

DESENVOLVIMENTO DE UM ROBO CAPAZ DE IDENTIFICAR UM TRAJETO DEFINIDO POR UMA LINHA UTILIZANDO TÉCNICAS DE APRENDIZAGEM DE MÁQUINA

Emily Caroline dos Santos Costa ¹; Anderson Harayashiki Moreira ²

¹ Aluna de Iniciação Científica da Escola de Engenharia Mauá (EEM/CEUN-IMT);

² Professor da Escola de Engenharia Mauá (EEM/CEUN-IMT).

Resumo. *Os robôs de serviço vem tomando um campo maior dentro da área da robótica, e dentro dessa classificação pode-se dar enfoque aos robôs do tipo AGV (Automated Guided Vehicle), robôs de transporte amplamente utilizados que precisam ter a capacidade de seguir um trajeto definido por uma linha. A otimização dos algoritmos utilizados para fazer com que o robô siga a linha sofrem grande influência do trajeto a ser percorrido, conhecer tal trajeto previamente pode trazer um benefício para esse tipo de robô, permitindo que os algoritmos do robô sejam atualizados/otimizados. O presente trabalho apresenta a utilização de algoritmos de aprendizagem de máquina para, a partir dos dados provenientes do sensoriamento e controle do robô, a classificação automática do tipo de trajeto a ser percorrido pelo robô dentro um universo de três possibilidades previamente estabelecido. Após a coleta dos dados e treinamento dos modelos de predição utilizando redes neurais convolucionais foi possível chegar em um modelo de classificação automática com 92% de acuracidade indicando que é possível utilizar os dados provenientes dos sensores do robô e um algoritmo de inteligência artificial para identificar o trajeto que está sendo percorrido pelo robô.*

Introdução

A área de robôs de serviço tem se mostrado uma das mais promissoras da robótica, ela é composta por todos os tipos robôs que possuem atuação fora do ambiente industrial, como por exemplo, robôs domésticos, de entretenimento e defesa. Segundo a IFR (International Federation Of Robotics, 2020) a robótica de serviço tem apresentado um crescimento na ordem de 15% nos últimos anos com projeção de crescimento de 25% até o final deste ano de 2020, o que justifica diversas pesquisas nesta área.

Os robôs móveis possuem grande destaque na área de serviços, sendo utilizados em substituição a mão de obra humana em tarefas como recepcionista, guia, acompanhante de pacientes em hospitais (Roy *et al.*, 2000)(Jayawardena *et al.*, 2010). Tais tarefas demandam robôs capazes de realizar uma série de tarefas complexas, como por exemplo, mapeamento e localização, interação com seres humanos, processamento de imagem e voz, porém, existe um grande número de aplicações onde o robô deve ser apenas capaz de seguir um trajeto determinado por uma linha, tais robôs são denominados AGV (*Automated Guided Vehicle*) (Aires, Alsina e Medeiros, 2001).

Para a detecção da linha a ser seguida normalmente duas soluções são adotadas, uma baseada em sistemas de visão computacional como proposto por Dupuis e Parizeau (2006) ou utilizando sensores dedicados, como os sensores infravermelho (IR) propostos por Hogg *et al.* (2002). Já para o controle dos robôs normalmente utiliza-se estruturas condicionais simples quando não se deseja alto desempenho ou quando se possui trajetos simples. Para robôs que possuem uma maior velocidade ou seguem trajetos com mudanças repentinas de direção, normalmente utiliza-se controladores digitais, como por exemplo, controladores Proporcional, Integral e Derivativo ou simplesmente PID (Parikh; Shah; Sheth, 2014).

Sintonizar os controladores PID para se obter um desempenho ótimo pode ser uma tarefa custosa e sem a garantia de sucesso, uma vez que os valores dos parâmetros do controlador estão fortemente relacionados ao trajeto que o robô deverá seguir. O conhecimento prévio do tipo de trajeto a ser seguido pode facilitar o processo de escolha dos parâmetros do controlador, ou até

mesmo permitir que tipos de controladores diferentes sejam selecionados a partir do momento que um determinado trajeto seja identificado.

O reconhecimento de padrões, na situação proposta a identificação do trajeto a ser percorrido pelo robô, é uma das tarefas usualmente atribuídas aos algoritmos de aprendizagem de máquina (Weiss; Kapouleas, 1989) e atualmente devido ao avanço da computação, seja por parte do poder de processamento dos novos *hardwares*, seja pelo avanço nos algoritmos de aprendizagem de máquina, tem se tornado cada vez mais simples implementar tais algoritmos.

O objetivo geral deste trabalho é criar um robô que seja capaz de identificar automaticamente o trajeto que está seguindo dentre um número finito de modelos disponíveis utilizando apenas os dados provenientes de seus sensores e técnicas de aprendizagem de máquina.

Tal objetivo geral poder ser dividido nos seguintes objetivos específicos:

- Construir e programar um robô capaz de seguir um trajeto definido por uma linha;
- Coletar dados provenientes dos sensores do robô para alimentar o algoritmo de aprendizagem de máquina responsável pela criação do modelo de classificação automática;
- Implementar algoritmos de aprendizagem de máquina utilizando diferentes técnicas ou abordagens de classificação;
- Analisar qual técnica de classificação ou abordagem é mais apropriada para a tarefa de identificação do trajeto seguido pelo robô com base em métricas de assertividade do modelo.

Material e Métodos

Para a consecução dos objetivos específicos deste trabalho fez-se necessário os seguintes materiais: robô seguidor de linha e trajetos com circuitos pré-estabelecidos. Primeiramente foi construído um robô capaz de seguir trajetos definidos por linhas, para tal utilizou-se os componentes presentes no kit de desenvolvimento LEGO® MINDSTORMS® EV3 já disponíveis no Instituto Mauá de Tecnologia. Foram utilizados quatro sensores infravermelhos para identificar a linha e dois motores com *encoders* responsáveis por fornecer os dados de movimentação das rodas. O robô foi construído seguindo uma configuração diferencial com três rodas, sendo duas rodas motoras e uma roda de apoio omnidirecional, assim como pode ser observado na Figura 1.



Figura 1 – Robô diferencial seguidor de linha com quatro sensores infravermelho e dois *encoders*.

Em seguida foram definidas três figuras geométricas que definiriam os trajetos a serem seguidos pelo robô, foram escolhidas as formas de um círculo, um quadrado e um círculo. As três figuras foram criadas sobre um lençol de borracha preta utilizando uma fita isolante branca de 19 mm de largura, todas as figuras deveriam possuir o mesmo perímetro, portanto primeiramente, definiu-se arbitrariamente que o raio do círculo deveria ser 40 cm, calculado o perímetro do círculo, definiu-se a medida dos lados do quadrado e do triângulo, que foi feito isósceles e com altura igual ao lado do quadrado. A Figura 2 apresenta os trajetos utilizados neste trabalho.

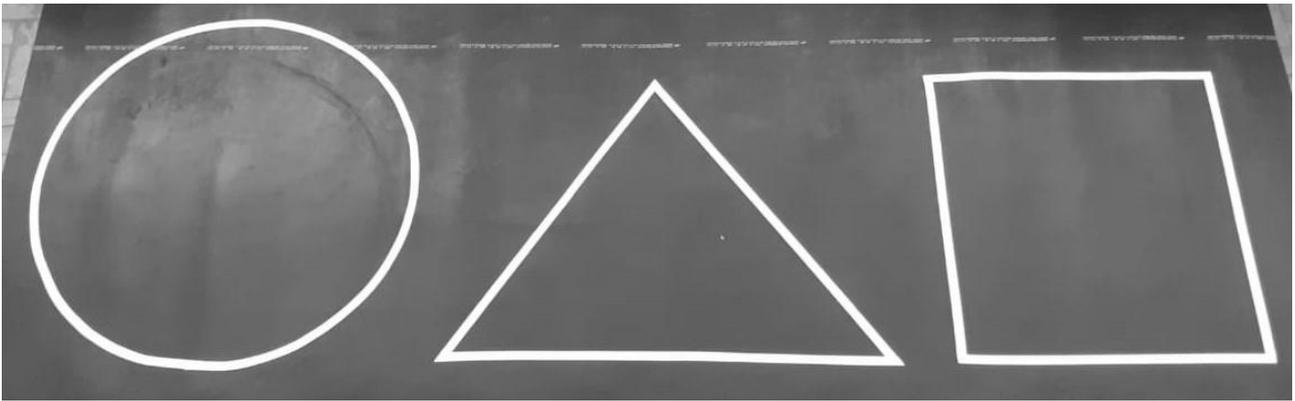


Figura 2 – Trajetos utilizados para a coleta de dados de movimentação do robô.

Com posse do robô e com os trajetos definidos, fez-se necessário criar um programa para o robô que permitisse a ele além de seguir a linha branca, também ser capaz de coletar os dados dos sensores infravermelhos, *encoders*, velocidades comandadas para as rodas e tempo de execução. Optou-se por utilizar a linguagem de programação Python para criar tal programa. O controlador LEGO® MINDSTORMS® EV3 não possui nativamente suporte à linguagem Python, portanto foi necessário configurar um ambiente de desenvolvimento Linux para o LEGO® MINDSTORMS® EV3 chamado EV3dev.

Tal ambiente de desenvolvimento pode ser instalado em um cartão MicroSD sendo possível instalar um interpretador Python bem como um conjunto de bibliotecas responsáveis por ler os dados dos sensores e enviar comandos para os motores. O EV3dev age como um novo sistema operacional para o LEGO® MINDSTORMS® EV3, criando um novo sistema de arquivos e interface de operação e acesso e configuração, que pode ser observado na Figura 3.

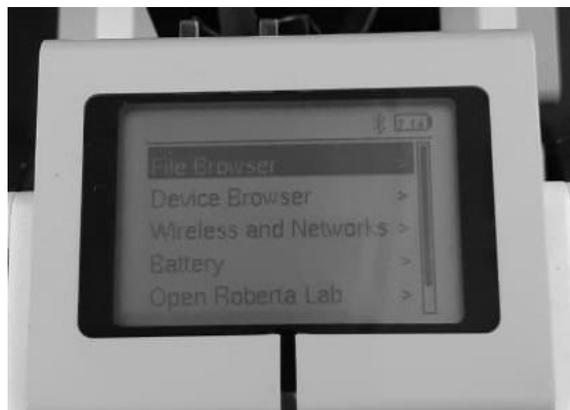


Figura 3 – Interface de operação e acesso do EV3dev.

O robô foi programado para além de seguir a linha, coletar os dados que seriam utilizados para o treinamento do algoritmo de aprendizagem de máquina. Os dados coletados foram:

- *left sensor*: sensor infravermelho da esquerda;
- *left middle sensor*: sensor infravermelho central da esquerda;
- *right middle sensor*: sensor infravermelho central da direita;
- *right sensor*: sensor infravermelho da direita;
- *left motor position*: dados de posicionamento do *encoder* da roda esquerda;
- *right motor position*: dados de posicionamento do *encoder* da roda direita;
- *left motor speed*: velocidade comandada para o motor da roda esquerda;
- *right motor speed*: velocidade comandada para o motor da roda direita;
- *time*: tempo de execução do programa;
- *figure*: valor numérico referente a figura geométrica do circuito (0: círculo, 1: quadrado e 2: triângulo)

Cada conjunto de dados era coletado a cada 100 ms, sendo que cada figura levava aproximadamente 20 s para ser completada o que resultava em aproximadamente 200 conjuntos de dados para cada figura. Esse procedimento de coleta foi repetido 50 vezes para cada figura geométrica, totalizando aproximadamente 30.000 conjuntos de dados. O programa foi implementado em linguagem Python, porém, é possível observar na Figura 4 uma representação da lógica do programa.

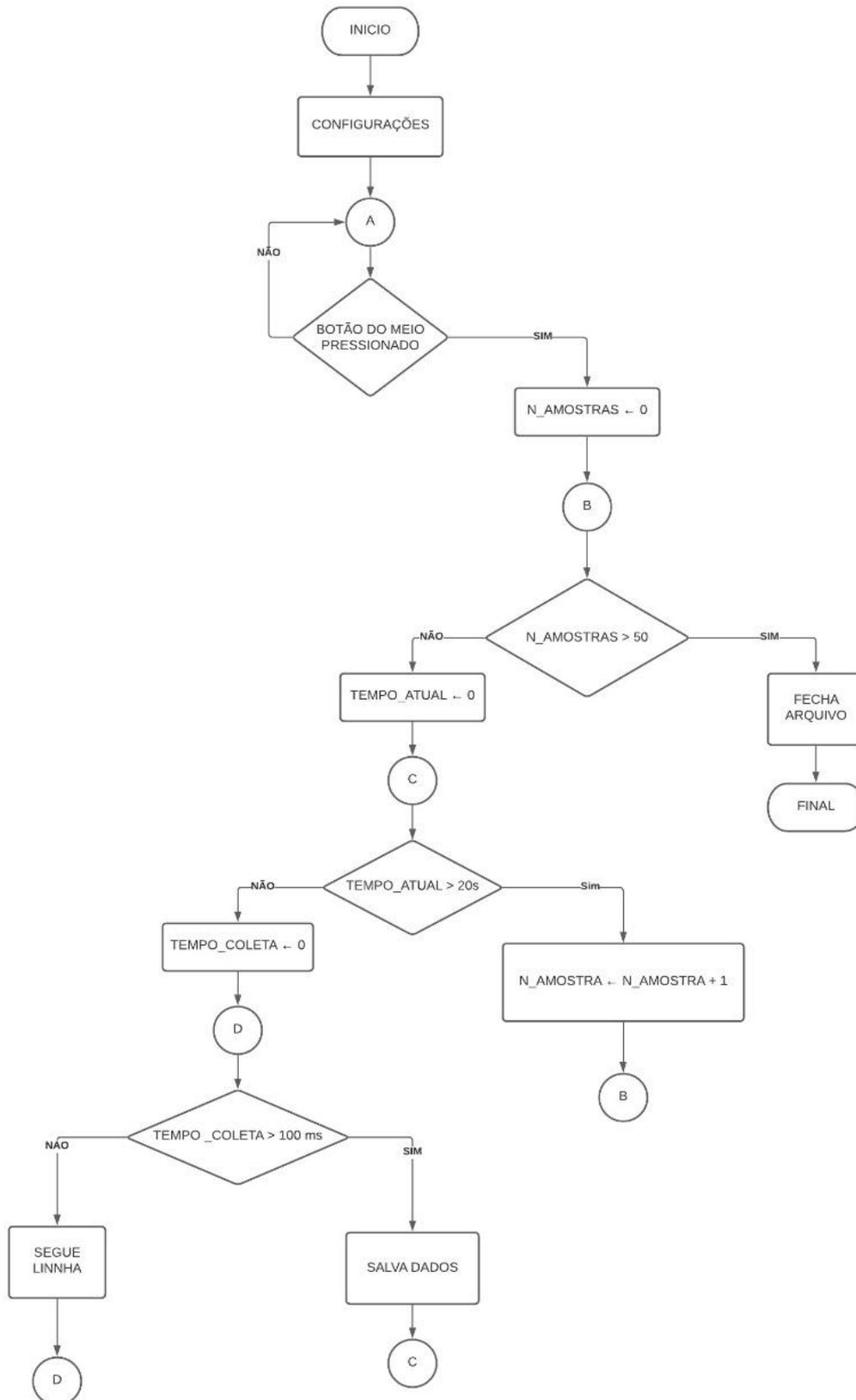


Figura 4 – Fluxograma do programa para controle do robô e coleta de dados.

Os dados coletados foram salvos em arquivos de texto que posteriormente foram carregados em um programa, também desenvolvido em linguagem Python, responsável por implementar o algoritmo de aprendizagem de máquina. Primeiramente os dados foram transformados em uma estrutura de dados chamada de *DataFrame* da biblioteca Pandas, a escolha pela utilização de tal estrutura deve-se a facilidade de se manipular os dados para se fazer as etapas de pré-processamento necessárias para a implementação dos algoritmos de aprendizagem de máquina. Na Tabela 1 é possível os 10 primeiros dados coletados pelo robô já transformados em *DataFrame* do Pandas.

Tabela 1 – Dados coletados pelo robô.

	left sensor	left middle sensor	right middle sensor	right sensor	left motor position	right motor position	left motor speed	right motor speed	time	figure
0	5	6.0	4.0	6.0	56.0	9.0	239.0	238.0	0.16	0
1	7	15.0	6.0	7.0	101.0	52.0	356.0	337.0	0.29	0
2	7	12.0	6.0	6.0	144.0	95.0	280.0	280.0	0.43	0
3	6	5.0	6.0	6.0	190.0	140.0	293.0	300.0	0.58	0
4	7	5.0	6.0	6.0	231.0	181.0	305.0	297.0	0.72	0
5	7	4.0	42.0	8.0	277.0	226.0	297.0	282.0	0.87	0
6	8	4.0	49.0	7.0	314.0	239.0	289.0	36.0	1.00	0
7	7	17.0	6.0	7.0	353.0	258.0	299.0	268.0	1.13	0
8	8	11.0	5.0	6.0	388.0	289.0	161.0	161.0	1.26	0
9	7	8.0	5.0	6.0	418.0	322.0	259.0	261.0	1.40	0

Antes de se iniciar a criação do algoritmo de aprendizagem de máquina decidiu-se plotar um gráfico de dispersão com os valores dos dados provenientes dos *esconders* dos motores de cada uma das rodas, os dados referentes à roda da esquerda estão localizados no eixo das abscissas (eixo x) e os dados referentes à roda da direita estão no eixo das ordenadas (eixo y). Um exemplo de cada uma das figuras pode ser observado na Figura 5.

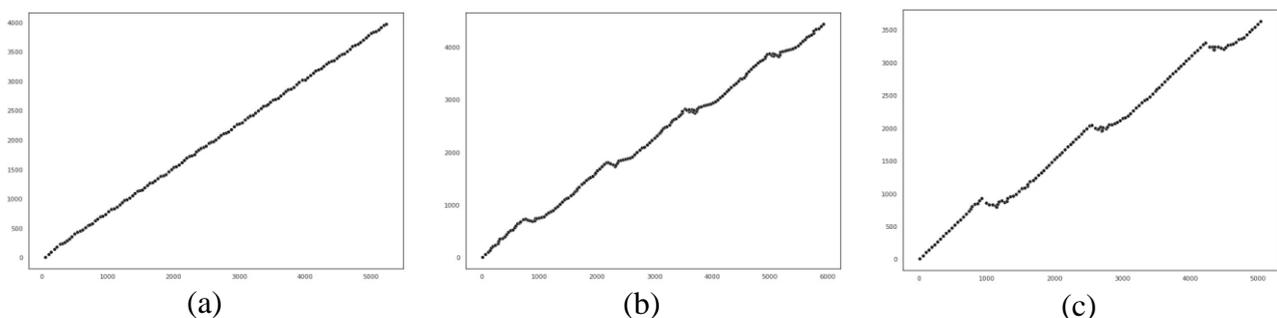


Figura 5 – Gráficos de dispersão dos dados dos *encoders* das rodas (a) gráfico referente ao círculo (b) gráfico referente ao quadrado (c) gráfico referente ao triângulo.

Nas Figuras 5(b) e 5(c) é possível observar patamares, ou seja, trechos onde os valores de deslocamento da roda esquerda aumentam significativamente sem que sejam acompanhados pelos valores da roda da direita, tal acontecimento pode ser notado quatro vezes na Figura 5(b) e três vezes na Figura 5(c), sabendo que o robô realizou os trajetos no sentido horário, esses gráficos apresentam indícios que os dados coletados podiam ser utilizados para fazer o reconhecimento dos padrões desejados. Uma vez que quatro patamares correspondem aos quatro vértices do quadrado, os três patamares correspondem aos três vértices do triângulo e ausência de patamares corresponde ao círculo.

Após a criação dos gráficos apresentados acima, deu-se início a criação do algoritmo de aprendizagem de máquina, optou-se por utilizar as técnicas de *Deep Learning* ou Aprendizagem

Profunda para se tentar classificar automaticamente os trajetos executados pelo robô. Tal escolha deve-se ao fato dos atuais algoritmos de aprendizagem profunda apresentarem os melhores resultados em tarefas similares à proposta por esse trabalho.

Neste trabalho optou-se por utilizar duas abordagens distintas, porém, ambas utilizando um algoritmo baseado em redes neurais convolucionais. A primeira abordagem todos os dados coletados foram utilizados de maneira independente, ou seja, todos os quase 30.000 conjuntos de dados foram utilizados como se fossem dados isolados, cada conjunto de dado era uma instância ou evento, e todas as nove características (*left sensor*, *left middle sensor*, *right middle sensor*, *right sensor*, *left motor position*, *right motor position*, *left motor speed*, *right motor speed* e *time*) foram utilizadas como entrada do modelo.

Nesta primeira abordagem, a variável de saída foi transformada em um *array* esparsa de dimensão três, sendo que cada uma das colunas correspondia ao um tipo de figura, a primeira coluna equivalia ao círculo, a segunda ao quadrado e a terceira ao triângulo. Tal ação tinha por objetivo obter um melhor resultado da rede neural, que por sua vez possuía três camadas com 32 neurônios na camada escondida, com função de ativação RELU e três neurônios na camada de saída.

A segunda abordagem utilizou apenas os dados dos *encoders* das rodas, porém, os dados foram rearranjados em conjuntos de dados que representavam o trajeto completo de uma figura. Ao invés de se utilizar os dados absolutos de posicionamento, optou-se por utilizar o deslocamento relativo entre cada uma das amostras. Cada conjunto de dados foi transformado em um *array* único com 150 dados para a roda da esquerda e 150 dados para a roda da direita, os 300 dados foram utilizados simultaneamente como entrada de uma rede neural também de 3 camadas, porém com 64 neurônios na primeira camada escondida e 32 na segunda camada, ambas as camadas também utilizaram função de ativação RELU. A camada de saída também possuía 3 neurônios, uma vez que os dados de saída possuíam o mesmo formato da abordagem anterior.

Nas duas abordagens os dados foram divididos em dados de treinamento e dados de validação. Dois terços dos dados foram utilizados para treinamento e um terço ficou reservado para a validação dos modelos obtidos. Os resultados de cada um dos modelos obtidos podem ser observados na seção de Resultados e Discussão.

Resultados e Discussão

Para a primeira abordagem utilizou-se 500 épocas para o treinamento, porém, foi adicionado um critério de parada antecipada (*EarlyStopping*) caso o valor da função custo dos dados de validação permanecessem inalterados por mais de 30 épocas (paciência). Na Figura 6 é possível observar a evolução do treinamento do modelo de classificação da primeira abordagem.

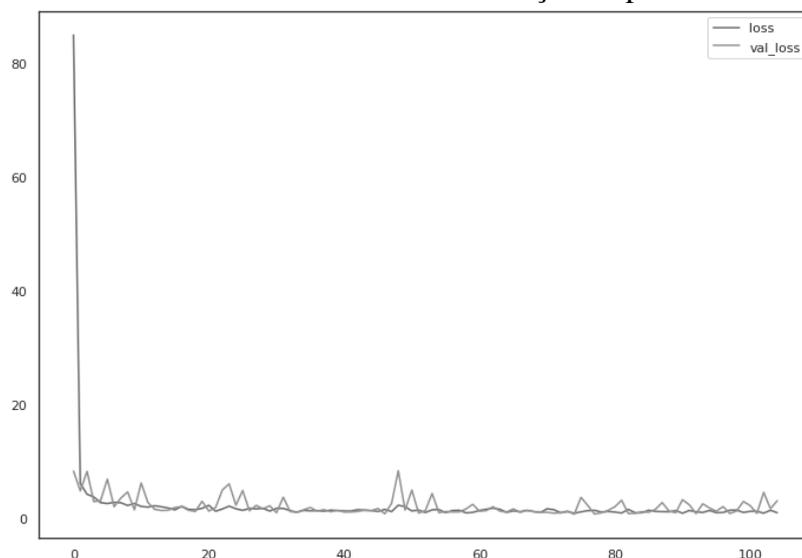


Figura 6 – Gráfico da evolução do treinamento do modelo de classificação da primeira abordagem.

O menor valor da função custo dos dados de validação obtido para esse modelo foi de 0.6860 com um valor máximo de acuracidade de 0,7654, apesar de não ser um valor baixo, dado a simplicidade dos dados utilizados para a predição perante a complexidade da variável de saída desejada, ao se realizar a predição utilizando-se os dados de validação chegou-se na seguinte matriz de confusão apresentada na Figura 7.

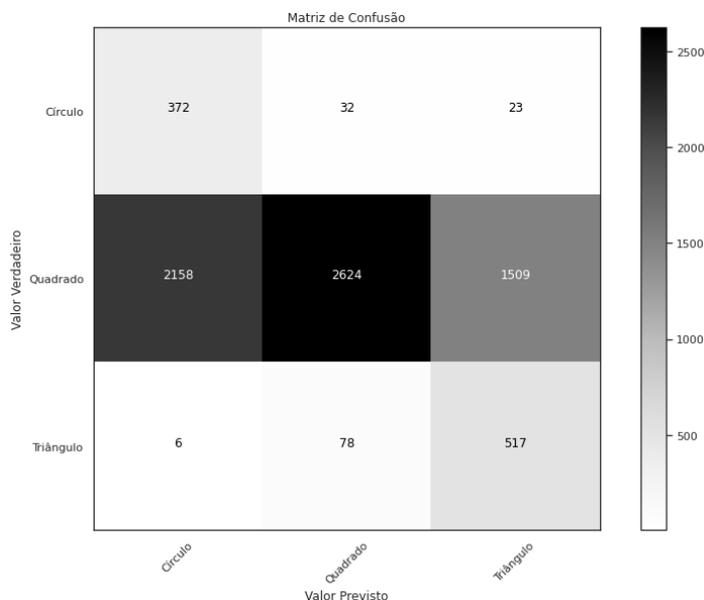


Figura 7 – Matriz de confusão referente aos dados de validação da primeira abordagem.

Pode-se observar que uma quantidade significativa de quadrados é classificada como círculos (2158) ou triângulos (1509). Tal resultado motivou a investigar uma segunda abordagem, onde os dados não foram tratados de maneira isolada, mas sim como figuras completas, isso reduziria o número de amostras para uma quantidade significativamente menor, de aproximadamente 30.000 para 150, porém, a qualidade da informação proveniente dos dados de entradas seria maior.

O treinamento do modelo da segunda abordagem foi estipulado com um máximo de 1500 épocas, porém, também foi definido uma parada antecipada com paciência de 500 épocas. A evolução do treinamento do modelo de classificação da segunda abordagem pode ser observada na Figura 7.

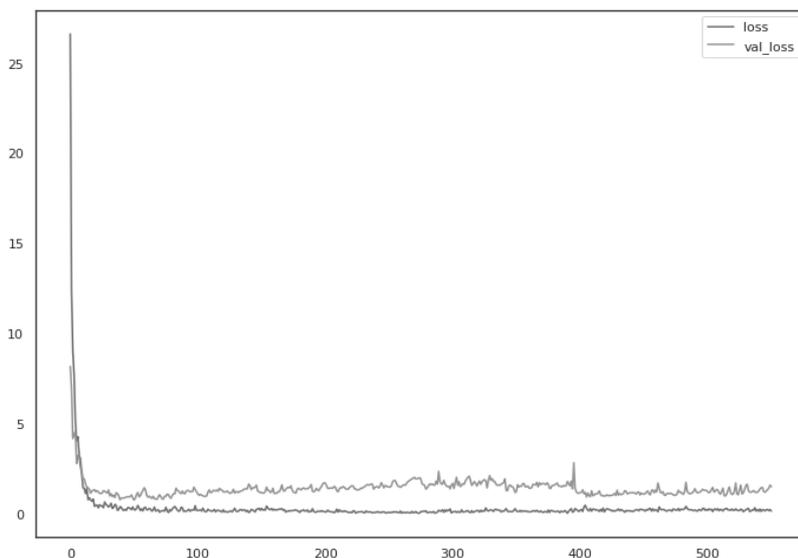
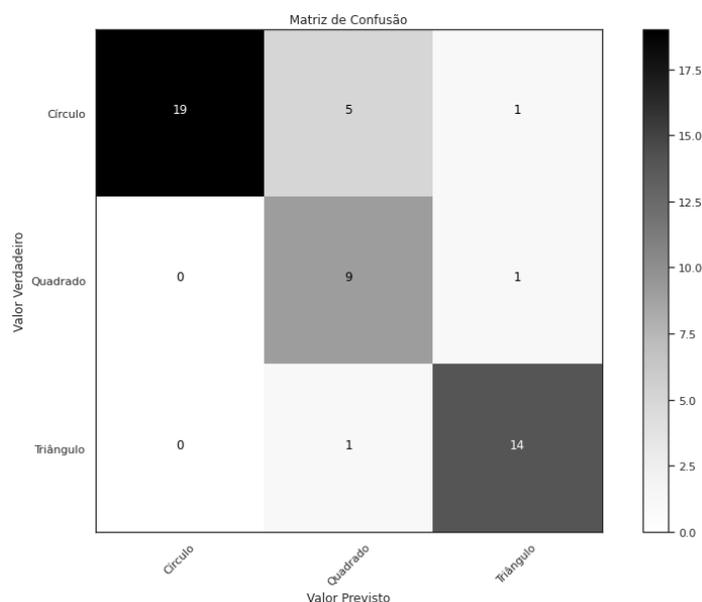


Figura 7 – Gráfico da evolução do treinamento do modelo de classificação da segunda abordagem.

Para a segunda abordagem o menor valor da função custo dos dados de validação obtido foi de 0.7818 com um valor máximo de acuracidade de 0,9200, um aumento expressivo na assertividade do modelo encontrado, tal aumento é facilmente observado ao se plotar a matriz de confusão do modelo obtido (Figura 8) predizendo os mesmo dados de validação.



Apesar de ainda apresentar algumas inconsistências, a segunda abordagem resultou em um modelo com um desempenho significativamente superior, 92,00% de acuracidade ante os 76,54% do modelo da primeira abordagem.

Conclusões

Este trabalho apresenta o desenvolvimento de um robô que seja capaz de identificar automaticamente o trajeto que está seguindo dentre um número finito de modelos disponíveis utilizando apenas os dados provenientes de seus sensores e técnicas de aprendizagem de máquina.

Os objetivos específicos declarados na Introdução deste trabalho foram atendidos. Foi construído e programado um robô capaz de seguir um trajeto definido por uma linha branca sobre uma superfície preta. Tal robô foi capaz de coletar dados provenientes de seus sensores para alimentar o algoritmo de aprendizagem de máquina responsável pela criação do modelo de classificação automática.

Foi implementado um algoritmos de aprendizagem de máquina utilizando técnicas de Aprendizagem Profunda (*Deep Learning*) e foram propostas duas abordagens distintas para a resolução do problema proposto de classificação automática dos trajetos percorridos pelo robô. As duas abordagens utilizaram redes neurais convolucionais e apresentaram resultado satisfatório.

Utilizando o valor da acuracidade dos modelos obtidos ao predizer os dados de validação coletados, foi possível obter uma acuracidade de 76,54% para a primeira abordagem que utilizou os dados de forma independente e uma acuracidade de 92,00% ao se utilizar os dados agrupados por trajeto completo de cada uma das figuras geométricas.

A utilização de técnicas de aprendizagem de máquina mostrou-se eficiente para a classificação automática dos trajetos percorridos pelo robô, porém, devido a uma limitação de capacidade do ambiente de desenvolvimento do robô não foi possível embarcar o modelo de classificação automática para ser executado em tempo real no robô, outras possibilidades de implementação estão sendo estudadas, porém, estão fora do escopo e período deste trabalho.

Referências Bibliográficas

- AIRES, K. R. T., ALSINA, P. J., MEDEIROS, A. A. D. (2001) A Global Vision System for Mobile Mini-Robots. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 5, Canela, 2001. Anais. Canela, Sociedade Brasileira de Automática.
- BIANCHI, R. A. C., SIMÕES, A. S., REALI-COSTA, A. H. (2001). Comportamentos Reativos para Seguir Pistas em um Robô Móvel Guiado por Visão. In: SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, 5, Canela, 2001. Anais. Canela, Sociedade Brasileira de Automática.
- DUPUIS, J.; PARIZEAU, M.. Evolving a Vision-Based Line-Following Robot Controller. In: CANADIAN CONFERENCE ON COMPUTER AND ROBOT VISION, 3., 2006, Quebec.Proceedings... . Quebec: Ieee, 2006. p. 75 - 75.
- HOGG, R.W. et al. Algorithms and sensors for small robot path following. Robotics And Automation, Pasadena, n. , p.3850-3857, 07 ago. 2002.
- INTERNATIONAL FEDERATION OF ROBOTICS. Why service robots are booming worldwide. 2020. Disponível em: <<https://ifr.org/news/why-service-robots-are-booming-worldwide/>>. Acesso em: 05 fev. 2020.
- JAYAWARDENA, Chandimal et al. Deployment of a service robot to help older people. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2010. p. 5990-5995.
- PARIKH, P.; SHAH, H.; SHETH, Saurin. A Mechatronics design of a line tracker robot using Ziegler Nichols control technique for P, PI and PID controllers. In: International Mechanical Engineering Congress (IMEC-2014). 2014. p. 13-15.
- ROY, Nicholas et al. Towards personal service robots for the elderly. In: Workshop on Interactive Robots and Entertainment (WIRE 2000). 2000. p. 184.
- WARD, Nigel. EV3dev Python. 2018. Disponível em: <https://sites.google.com/site/ev3devpython/>. Acesso em: 05 maio 2020.
- Weiss, S.; Kapouleas, I.. “An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods.” IJCAI (1989).