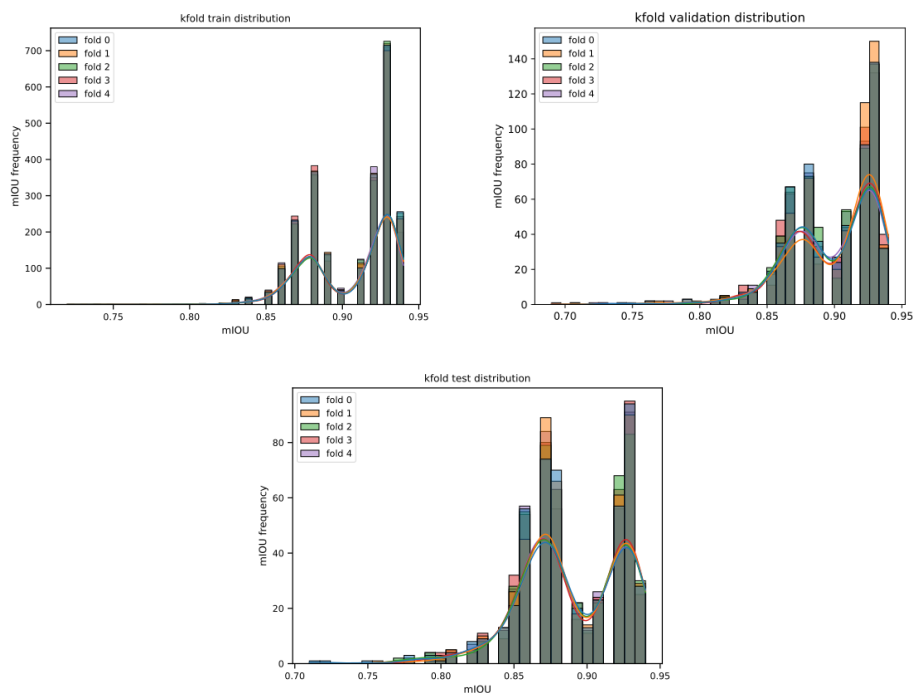
Também foi modificado o tamanho de entrada do modelo, pois, foram usadas imagens de $256 \times 256 \times 3$, mas a câmera usada no Raspberry PI possui como menor resolução a proporção de $320 \times 240 \times 3$, foi feita alteração para não precisar ser feito alterações de tamanho entre a captura da câmera e a passagem dos dados para o modelo, buscando diminuir computação e distorções nas imagens.

Outra mudança foi usar o modelo B, explicado na parte de desenvolvimento, pois foi pensado que a rede ter 2 filtros antes de voltar para o tamanho original perdia capacidade de aprender mais sobre o problema, a rede possui 234.042 parâmetros, que é menor que o modelo A, mas mesmo assim uma performance em fps menor, com média 6 fps.

Para a validação cruzada, foi-se fixando valores e começando os treinos, para ver se o

modelo aparentava nas primeiras épocas sobre ajuste ou subajuste. Quando o modelo demonstrou um comportamento que não parecia nenhum dos dois, foi feita a validação cruzada. Os valores escolhido de começo foram os parâmetros para a regularização l2 de 1×10^{-4} , taxa de aprendizado de 1×10^{-3} , e 750 épocas de treinamento, como já citado nos métodos diminuimos a profundidade da rede usando a camada 12 do backbone MobileNetV2, mas com o objetivo de reduzir mais o modelo foi utilizado o alfa como 0.5, o que faz com que existam metade dos filtros nas camadas do modelo. Na Figura 4.10 é apresentado os resultados da validação cruzada e a distribuição dos resultados.

Figura 4.10: Distribuição dos resultados nas 5 partições



Fonte: Elaborado pelos autores

Como pode ser observado na Tabela 4.1, o resultado do modelo se mostrou estável entre as partições, nas quais a diferença entre o pior e o melhor modelo de validação é $\approx 0.2\%$ e com uma piora de apenas $\approx 0.1\%$ para os dados nunca visto no conjunto de teste, como também uma melhora considerável do remapeamento de 7 categorias para binário, mesmo treinando com 4/5 do conjunto de dados. Chegando à conclusão de que o modelo atual pode ser utilizado para treinar um modelo que tenha boa predição em dados futuros e que a distribuição dos dados esteja estável.

Tabela 4.1: Resultado dos modelos de validação cruzada

IoU médio - Validação cruzada			
	Conjunto de treino	Conjunto de validação	Conjunto de teste
fold	<i>média ± sd</i>	<i>média ± sd</i>	<i>média ± sd</i>
0	0.9065 ± 0.0285	0.8982 ± 0.0331	0.8892 ± 0.0369
1	0.9057 ± 0.0289	0.9006 ± 0.0338	0.8901 ± 0.0345
2	0.9065 ± 0.0289	0.8990 ± 0.0316	0.8894 ± 0.0351
3	0.9062 ± 0.0285	0.8997 ± 0.0325	0.8898 ± 0.0345
4	0.9055 ± 0.0286	0.8980 ± 0.0335	0.8890 ± 0.0353
Média	0.9061	0.8991	0.8895

Fonte: Elaborado pelos autores

O próximo passo foi a busca de hiperparâmetros. A escolha foi rodar em um espaço de busca menor que só buscava a mudança na regularização l2 nos valores 1×10^{-2} , 1×10^{-4} e 1×10^{-6} , e a função de erro entre as duas *loss_dice_coef* e *loss_iou_coef*, assim treinando 6 possíveis modelos. A busca foi realizada usando todo o conjunto de treino e testando no conjunto de teste, pois já foi demonstrada a capacidade do modelo de previsão em dados futuros. Código das funções de erro disponíveis no github no link <https://github.com/RWaiti/Keras-DeepLabV3Plus-MobilenetV2/blob/15c24fb221893577a0a6320af10f05edb0utils/lossesAccuracyfuncs.py#L38>.

Os resultados muito próximos, demonstrado na Tabela 4.2 mostram que o espaço de busca não afeta o resultado, já que um modelo estocástico sempre tende a ter pequenas variações. Mesmo assim, foi escolhido o melhor resultado no teste para o treinamento do último modelo.

Tabela 4.2: Resultado dos modelos de busca de hiperparâmetros

IoU médio - busca de hiperparâmetros			
		Conjunto de treino + validação	Conjunto de teste
função de erro	l2	<i>média ± sd</i>	<i>média ± sd</i>
<i>loss_dice_coef</i>	1×10^{-2}	0.9062 ± 0.0284	0.8902 ± 0.0347
<i>loss_dice_coef</i>	1×10^{-4}	0.9063 ± 0.0286	0.8904 ± 0.0349
<i>loss_dice_coef</i>	1×10^{-6}	0.9063 ± 0.0287	0.8895 ± 0.0356
<i>loss_iou_coef</i>	1×10^{-2}	0.9066 ± 0.0287	0.8907 ± 0.0340
<i>loss_iou_coef</i>	1×10^{-4}	0.9057 ± 0.0286	0.8898 ± 0.0348
<i>loss_iou_coef</i>	1×10^{-6}	0.9072 ± 0.0284	0.8904 ± 0.0346
Média			0.8893

Fonte: Elaborado pelos autores

O último modelo foi treinado utilizando os mesmo parâmetros do modelo anterior, mas com uma leve variação nos dados de treino, pelo fato de que foram feitas aleatoriamente mudanças na imagem de entrada, com chances de 20% da imagem ser invertida horizontalmente, 20% verticalmente e 20% de ter um zoom em partes da imagem, transformações usadas para mitigar um problema explicado após as métricas, bem como foram usados os dados com névoa artificial.

Foram então utilizadas 8925 imagens, dentre as quais 2975 imagens não tinham mudanças, 2975 imagens tinham pouca névoa e 2975 tinham muita névoa. Resultados demonstrados na Tabela 4.3

Tabela 4.3: Resultado do modelo final

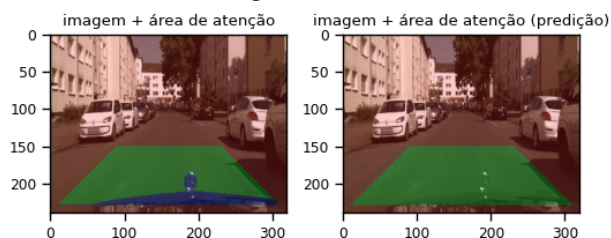
IoU médio - Modelo final						Conjunto de teste
função de erro	imagem	lr	alpha	l2	batch	<i>média ± sd</i>
<i>loss_dice_coef</i>	(240, 320)	1×10^{-3}	0.5	1×10^{-2}	2	0.8894 ± 0.0364

Fonte: Elaborado pelos autores

Pode-se perceber que não houve melhora na métrica entre a procura de hiperparâmetros e no treino final, mesmo com dados aumentados e os giros na imagem, assim, foi criado um modelo com resultados considerados bons, com o desvio padrão baixo e com uma performance média de 6 fps.

Com o modelo B final, foi rodado o modelo para o mesmo conjunto de teste, mas apenas na área de atenção (Figura 4.11). As métricas extraídas se encontram na Tabela 4.4.

Figura 4.11: Área de atenção predita verde caminho, vermelho não caminho e azul ignorado no aprendizado



Fonte: Imagem retirada e editada de Cityscapes(CORDTS et al., 2016)

Tabela 4.4: Resultado do modelo final na área de atenção

IoU médio
Conjunto de teste
<i>média ± sd</i>
0.9287 ± 0.03656

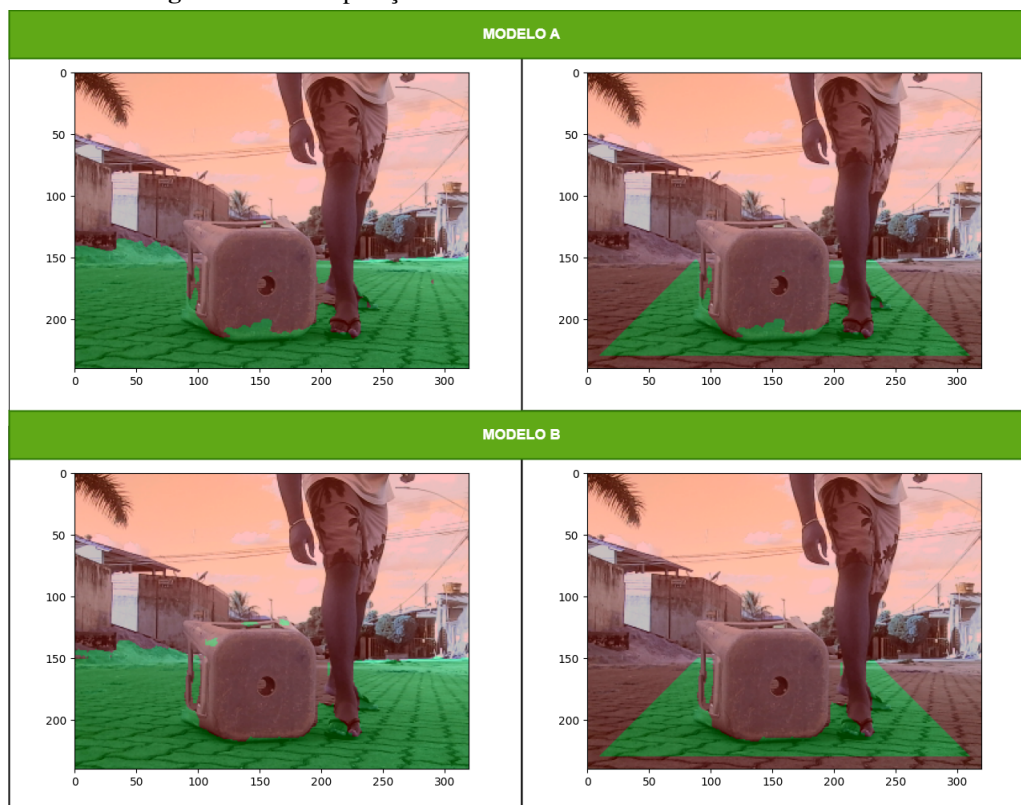
Fonte: Elaborado pelos autores

Nas métricas, existe uma pequena punição por causa da classe ignorada. Rodando o modelo para a área de atenção igualando a área ignorada nos dois lados, existe um aumento nos resultados das métricas para 0.9975 ± 0.0091 . Essa punição ocorre porque o modelo não usou aquela área para aprender, então aquela área se torna uma área ruim para determinar o resultado do modelo naquela imagem. Uma área que o modelo não ignora totalmente no treinamento, pois inversão da imagem vertical, horizontal e zoom fazem com que aquela área possa ter classes não

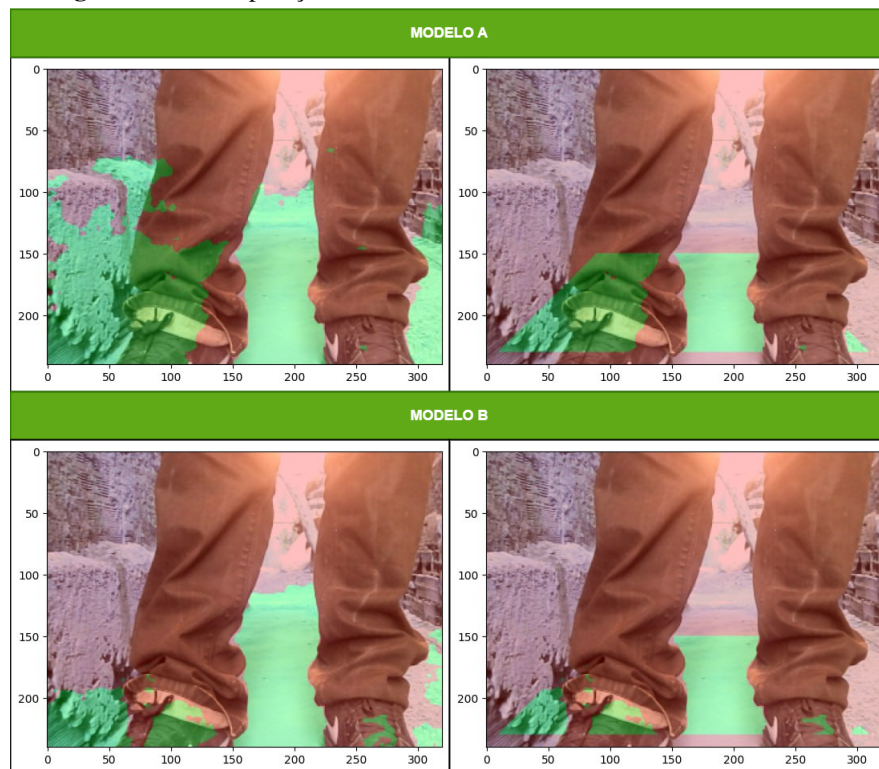
ignoradas. Mesmo com a punição, o modelo tem um resultado aceitável para a predição de ter ou não caminhos e ser usado para aumentar a segurança de alguém, não como algo principal, mas como auxiliar ao usuário.

O modelo B foi escolhido no lugar do modelo A pois, mesmo que tenha tido uma performance praticamente equivalente e sendo 2 fps mais devagar em performance, o modelo B possui mais filtros na penúltima camada de convolução, que passam mais características para a segmentação, e foi percebido que, em fotos de ruas brasileiras (Figura 4.13) e até de um corredor de uma casa (Figura 4.12), fora do conjunto de treinamento e da realidade que o modelo foi treinada, o modelo B tem uma segmentação mais coerente com o objetivo buscado, segmentando melhor áreas que não são caminhos e as bordas de caminhos, como com um banco de plástico, uma pessoa muito perto da câmera, ou a divisão da parede do corredor e do chão. Resultados que precisam ser quantificados e estudados em trabalhos futuros, para uma melhor conclusão.

Figura 4.12: Comparação entre o modelo A e o modelo B em uma rua



Fonte: Elaborado pelos autores

Figura 4.13: Comparação entre o modelo A e o modelo B em um corredor

Fonte: Elaborado pelos autores

4.3 Integração dos sistemas

Para integrar os sistemas, foi construído um código em python utilizando o conceito de threading para que os dois sistemas pudessem ser executados simultaneamente e conseguissem interagir entre si de acordo com a detecção das imagens pela rede neural. O sistema computacional conseguiu interferir no sistema mecânico, em alguns momentos havendo lentidão na resposta do hardware para enviar o sinal do sistema computacional para o código do sistema mecânico, mesmo já havendo o indício da detecção de objetos à frente. Porém, esse *delay* não é o suficiente para que em tese o protótipo venha a colidir com os obstáculos em seu caminho.

Devido à insuficiência da capacidade das baterias em fornecer energia adequada para movimentar o protótipo linearmente como argumentado na seção 4.1, as possibilidades de conduzir testes mais rigorosos sobre o deslocamento do modelo foram limitadas. Não obstante, nos experimentos feitos em que foram dispensados alguns pesos do sistema, foi constatado que, ao detectar um objeto dentro da área de atenção predita, o sistema computacional consegue enviar sinais para realizar o bloqueio e desbloqueio da aceleração para o código do sistema mecânico fazendo com o que a velocidade dos motores seja reduzida, ou, dependendo da distância do objeto, pode haver o bloqueio da movimentação bruscamente, levando os motores ao repouso.

5

Conclusão e trabalhos futuros

Foi construído um protótipo para representar uma cadeira rodas motorizada dispendo de um sistema inteligente que tem a finalidade de prevenção de acidentes.

O sistema computacional está concreto o suficiente para ser finalizado, no entanto, para finalizar todo o projeto, falta o arremate da mecânica das rodas para que seja possível que os motores façam a movimentação das rodas utilizando a transmissão por correntes, além de um cálculo mais assertivo para a corrente e a tensão elétrica necessária das baterias para conseguir a autonomia e a força necessárias para carregar uma pessoa sentada na cadeira de rodas motorizada. A integração do sistema de detecção de perigo com o controle da cadeira está funcionando de modo esperado, podendo haver melhorias a serem realizadas após toda a construção funcional da cadeira motorizada. E, por fim, construir a interface de controle utilizando o *joystick*, pois, até o presente momento foi utilizado um teclado sem fio para o controle dos motores.

Existe a pretensão de realizar a conclusão do projeto e futuramente testar o modelo sem a redução de parâmetros em um *hardware* com mais potência, fazendo uso de mais poder computacional para melhorar os resultados de performance de execução e tempo de resposta dos comandos. Bem como, possivelmente criar outro modelo de ML com detecção de objetos utilizando a delimitação inferior das detecções como análise para as paradas, no qual se tem um foco em objetos comuns nos percursos. Modelo esse que pode rodar em conjunto com a segmentação e com possíveis sensores de distância, criando um sistema de votação que aumenta a confiança do desempenho.

Os códigos desenvolvidos para o projeto podem ser encontrados no repositório do Github: <https://github.com/lipscheech/sistema-de-detecao-de-perigo.git> com seus respectivos comandos necessários para serem executados.

Referências

- ALIEXPRESS. **Motor escovado para scooter elétrica, 24v, 300w, my1016, com polia**. Access: 2022-06-19, <https://pt.aliexpress.com/item/32748528910.html?channel=twinner>.
- ALIEXPRESS. **Step Down Power Supply Buck Converter, DC-DC 24V, 12V a 5V, 5A**. Access: 2023-05-25, <https://pt.aliexpress.com/item/32785352477.html?channel=twinner>.
- BAREA, R. et al. System for assisted mobility using eye movements based on electrooculography. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, [S.l.], v.10, n.4, p.209–218, 2002.
- BOURHIS, G. et al. An autonomous vehicle for people with motor disabilities. **IEEE Robotics Automation Magazine**, [S.l.], v.8, n.1, p.20–28, 2001.
- CHASE, O. Sistemas embarcados. **Mídia Eletrônica. Página na internet:< www.sbjovem.org/chase>, capturado em**, [S.l.], v.10, n.11, p.13, 2007.
- CHEN, L. et al. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. **CoRR**, [S.l.], v.abs/1802.02611, 2018.
- CORDTS, M. et al. The Cityscapes Dataset for Semantic Urban Scene Understanding. In: **IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR). Proceedings...** [S.l.: s.n.], 2016.
- ELETROGATE. **Módulo Driver Ponte H Dupla - BTS7960**. Access: 2023-04-07, <https://www.eletrogate.com/ponte-h-dupla-bts7960>.
- FILHO, W. d. B. V. et al. **DESENVOLVIMENTO DE KIT PARA AUTOMAÇÃO DE CADEIRA DE RODAS CONVENCIONAL.** , Campina Grande, Paraíba, Brasil, 2010.
- FREEDOM. **Doação de cadeira de rodas: saiba mais sobre o assunto**. <https://blog.freedom.ind.br/doacao-de-cadeira-de-rodas-saiba-mais-sobre-o-assunto>.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.
- IBGE. **Censo Demográfico 2010 - Características gerais da população, religião e pessoas com deficiência**. Rio de Janeiro, Brasil, 2012.
- IVO, R. M. **SISTEMA DE CONTROLE DE CADEIRA DE RODAS MOTORIZADA PARA USÁRIOS PORTADORES DE TETRAPLEGIA.** , [S.l.], 2016.
- JAFFE, D. L. An Ultrasonic Head Position Interface for Wheelchair Control. **Journal of Medical Systems**, [S.l.], v.6, p.337–342, 1982.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep Learning. **Nature**, [S.l.], v.521, p.436–44, 05 2015.

- NAVARRO, W. F. **TRANSMISSÕES MECÂNICAS**. Access: 2022-06-19, <https://www.linkedin.com/pulse/transmiss%C3%B5es-mec%C3%A2nicas-walmir-fernandes-navarro/?originalSubdomain=pt>.
- RAMÍREZ, L. G. C.; JIMÉNEZ, G. S. A.; CARREÑO, J. M. **Sensores y actuadores**. [S.l.]: Grupo Editorial Patria, 2014.
- RASPBERRY PI. **Raspberry Pi 3 Model b+**. Access: 2023-07-31, <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>.
- REAL, A.; DORADO, F.; DURÁN, J. Energy Demand Forecasting Using Deep Learning: application to the french grid. , [S.l.], 03 2020.
- REIS, F. dos. Curso de Eletrônica – O que é PWM – Pulse Width Modulation. , [S.l.], Junho 2017. <http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/curso-de-eletronica-o-que-e-pwm-pulse-width-modulation/>.
- REIS, F. dos. Como funciona uma Ponte H – Controle direcional de motores DC. , [S.l.], Julho 2017. <http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/como-funciona-uma-ponte-h-contrrole-direcional-de-motores-dc/>.
- SANDLER, M. et al. Inverted Residuals and Linear Bottlenecks: mobile networks for classification, detection and segmentation. **CoRR**, [S.l.], v.abs/1801.04381, 2018.
- SILVEIRA, C. B. **Motor CC: saiba como funciona e de que forma especificar**. [S.l.: s.n.], 2016. <https://www.citisystems.com.br/motor-cc/>.
- SILVEIRA, M. G. Descrição de acidentes com usuários de cadeira de rodas. , [S.l.], 2019. Monografia (Bacharel em Terapia Ocupacional), UNIFESP (UNIVERSIDADE FEDERAL DE SÃO PAULO CAMPUS BAIXADA SANTISTA), Santos-SP, Brazil.
- UMMAT, S. K.; KIRBY, R. L. NONFATAL WHEELCHAIR-RELATED ACCIDENTS REPORTED TO THE NATIONAL ELECTRONIC INJURY SURVEILLANCE SYSTEM. **American Journal of Physical Medicine & Rehabilitation**, [S.l.], v.73, p.163–167, 1994. https://journals.lww.com/ajpmr/Abstract/1994/06000/NONFATAL_WHEELCHAIR_RELATED_ACCIDENTS_REPORTED_TO.4.aspx.
- WANG, Z. et al. Efficient Rail Area Detection Using Convolutional Neural Network. **IEEE Access**, [S.l.], v.6, p.77656–77664, 2018.
- ZEILER, M. D.; FERGUS, R. Visualizing and Understanding Convolutional Networks. **CoRR**, [S.l.], v.abs/1311.2901, 2013.