

Construção de uma metodologia para concepção de um veículo autônomo não tripulado de baixo custo

Building a methodology for designing a low cost unmanned autonomous vehicle

RESUMO

Este artigo descreve a metodologia aplicada para construir uma rede neural convolucional usada no projeto de um módulo autônomo de baixo custo. As redes neurais convolucionais são amplamente utilizadas para criar agentes que têm a capacidade de classificar imagens com base nos padrões nelas encontrados, podendo assim encontrar características específicas dos animais, e até mesmo a possibilidade de criar veículos autônomos através dos padrões presentes nas rodovias. Assim, por meio de um problema teórico como percorrer corredores, foi possível determinar os parâmetros necessários e a viabilidade de se realizar a construção de um veículo autônomo de baixo custo, a partir do qual a captação de dados alcançou uma precisão de 76%.

PALAVRAS-CHAVE: Processamento de imagens. Raspberry Pi. Redes neurais.

ABSTRACT

This paper describes the methodology applied to build a convolutional neural network used in the design of a low-cost standalone module. Convolutional neural networks are widely used to create agents that have the ability to classify images based on patterns found in them, thus being able to find specific characteristics of animals, and or even the possibility of creating autonomous vehicles through the patterns present in highways. Thus, through a theoretical problem such as traveling through corridors, it was possible to determine the necessary parameters and the feasibility of carrying out the construction of a low-cost autonomous vehicle, wherefrom the data capture it was able to reach an accuracy of 76%.

KEYWORDS: Image processing. Raspberry Pi. Neural Networks.

Daniel Peixoto Pinto da Silva
peixoto.daniel@hotmail.com
Universidade Tecnológica do
Paraná, Medianeira, Paraná, Brasil

Marcos Tiago Araújo de França
tiagofranca04@gmail.com
Universidade Tecnológica do
Paraná, Medianeira, Paraná, Brasil

Pedro Luiz de Paula
plpf2004@gmail.com
Universidade Tecnológica do
Paraná, Medianeira, Paraná, Brasil

Arnaldo Candido Junior
arnaldocan@gmail.com
Universidade Tecnológica do
Paraná, Medianeira, Paraná, Brasil

Recebido: 19 ago. 2019.

Aprovado: 01 out. 2019.

Direito autorial: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

As Redes Neurais Artificiais (RNAs) são modelos computacionais inspirados no funcionamento do cérebro humano e seus neurônios biológicos, assim sendo capazes de realizar o aprendizado de máquina e reconhecimento de padrões. Entre as arquiteturas de RNAs disponíveis se destacam as Redes Profundas (do Inglês, *Deep Networks*), contendo múltiplas camadas intermediárias para realizar o processamento de informação (DENG, 2014). Entre as Redes Profundas disponíveis, as Redes Neurais Convolucionais (CNN do inglês *Convolution Neural Network*) são utilizadas na resolução de problemas que envolvam visão computacional como reconhecimento e classificação de objetos, incluindo a concepção de veículos autônomos.

Veículo autônomo designa qualquer tipo de veículo terrestre com capacidade de transporte de pessoas ou bens sem a necessidade de um condutor humano (PISSARDINI, *et al.*, 2013). O seu principal objetivo é utilizar agentes a fim de determinar e executar o melhor conjunto de ações de forma mais segura que condutores humanos, assim reduzindo as consequências promovidas pelas falhas humanas.

Grandes empresas como a *Nvidia* (BOJARSKI, *et al.*, 2016) e Google (POCZTER; JANKOVIC, 2013) já utilizam redes convolucionais para concepção de veículos autônomos, através de dados visuais em conjunto com sensores atingindo bons resultados com esta combinação. Devido os trabalhos existentes envolverem altos custos de produção e serem de tecnologia proprietária, este trabalho visa propor a construção de uma metodologia de aplicação de baixo custo. Para o experimento foram utilizados os corredores da universidade onde este módulo deve ser capaz de aprender padrões para movimentação autônoma.

MATERIAIS E MÉTODOS

Para a construção da estrutura física do módulo, nomeado de M.A.R.C. (Módulo Autônomo de Reconhecimento Cognitivo), foi utilizado o conjunto Lego Mindstorms, já que possui diversos sensores e fácil montagem de seus componentes.

Porém, à medida que não apenas a captura de dados era realizada pelo módulo como também o pré-processamento, foi necessária uma atualização do microcontrolador de hardware considerando que o microcontrolador padrão do EV3 é dotado de um processador com *clock* de apenas 300 MHz.

Assim foi necessário buscar outras soluções, e a encontrada foi substituir o microcontrolador presente por um Raspberry Pi 3, que possui um processador de 1.2 GHz. Para os sensores e motores foi acoplado um módulo adaptador conhecido BrickPI 3.

Como sistema operacional o módulo robótico M.A.R.C utiliza-se do EV3DEV¹, sendo esta uma distribuição Linux baseada no Debian 8, capaz de se comunicar com os motores e sensores de forma nativa.

Com o objetivo de melhor aproveitar processamento dos dispositivos, foi utilizado uma estrutura distribuída baseada no Protocolo TCP/IP Cliente/Servidor. Para este feito foi codificado um software que permitiu a comunicação entre o módulo M.A.R.C. e o dispositivo móvel, esse foi chamado de M.A.R.C.O.S. (Módulo Autônomo de Reconhecimento Cognitivo Operating System).

Além de permitir a comunicação através do protocolo TCP/IP, o M.A.R.C.O.S. funciona como uma *dashboard* para as operações de execução de diagnóstico em tempo real. Para garantir a integridade dos dados capturados foi elaborado um protocolo de comunicação de forma que fossem identificados as operações executadas, dispositivo de origem, dispositivo de destino, e valores referentes, este protocolo ficou conhecido como protocolo RST (Robô, Smartphone, Tablet).

Para os dispositivos que funcionam como clientes (módulo robótico e o dispositivo móvel), foi construído o S.C.D. (Sistema de Captura de Dados). Sendo programado em Python o S.C.D. através das instruções do M.A.R.C.O.S. é responsável pelas capturas de dados e validação. Além das funções referentes a captura o S.C.D. também é responsável pelo envio de informações em tempo real para a *dashboard* como dados dos sensores do módulo robótico em tempo real.

Após o fim da captura de dados, o módulo robótico gera um arquivo de *log* com todos os dados referentes aquela determinada captura no formato marcset, como o início e fim, movimentos realizados e etc. Já o dispositivo móvel deverá gerar dois arquivos, sendo o primeiro um *log* no formato marcset onde estarão dados como tempo de início e fim de captura e o segundo o arquivo de vídeo no formato MP4 referente a captura. No fim após estes arquivos serem gerados o S.C.D. iniciará uma sessão SSH com o dispositivo móvel para validar a integridade de todos os arquivos referentes a captura, caso tudo esteja de acordo estes arquivos serão processados, do contrário serão descartados.

Após verificada a integridade dos arquivos é feita uma normalização dos dados. Os *frames* são extraídos do vídeo através do software FFMPEG². O módulo capturou dados a cada um segundo, devido à necessidade de garantir integridade dos dados. Após a extração, estes *frames* são salvas no disco para a etapa de pré-processamento.

Após as imagens estarem separadas do vídeo é necessário realizar correções de forma que possam ser alimentadas na CNN. Primeiramente é feito um recorte superior e inferior. Após feito o recorte é realizada uma operação matricial de rotação em 90°, tornando-a horizontal, através do uso da biblioteca OpenCV, as imagens representarão fielmente como o mundo é visualizado por humanos.

Após as rotações tem-se uma imagem com dimensões de 480x800px, em seguida é feito o recorte resultando em uma imagem com dimensões de 480x478px. A próxima etapa insere as intenções de movimento de forma que sejam visualizados pela rede. Logo depois a imagem é redimensionada para 224x224 *pixels*. Após este processo os dados estão prontos para serem treinados.

¹<http://ev3dev.org>

²<https://ffmpeg.org/>

A GoogleLeNet (InceptionV3), vencedora da competição ILSVRC 2015, foi escolhida para treinar o modelo. Ela consiste em CNN de 48 camadas de profundidade e aproximadamente 7 milhões de parâmetros (SZEGEDY *et al*, 2016). Esse modelo é o auge de muitas ideias desenvolvidas por vários pesquisadores ao longo dos anos, assim se mostrando ideal para utilização neste experimento.

Para o experimento foi utilizado o método de *Data Augmentation*, que consiste em aumentar significativamente a diversidade de dados disponíveis para modelos de treinamento, sem necessidade de novas coletas (PEREZ; WANG, 2017). Para imagens é necessária alteração sem retirar características essenciais presentes nelas, para que a rede ainda consiga classificar de maneira correta. Técnicas como rotação horizontal, rotação e *crop* são comumente utilizadas. Outras como correção de histograma e ajuste de gama também podem ser utilizadas.

RESULTADOS E DISCUSSÃO

PROCESSO DE CAPTURAS

Obteve-se um total de 1.732 imagens para o treinamento na *ConvNet*. Esse número é pouco expressivo comparado a outros *datasets* presentes como o MNIST *dataset* que possui um total de 70.000 imagens.

Inicialmente visou-se obter grandes quantidade de imagens de forma rápida, devido à necessidade de validar a metodologia desenvolvida de captura na rede de forma que os resultados fossem promissores, após esta verificação se confirmar iniciou-se o processo de balanceamento das capturas com base nos respectivos movimentos executados pelo módulo robótico durante o treinamento.

Este processo de balanceamento é extremamente necessário pois durante uma captura a maior parte das instâncias são referentes a movimentos frontais 'frente', gerando o efeito de *overfitting* na *ConvNet*. Todas as capturas foram realizadas no período da manhã ou em dias nublados. Porque percebeu-se que durante dias ensolarados a luminosidade no final do corredor era muito alta comprometendo a qualidade da imagem capturada pela câmera do dispositivo móvel.

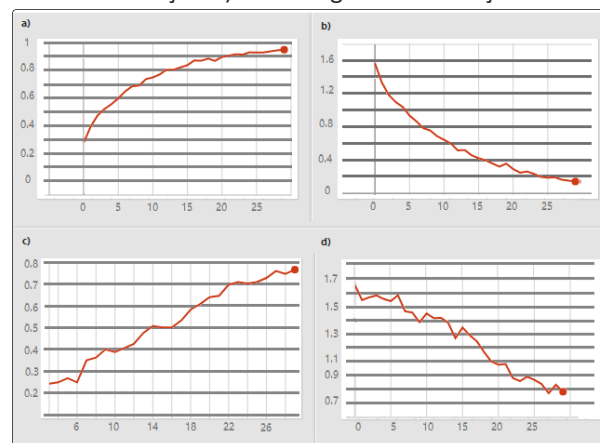
TREINAMENTO

Com o objetivo de validar a hipótese proposta para este trabalho, foi utilizado um modelo pré-treinado da rede Inception v3. Antes de tudo é adicionado uma faixa ao topo das imagens do conjunto para indicar qual é seu comando, sendo eles frente, esquerda, direita e parar. Depois as imagens do conjunto de treinamento foram redimensionadas no formato 224x224 *pixels* em RGB por ser o formato aceito pelo modelo.

Balacear o *dataset* resultou em 53 imagens de cada classe. Por ser um total de 212 imagens foi necessário utilizar as técnicas de *data augmentation* sobre o *dataset* balanceado. As técnicas utilizadas foram o *crop*, ajuste de gama, correção de histograma e rotação horizontal. Resultando num total de 1.060 imagens, sendo elas separadas em uma proporção 0.15 (85% para treinamento e 15%).

O modelo foi treinado com um total de 30 épocas, pois o número de imagens é pouco podendo gerar *overfitting* em um grande número de épocas. A taxa de aprendizado utilizada foi de $1 * 10^{-6}$ e tamanho do *batch* 4, isso aumentou a qualidade dos resultados em troca de elevar o tempo de treinamento da rede. Aplicando o *dataset* a rede InceptionV3 foi possível atingir uma acurácia de 76.73% após as 30 épocas (Figura 1).

Figura 1– a)Acurácias imagens treino b) Loss imagens treino c) Acurácias imagens de validação d) Loss imagens de validação



Fonte: Autoria própria (2019).

CONCLUSÃO

Portanto, dada a execução desse trabalho conclui-se que a criação de um veículo autômato é viável. Uma vez que, com poucas instâncias foi possível obter um resultado acima de 70%. Com possível aumento de acurácia conforme mais imagens forem coletadas.

A coleta balanceada e utilização de técnicas de aumento de dados é essencial para o aprendizado correto da rede. Balancear o conjunto de dados reduz a probabilidade da rede não conseguir classificar corretamente classes com poucas imagens. *Data augmentation* aumenta a quantidade de imagens sem a necessidade de novas capturas, assim reduzindo os efeitos do *overfitting*.

AGRADECIMENTOS

Agradeço a Fundação Araucária por disponibilizar uma bolsa de iniciação científica, e também a meus dois orientadores por estarem disponíveis e presentes durante a execução deste projeto. Só assim este projeto foi possível.

REFERÊNCIAS

BOJARSKI, M.; TESTA, D; DWORAKOWSKI, D.; FIRNER, B.; FLEPP, B.; GOYAL, P.; JACKEL, L.D.; MONFORT, M; MULLER U.; ZHANG, J; ZHANG, X; ZHAO, J; ZIEBA, K. *End to end learning for self-driving cars*. In: eprint arXiv:1604.07316, 2016.

DENG, L. *A tutorial survey of architectures, algorithms, and applications for deep learning*. In: APSIPA Transactions on Signal and Information Processing, 2014.

FRANÇA, M. T.; FRANÇA, M. G. A utilização de visão robótica para captura de dados do ambiente em tempo real com o protótipo de baixo custo. In: [S.l.: s.n.], 2018.

PEREZ, L.; WANG, J.. *The effectiveness of data augmentation in image classification using deep learning*. In: CoRR, abs/1712.04621, 2017.

PISSARDINI, R.S.; WEI, D.C.M.; JÚNIOR, E.F. Veículos Autônomos: conceitos, histórico e estado-da-arte. In. Anais do XXVII Congresso de Pesquisa e Ensino em Transportes-ANPET, 2013.

POCZTER, S. L.; JANKOVIC, L. M. *The Google Car: Driving Toward A Better Future?*. Journal of Business Case Studies (JBCS), v. 10, n. 1, p. 7-14, 31 Dec. 2013.

SZEGEDY, C; VANHOUCHE, V.; LOFFE, S; SHLENS, J.; WOJNA, Z. *Rethinking the Inception Architecture for Computer Vision*. In: IEEE Conference on Computer Vision and Pattern Recognition, 2016.