



Estacionamentos Inteligentes (*Smart Parking*)

Utilizando técnicas de aprendizagem profunda

Pedro Ivo Aragão Guimarães

Departamento de engenharia elétrica - DEE
Universidade Federal de Campina Grande - UFCG

Trabalho de Conclusão de Curso.
Orientador : Dr. Rafael Bezerra Correia Lima.

17 de outubro de 2021



Visão geral.

1. Motivação.
2. Conceitos.
3. Coleta e pré-processamento de dados.
4. Treino e validação do modelo.
5. Resultados e projeto.
6. Considerações finais.
7. Referências.



Seção.

1. Motivação.
2. Conceitos.
3. Coleta e pré-processamento de dados.
4. Treino e validação do modelo.
5. Resultados e projeto.
6. Considerações finais.
7. Referências.



Motivação.

- Entre os anos de 2008 e 2018, houve um número crescente de automóveis no país, que passou de 37,1 milhões para 65,7 milhões [Azevedo e Ribeiro 2019].
- 30% do congestionamento do tráfego é causado por buscas de locais para estacionar, que duram cerca de 7 a 8 minutos por tentativa [Arnott e Inci 2006].



Motivação.

- Entre os anos de 2008 e 2018, houve um número crescente de automóveis no país, que passou de 37,1 milhões para 65,7 milhões [Azevedo e Ribeiro 2019].
- 30% do congestionamento do tráfego é causado por buscas de locais para estacionar, que duram cerca de 7 a 8 minutos por tentativa [Arnott e Inci 2006].

Problemas causados

Falta de mobilidade urbana, estresse no trânsito, maior consumo de combustível e consequente aumento da poluição do ar.



Proposta para solução.

- A inteligência artificial (IA) busca métodos para solucionar este problema.
- Aprendizagem profunda vem se destacando nas pesquisas devido à:
 - A grande volume de dados disponíveis.
 - Melhoria na capacidade de processamento dos computadores.



Proposta para solução.

- A inteligência artificial (IA) busca métodos para solucionar este problema.
- Aprendizagem profunda vem se destacando nas pesquisas devido à:
 - A grande volume de dados disponíveis.
 - Melhoria na capacidade de processamento dos computadores.

Exemplo: Sistemas de visão computacional.

- Fornecer em tempo real informação sobre vagas para os motoristas.



Proposta para solução.

- A inteligência artificial (IA) busca métodos para solucionar este problema.
- Aprendizagem profunda vem se destacando nas pesquisas devido à:
 - A grande volume de dados disponíveis.
 - Melhoria na capacidade de processamento dos computadores.

Exemplo: Sistemas de visão computacional.

- Fornecer em tempo real informação sobre vagas para os motoristas.

Objetivo proposto

Oferecer um modelo classificador para vagas de estacionamentos, com técnicas de aprendizagem profunda.



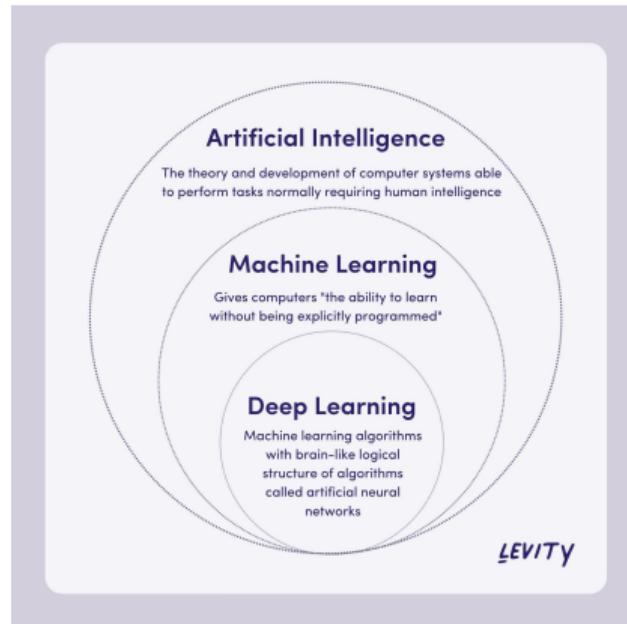
Seção.

1. Motivação.
2. Conceitos.
3. Coleta e pré-processamento de dados.
4. Treino e validação do modelo.
5. Resultados e projeto.
6. Considerações finais.
7. Referências.



Campo de atuação.

Figura 1: Campos da IA.



Levity



Redes Neurais Artificiais (RNA).

- As Redes neurais artificiais (RNA) são algoritmos de IA.



Redes Neurais Artificiais (RNA).

- As Redes neurais artificiais (RNA) são algoritmos de IA.
- As RNA são inspiradas no comportamento do cérebro humano (neurônios e sinapses).



Redes Neurais Artificiais (RNA).

- As Redes neurais artificiais (RNA) são algoritmos de IA.
- As RNA são inspiradas no comportamento do cérebro humano (neurônios e sinapses).
- A semelhança é utilizar elementos simples (neurônios), interligados, para resolver um problema.



Redes Neurais Artificiais (RNA).

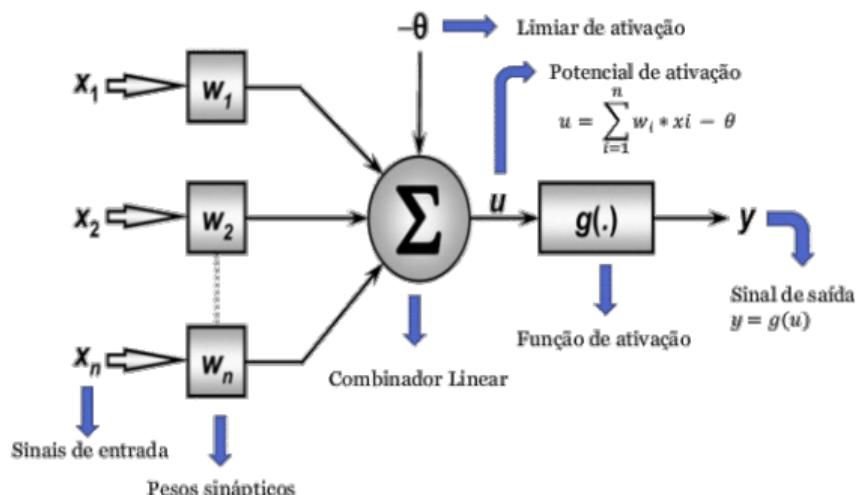
- As Redes neurais artificiais (RNA) são algoritmos de IA.
- As RNA são inspiradas no comportamento do cérebro humano (neurônios e sinapses).
- A semelhança é utilizar elementos simples (neurônios), interligados, para resolver um problema.

Compor funções lineares, conectadas em redes, para modelar um sistema complexo.



Exemplo de rede neural artificial.

Figura 2: Exemplo de RNA de uma camada.

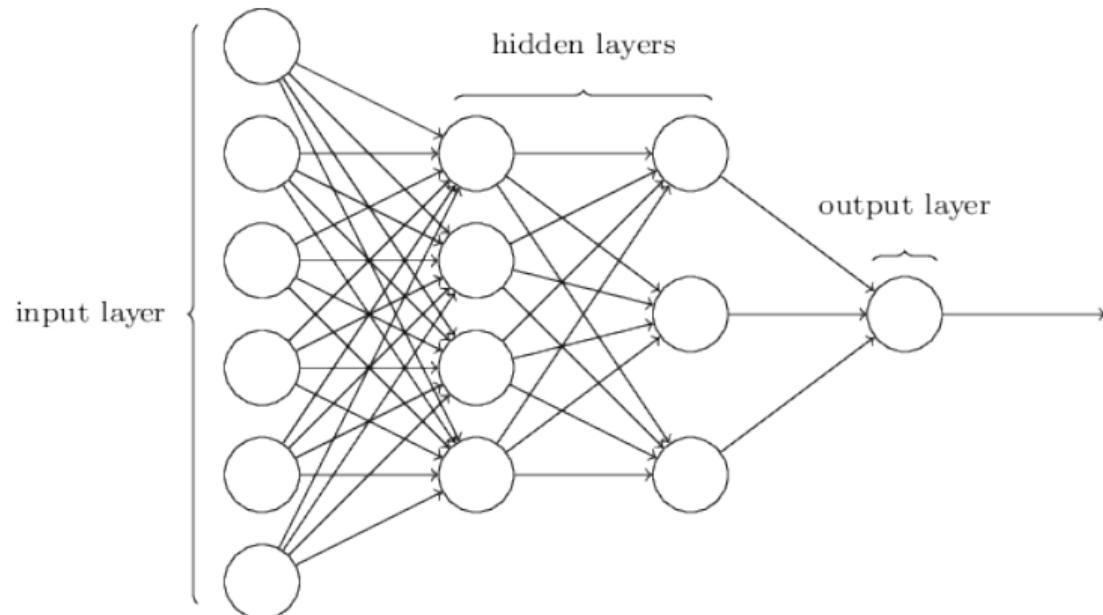


[Palmiere 2016]



Perceptron de Multicamadas (MLP).

Figura 3: MLP.



[Elipe 2020]



Gradiente descendente, Propagação e Retropropagação.

Gradiente descendente (∇)

Apontar a direção que a função deve seguir para minimizar a função de custo (erro).



Gradiente descendente, Propagação e Retropropagação.

Gradiente descendente (∇)

Apontar a direção que a função deve seguir para minimizar a função de custo (erro).

Propagação (*feedforward (FF)*) \Rightarrow

Compor funções simples por meio das funções de ativações.



Gradiente descendente, Propagação e Retropropagação.

Gradiente descendente (∇)

Apontar a direção que a função deve seguir para minimizar a função de custo (erro).

Propagação (*feedforward (FF)*) \Rightarrow

Compor funções simples por meio das funções de ativações.

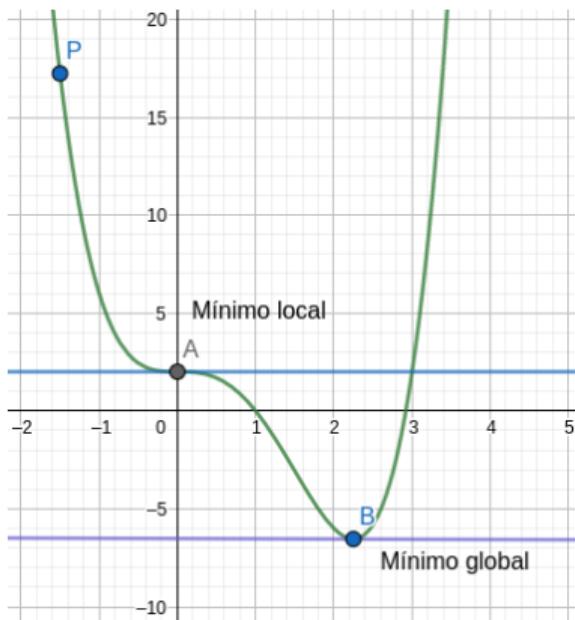
Retropopagação (*backpropagation (BP)*) \Leftarrow

Estimar o erro e a direção do gradiente para cada peso individualmente (derivadas parciais e regra de cadeia).



Mínimo local e mínimo global.

Figura 4: Função de custo.



Guimarães (2021)



Redes neurais convolucionais (Convnet / CNN).

- CNN é um algoritmo de *deep learning* para trabalhar com visão computacional.



Redes neurais convolucionais (Convnet / CNN).

- CNN é um algoritmo de *deep learning* para trabalhar com visão computacional.

Problema

Uma arquitetura tradicional MLP, cada pixel seria uma entrada na RNA (inviável).



Redes neurais convolucionais (Convnet / CNN).

- CNN é um algoritmo de *deep learning* para trabalhar com visão computacional.

Problema

Uma arquitetura tradicional MLP, cada pixel seria uma entrada na RNA (inviável).

Camadas Convolucionais

Convolução com um filtro para extrair os pixels mais significantes da imagem e formar mapas de características com as informações relevantes.

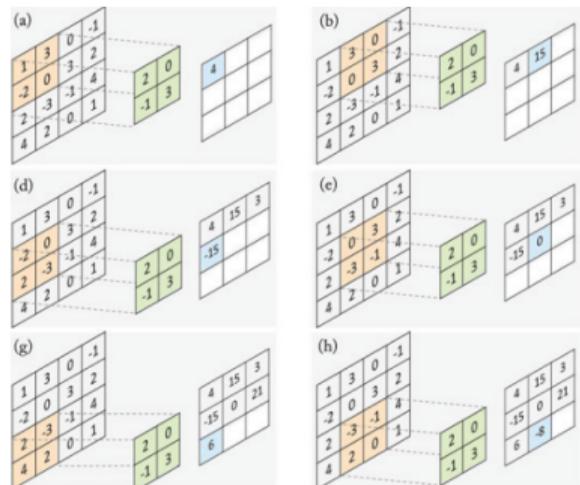
Camadas de amostragem (*pooling*)

Recebe o mapa de característica e condensa por meio de amostragem (*downsampling*).



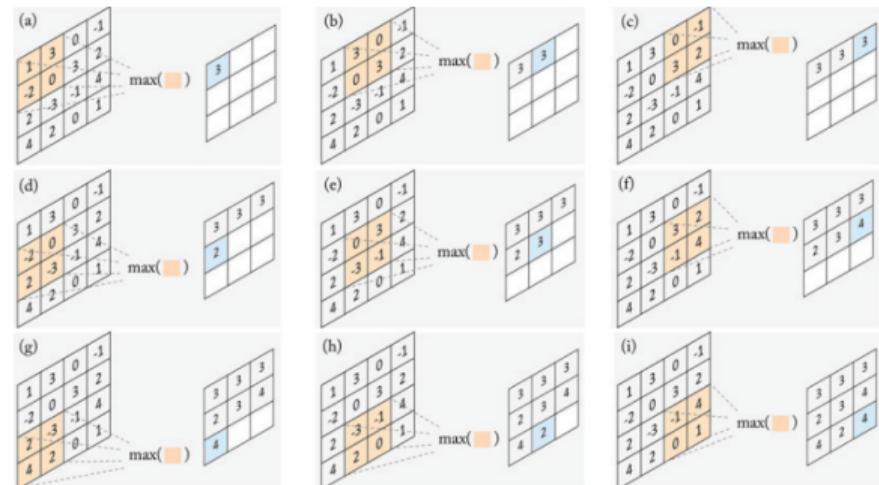
Exemplos de Convolução e Máx pooling.

Figura 5: Convolução.



[Khan et al. 2018]

Figura 6: Máx Pooling.



[Khan et al. 2018]



Aprendizado por transferência (*transfer learning*).

Uma das grandes vantagens do algoritmo de CNN é a possibilidade de reaproveitar a arquitetura e pesos (ou parte) de uma CNN pré-treinada.



Seção.

1. Motivação.
2. Conceitos.
3. Coleta e pré-processamento de dados.
4. Treino e validação do modelo.
5. Resultados e projeto.
6. Considerações finais.
7. Referências.



Coleta de dados.

- A coleta de dados é uma das fases mais importantes em um projeto de aprendizagem profunda.



Coleta de dados.

- A coleta de dados é uma das fases mais importantes em um projeto de aprendizagem profunda.
- A quantidade e qualidade dos dados são os grandes responsáveis pelo crescimento deste algoritmo nos últimos anos.



Coleta de dados.

- A coleta de dados é uma das fases mais importantes em um projeto de aprendizagem profunda.
- A quantidade e qualidade dos dados são os grandes responsáveis pelo crescimento deste algoritmo nos últimos anos.
- “Comparado com o aprendizado de máquina tradicional, o aprendizado profundo é mais adequado para o processamento de *big data*, pois o desempenho do algoritmo melhora com o aumento do volume de dados” [Goodfellow, Bengio e Courville 2016].



dataset PKLot.

- Banco de dados desenvolvido por pesquisadores da Universidade Federal do Paraná.



dataset PKLot.

- Banco de dados desenvolvido por pesquisadores da Universidade Federal do Paraná.
- Todas as imagens foram retiradas de estacionamentos da Universidade Federal do Paraná (UFPR) e da Pontifícia Universidade Católica do Paraná (PUCPR).



dataset PKLot.

- Banco de dados desenvolvido por pesquisadores da Universidade Federal do Paraná.
- Todas as imagens foram retiradas de estacionamentos da Universidade Federal do Paraná (UFPR) e da Pontifícia Universidade Católica do Paraná (PUCPR).
- Contém 12.417 imagens dos estacionamentos e 695.899 ROI'S de vagas que foram manualmente checadas e rotuladas.



dataset PKLot.

- Banco de dados desenvolvido por pesquisadores da Universidade Federal do Paraná.
- Todas as imagens foram retiradas de estacionamentos da Universidade Federal do Paraná (UFPR) e da Pontifícia Universidade Católica do Paraná (PUCPR).
- Contém 12.417 imagens dos estacionamentos e 695.899 ROI'S de vagas que foram manualmente checadas e rotuladas.
- O *dataset* é composto por três base dados diferentes: PUCPR, UFPR04 e UFPR05.



Tabela 1: Resumo do *dataset* PKLOT

LOCAL	Condições	Dias	Img.	Ocupada	Vazia	Total
UFPR04	ENSOLARADO	20	2098	32166 (54.98%)	26334 (45.02%)	58400
	NUBLADO	15	1408	11608 (29.47%)	27779 (70.53%)	39387
	CHUVOSO	14	285	2351 (29.54%)	5607 (70.46%)	7958
	TOTAL	49	3791	46125 (43.58%)	59720 (56.42%)	105845
UFPR05	ENSOLARADO	25	2500	57584 (57.65%)	42306 (42.35%)	99890
	NUBLADO	19	1426	33764 (59.27%)	23202 (40.73%)	56966
	CHUVOSO	8	226	6078 (68.07%)	2851 (31.93%)	8929
	TOTAL	52	4152	97426 (58.77%)	68359 (41.23%)	165785
PUCPR	ENSOLARADO	24	2315	96762 (46.42%)	101672 (53.58%)	208433
	NUBLADO	11	1328	42363 (31.90%)	90417 (68.10%)	132780
	CHUVOSO	8	831	55104 (66.35%)	27951 (33.65%)	83056
	TOTAL	43	4474	194229 (45.78%)	230040 (51.46%)	424269



Figura 7: PKlot.



[de Almeida et al. 2015]



Segmentação das regiões de interesses (ROI's)

Segmentação

Segmentação das vagas refeitas para generalização do projeto.

Extração das ROIs

Aumento do *time lapse* de 5 para 30 minutos para evitar repetições de imagens no treinamento do modelo.



Segmentação das regiões de interesses (ROI's)

Figura 8: demarcando ROIs.

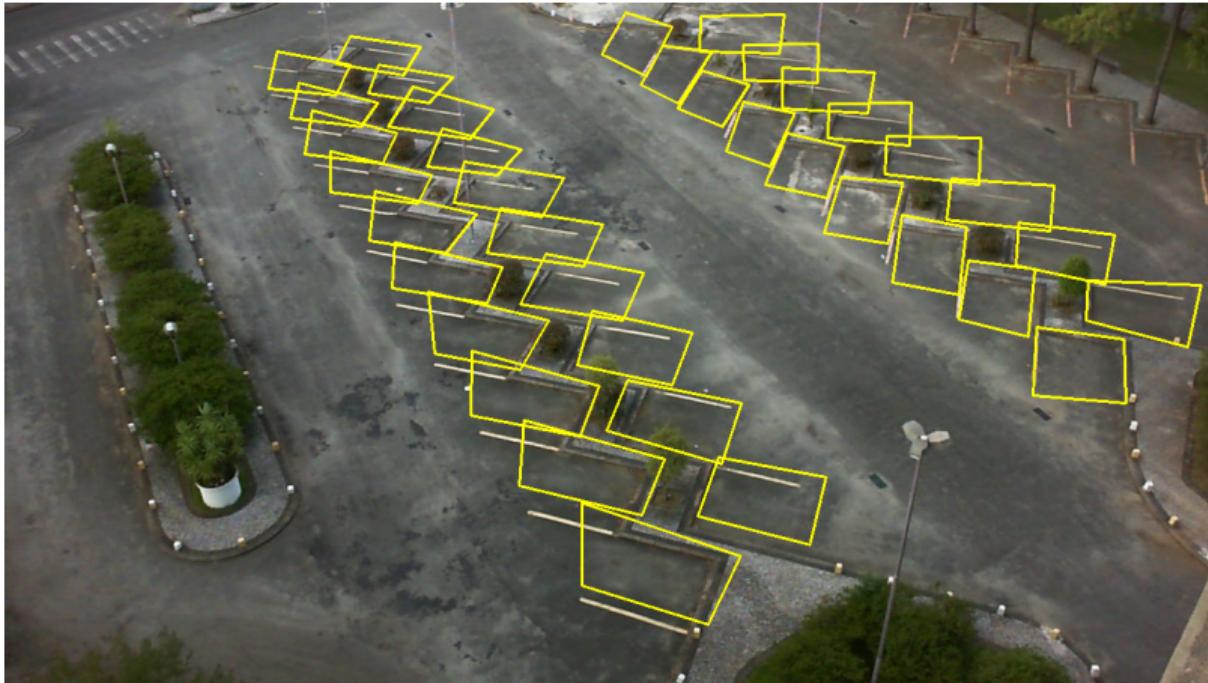
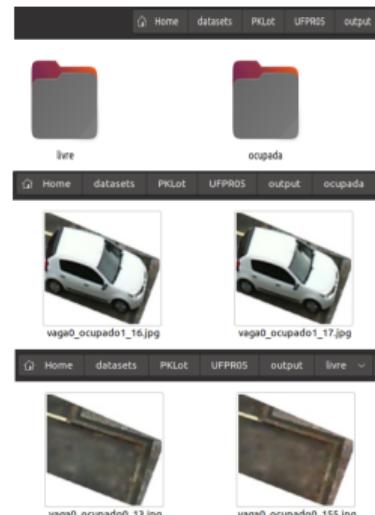


Figura 9: Segmentação.

Tabela 2: Tabela das ROI'S

LOCAL	Qtd. imagens	Vagas Ocupada	Vagas Vazia	Total
UFPR04	3791	5336	5248	58400
UFPR05	4152	16292	11388	27680
PUCPR	4474	21598	33302	54900



Seção.

1. Motivação.
2. Conceitos.
3. Coleta e pré-processamento de dados.
- 4. Treino e validação do modelo.**
5. Resultados e projeto.
6. Considerações finais.
7. Referências.



Arquitetura do modelo.

Figura 10: Arquitetura do modelo.

Layer (type)	Output Shape	Param #
<hr/>		
input_2 (InputLayer)	[(None, 160, 160, 3)]	0
rescaling_1 (Rescaling)	(None, 160, 160, 3)	0
mobilenetv2_1.00_160 (Function)	(None, 5, 5, 1280)	2257984
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dropout (Dropout)	(None, 1280)	0
dense (Dense)	(None, 1)	1281
<hr/>		
Total params: 2,259,265		
Trainable params: 1,281		
Non-trainable params: 2,257,984		



Hiper parâmetros.

- Número de camadas densas = 1.
- $\text{Dropout} = 20\%$.
- Função de ativação - sigmoide.
- Função de perda - *BinaryCrossentropy* (Utilizada para problemas de classificação).
- Taxa de aprendizado = 10^{-4} .
- Número de épocas = 10.
- Tamanho do lote (*batch size*) = 32.



Seção.

1. Motivação.
2. Conceitos.
3. Coleta e pré-processamento de dados.
4. Treino e validação do modelo.
5. Resultados e projeto.
6. Considerações finais.
7. Referências.



Treinamento - UFPR04.

Tabela 3: Resumo do treinamento UFPR04.

UFPR04 - 28 vagas	Acurácia	Perda
Avaliação na base de treino	0,9902	0,0388
Avaliação na base de validação	0,9907	0,0410
Avaliação na base de teste	0,9918	0,0441
Épocas	10	
Lote	32	
Taxa de aprendizado	0,0001	
<i>Dropout</i>	20%	
Otimizador	Adam	
Função de Perda	<i>BinaryCrossentropy</i>	
Métricas	Precisão	



Figura 11: Precisão e perda - UFPR04.

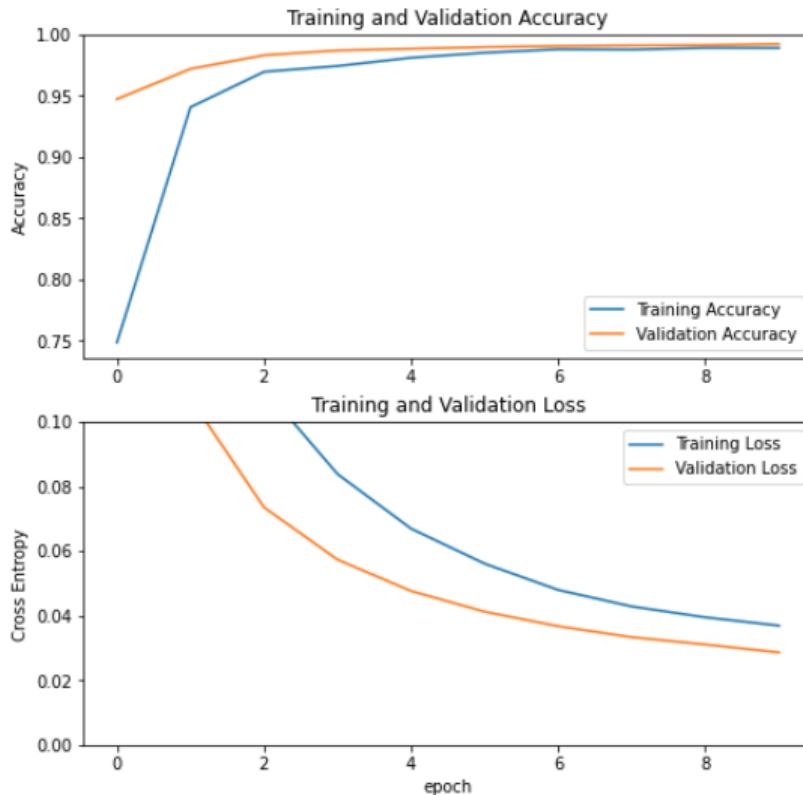


Figura 12: Predição nas ROIs - UFPR04.



Treinamento - UFPR05.

Tabela 4: Resumo do treinamento UFPR05.

UFPR05 - 37 vagas	Acurácia	Perda
Avaliação na base de treino	0,9922	0,0287
Avaliação na base de validação	0,9923	0,0286
Avaliação na base de teste	0,9920	0,0285
Épocas	10	
Lote	32	
Taxa de aprendizado	0,0001	
<i>Dropout</i>	20%	
Otimizador	Adam	
Função de Perda	<i>BinaryCrossentropy</i>	
Métricas	Precisão	



Figura 13: Precisão e perda - UFPR05.

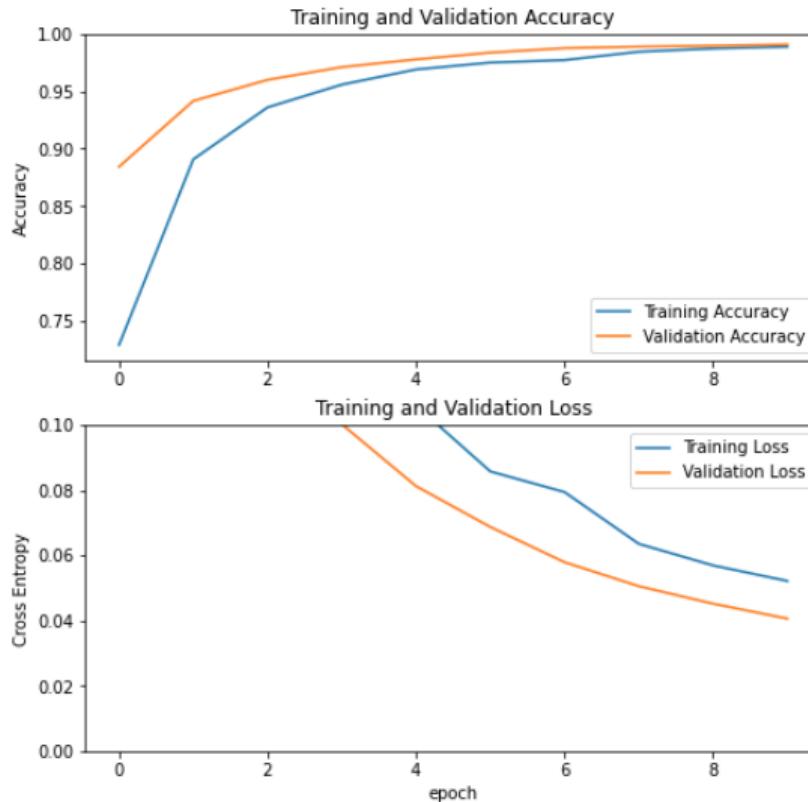
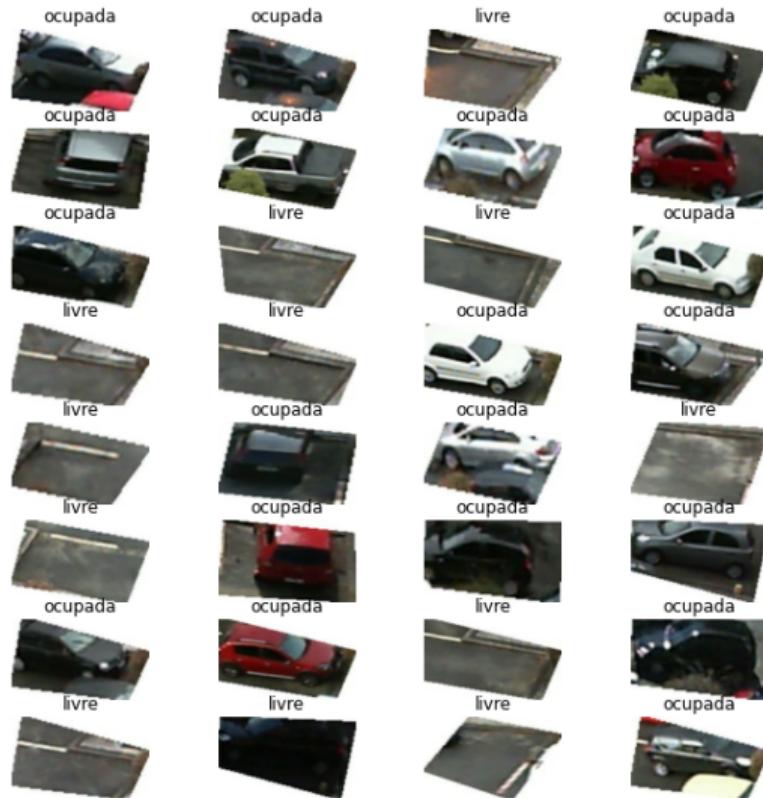


Figura 14: Predição nas ROIs - UFPR05.



Treinamento - PUCPR.

Tabela 5: Resumo do treinamento PUCPR.

PUCPR - 100 vagas	Acurácia	Perda
Avaliação na base de treino	0,9313	0,1976
Avaliação na base de validação	0,9267	0,2069
Avaliação na base de teste	0,9267	0,2061
Épocas	10	
Lote	32	
Taxa de aprendizado	0,0001	
<i>Dropout</i>	20%	
Otimizador	Adam	
Função de Perda	<i>BinaryCrossentropy</i>	
Métricas	Precisão	



Figura 15: Precisão e perda - PUCPR.

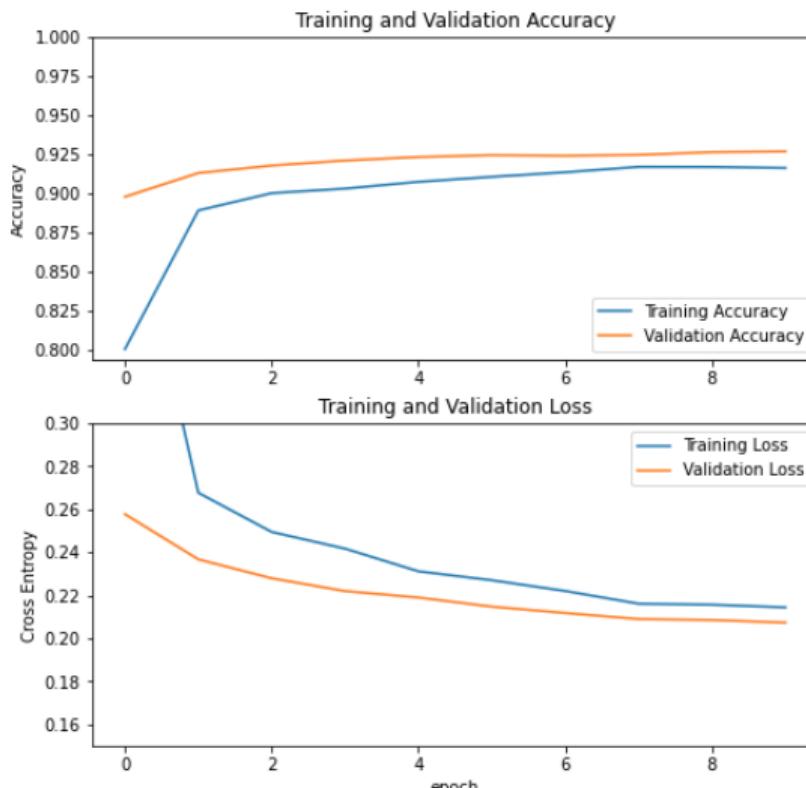
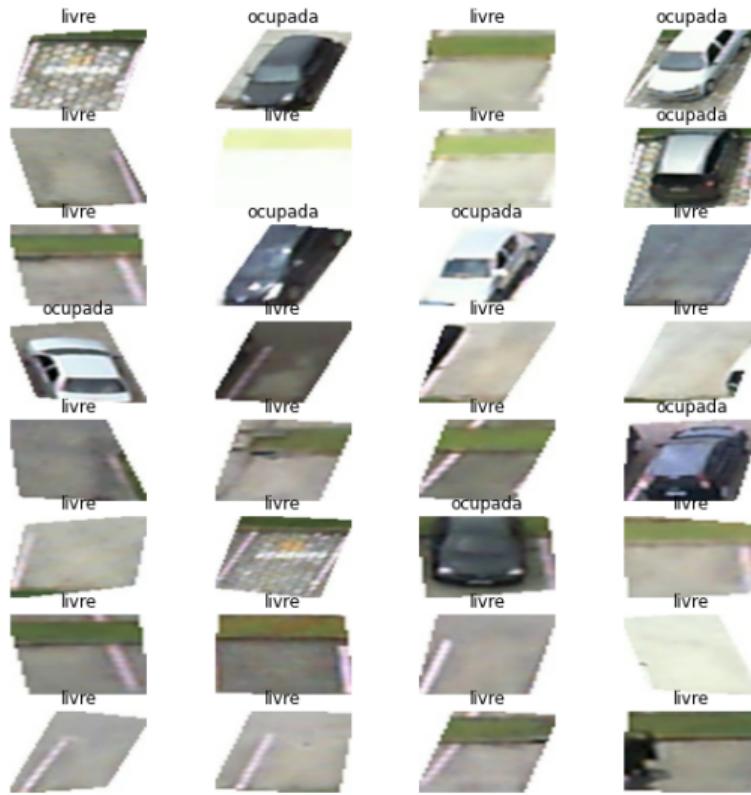


Figura 16: Predição nas ROIs - PUCPR.



Projeto.

Carregar imagens aleatórias do estacionamento na base de dados do UFPR05.

Fazer a predição em tempo real das 37 vagas.



Figura 17: Fluxograma.

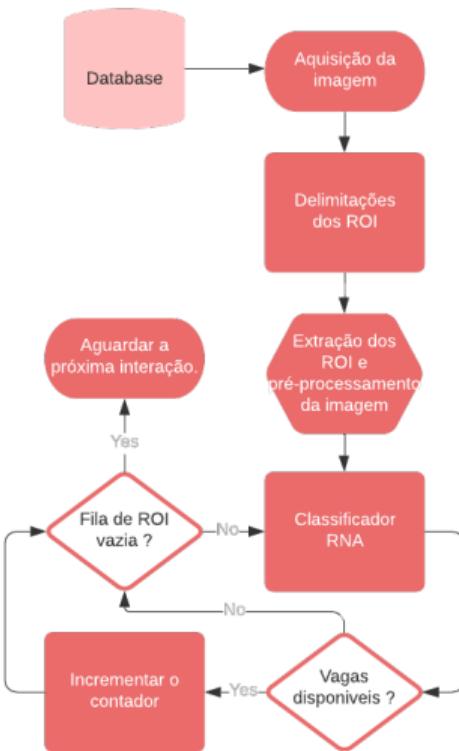
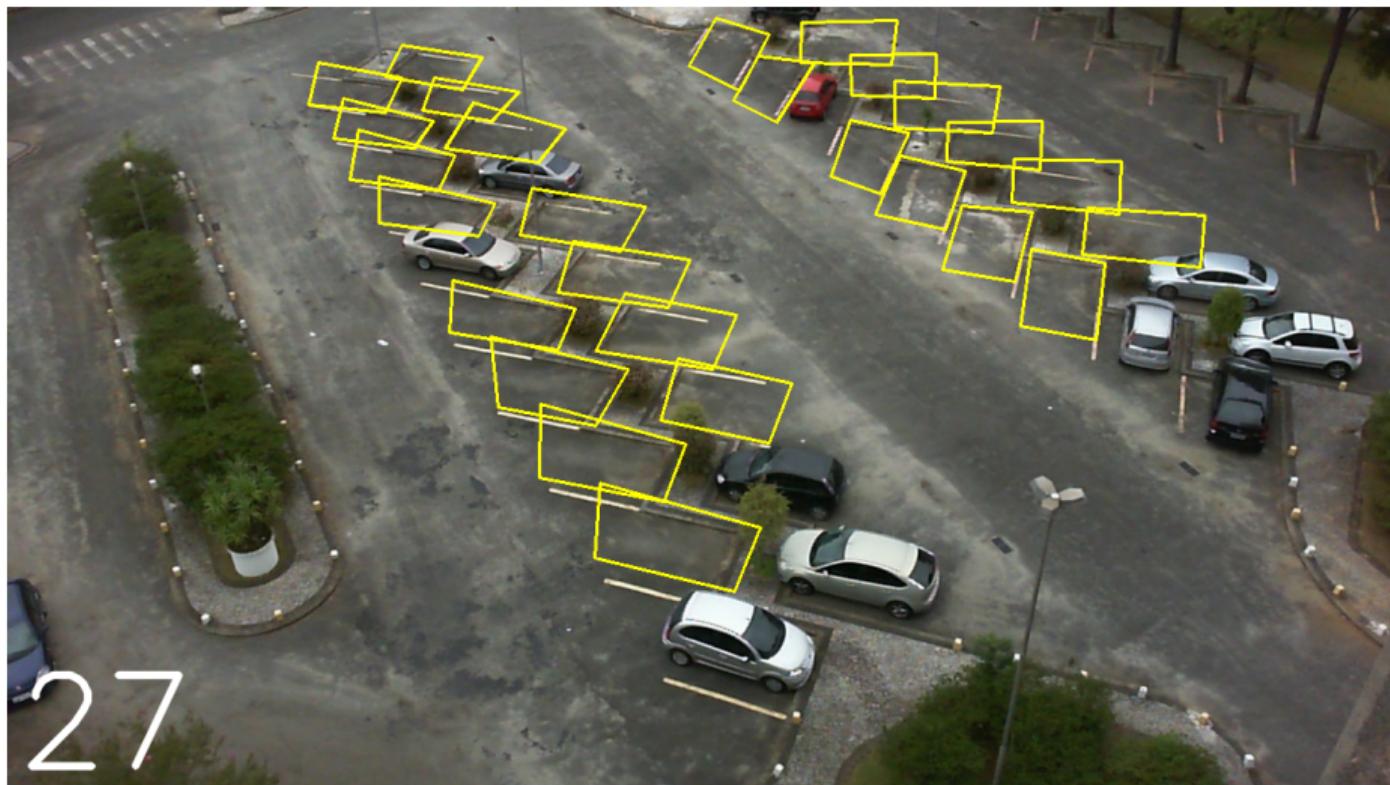
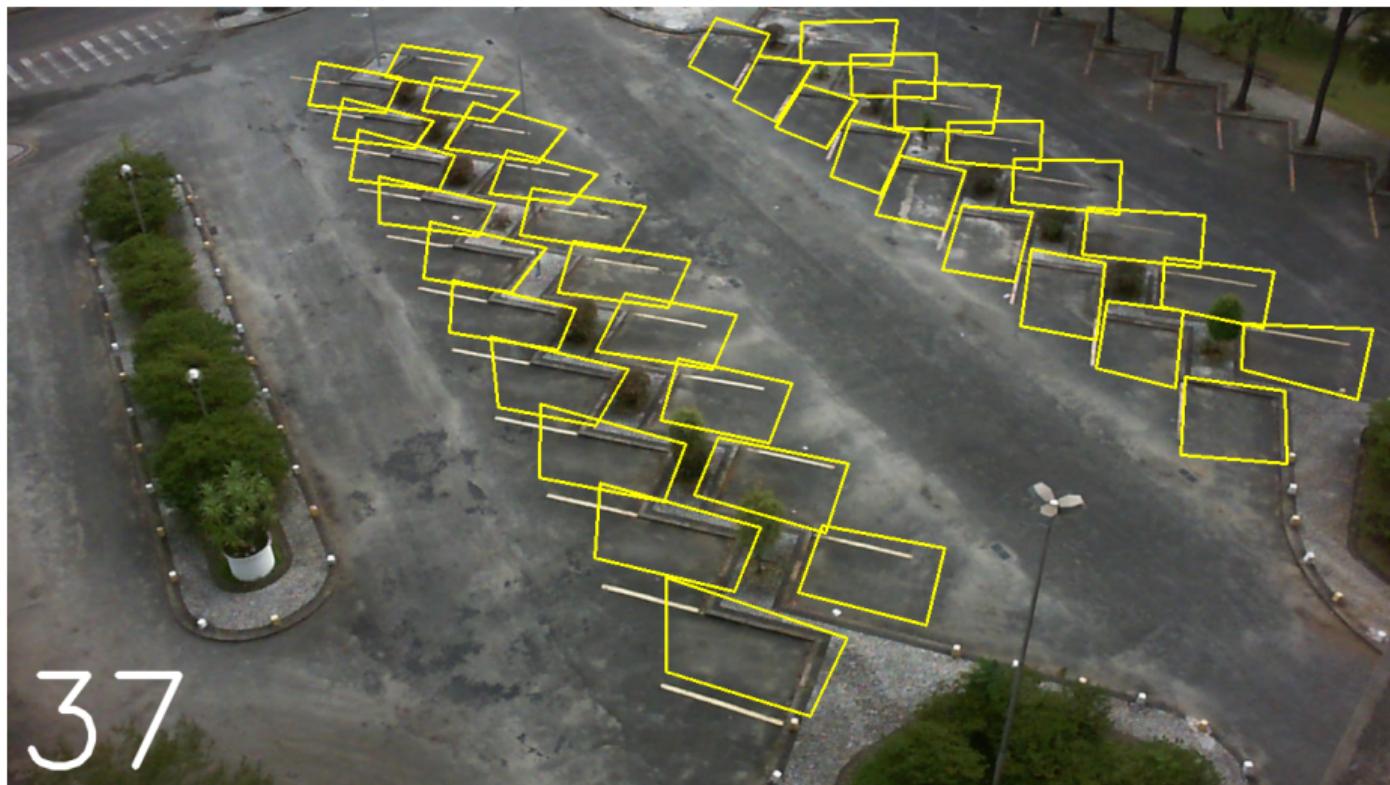


Figura 18: Predição do sistema.



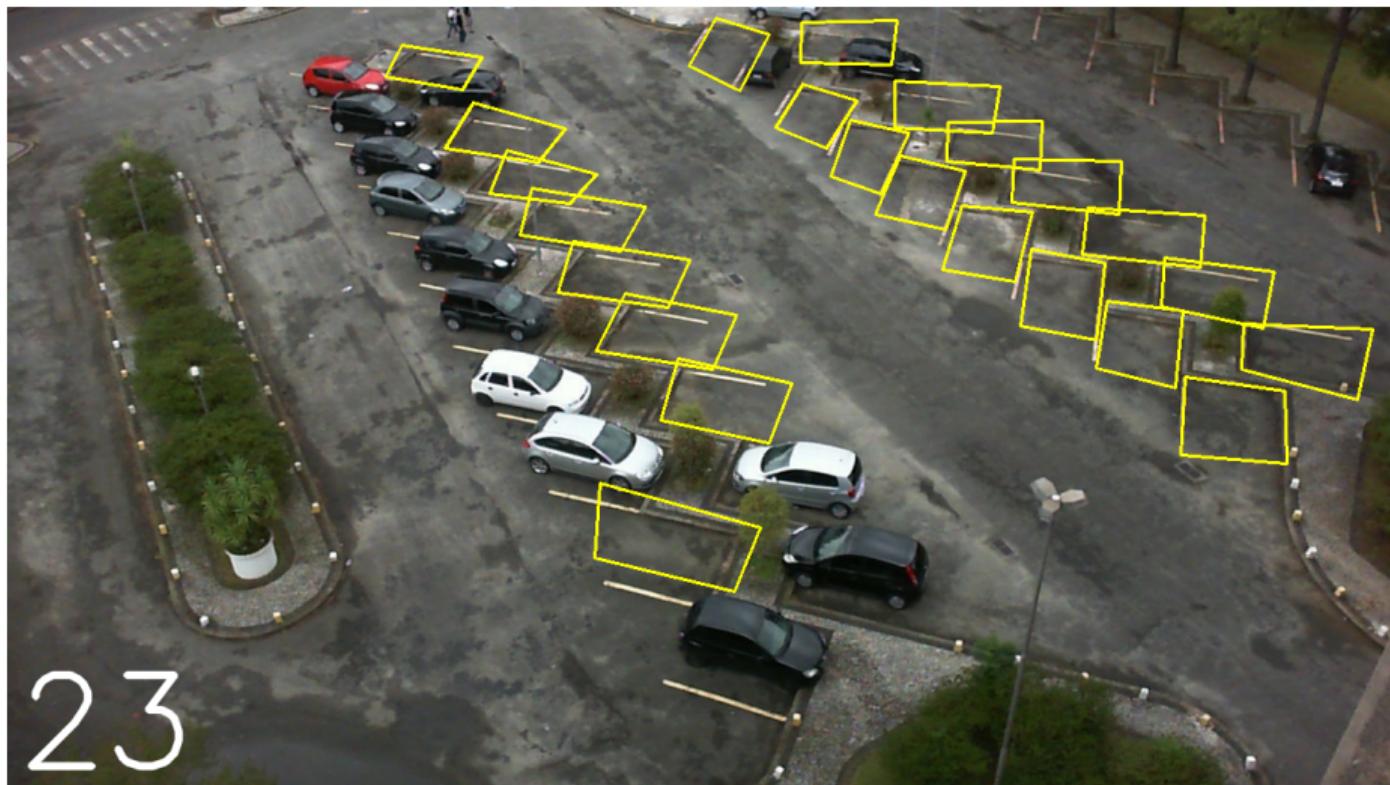
27

Figura 19: Predição do sistema.



37

Figura 20: Predição do sistema.



23

Seção.

1. Motivação.
2. Conceitos.
3. Coleta e pré-processamento de dados.
4. Treino e validação do modelo.
5. Resultados e projeto.
6. Considerações finais.
7. Referências.



Avaliação dos resultados.

Os resultados do projeto do classificador foram satisfatórios, equiparado a literatura atual em resultados e em metodologia.

A predição das 37 vagas (UFPR05) ocorre com baixa latência (2s), com precisão alta. A metodologia aplicada é compatível com as pesquisas atuais na área

TRABALHOS FUTUROS:

- Validar os resultados em outras bases de dados.
- Adicionar as etapas de implementações e monitoramento.
- Sintonizar os hiper parâmetros do modelo.
- Criar interface móvel para o público alvo.
- Publicar artigos para disseminar o conhecimento científico obtido.



Obrigado!¹

¹Todo o material de código-fonte deste projeto está disponível neste link no github em um repositório público com o passo a passo para a reprodução.



Seção.

1. Motivação.
2. Conceitos.
3. Coleta e pré-processamento de dados.
4. Treino e validação do modelo.
5. Resultados e projeto.
6. Considerações finais.
7. Referências.



-  ARNOTT, R.; INCI, E. An integrated model of downtown parking and traffic congestion. *Urban Economics*, v. 60, n. 3, p. 418–442, 2006.
-  AZEVEDO, S.; RIBEIRO, L. *Mapa da motorização individual no Brasil*. 2019. Disponível em: <https://www.observatoriodasmetropoles.net.br/wp-content/uploads/2019/09/mapa_moto2019v2.pdf>. Acesso em: 28 de julho de 2021.
-  de Almeida, P. R. et al. Pklot – a robust dataset for parking lot classification. *Expert Systems with Applications*, v. 42, n. 11, p. 4937–4949, 2015. ISSN 0957-4174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417415001086>>.
-  ELIPE, D. R. *Rede Perceptron de uma única camada*. 2020. Disponível em: <<https://github.com/d-r-e/multilayer-perceptron>>. Acesso em: 05 de outubro de 2021.
-  GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.I.]: MIT Press, 2016. <http://www.deeplearningbook.org>.
-  KHAN, S. et al. *A Guide to Convolutional Neural Networks for Computer Vision*. [S.I.: s.n.], 2018.



 PALMIERE, S. E. *Rede Perceptron de uma única camada.* 2016. Disponível em: <<https://www.embarcados.com.br/rede-perceptron-de-uma-unica-camada/>>. Acesso em: 02 de outubro de 2021.

