



Engenharia da Computação

Artigo Científico

Matheus da Silva Rego Gomes

Jessica Maria Ferreira Borges

Rio de Janeiro

2021

Instituto INFNET

ESTI – Escola Superior da Tecnologia da Informação

Engenharia da Computação

Artigo Científico

Matheus da Silva Rego Gomes

Jessica Maria Ferreira Borges

Rio de Janeiro

2021

Drone Arara

O Projeto Drone Arara foi desenvolvido com o intuito de resolver a problemática relacionada às emboscadas contra policiais, em razão de uma infraestrutura desordenada e precária das ruas, casas e vielas em comunidades carentes, visto que, os agentes da lei correm risco de serem encurralados por criminosos. Utilizando reconhecimento de gestos com a finalidade de distinguir sinais específicos das mãos, em que os policiais usam, o drone executa movimentos ou sobrevoa os com a intenção de mapear e acabar com pontos cegos ao seu redor. No que se refere às tecnologias usadas a fim de desenvolver o projeto foram usadas a biblioteca OpenCV, o Framework ROS Noetic com o simulador Gazebo 11.

I. INTRODUÇÃO

Em operações policiais podem ocorrer erros humanos e emboscadas, os números de policiais mortos em operações tende a vir com o número de vítimas feitas pelos mesmos. Essas altas quantidades de mortes e precariedade sofrida pela polícia vem sendo demonstrada em notícias, desde 2017 até atualmente (2021), de revistas virtuais, tais como G1 [1], UOL [2], Radio Agência Nacional [3], O Globo [4] e GZH Segurança [5].

A fim de trazer mais segurança aos policiais e moradores das comunidades o projeto Drone Arara foi pensado como uma solução com a finalidade de explorar e mapear áreas que são pontos cegos para policiais, como por exemplo: esquinas, vielas e visão aérea. Reconhecendo sinais do alfabeto de libras o drone realizará varreduras no entorno ou no alto, assim, podendo reduzir erros policiais e emboscadas contra os mesmos.

O projeto desenvolvido foi pensado e testado em ambiente ideal, simulado no Gazebo, já que não houve a implementação do projeto em um drone verdadeiro. Outra decisão foi não realizar a comunicação direta entre Inteligência Artificial e o Drone Arara simulado no Gazebo.

II. TRABALHOS RELACIONADOS

Com o objetivo de desenvolver a simulação do drone foram usadas as ferramentas OpenCV, o Framework ROS Noetic e o simulador Gazebo 11:

A. Inteligência Artificial – OpenCV

O OpenCV foi projetado para eficiência computacional e com um forte foco em aplicativos de tempo real. Ele é escrito em C++ otimizado e pode tirar proveito de processadores multicore. Utilizaremos o OpenCV por conta do baixo custo computacional.

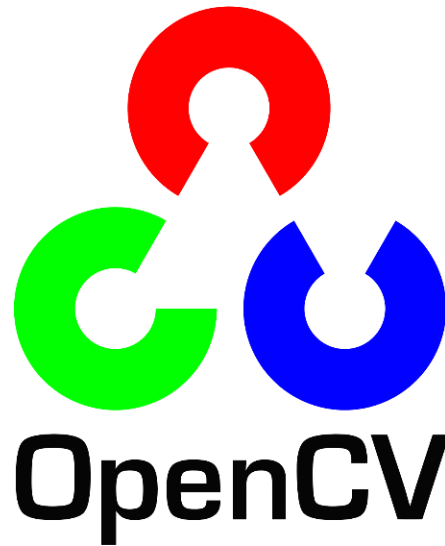


Fig 1. Logo, OpenCV.

O projeto base é a Rede Neurais + Alfabeto Libras [6] uma Rede Neural Convolucional capaz de identificar gestos do alfabeto de Libras, foi utilizado uma biblioteca OpenCV e modelo CNN para identificação real-time o projeto demonstra uma acurácia de aproximadamente 97%. As funções de ativação mais comumente utilizadas em uma MLP são a ReLU (Unidade Linear Retificada), Leaky ReLU e Tanh (Tangente hiperbólica). Cada função possui uma particularidade para otimizar o treinamento, podendo desativar neurônios que podem não ser influentes para o aprendizado dos neurônios da camada seguinte.

Essas funções de ativação devem ser utilizadas nas camadas ocultas de treinamento da Rede. A camada de Pooling serve para simplificar a informação e reduzir a dimensionalidade da convolução. Essa redução é importante por questão de agilidade no treinamento.



Fig 2. Em libras, Letra B.



Fig 3. Reconhecida, Letra B.



Fig 4. Libras, Alfabeto.

Convolutional Neural Network — CNN

As CNN estão incluídas na área de visão computacional por serem utilizadas para o reconhecimento de padrões e classificações de imagens. Para analisar uma imagem uma CNN trabalha com os chamados Campos Receptivos, que tem como base o funcionamento do córtex visual biológico. A ideia é poder analisar a imagem em pequenos pedaços processando pixels em neurônios diferentes sabendo manter a proximidade espacial dos pixels para capturar traços e detalhes importantes. Podemos dividir a estrutura da CNN em dois blocos, o primeiro é o bloco de extração de características onde existe as camadas de convolução e camadas de Pooling o segundo é o bloco de classificação onde possui as camadas totalmente conectadas que se comportam igual a uma MLP, onde os neurônios têm

conexões completas com todos os neurônios ativados da camada anterior.

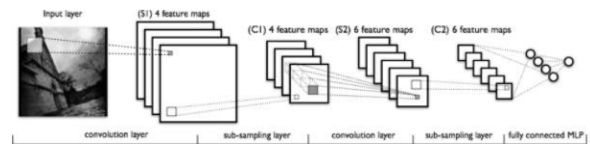


Fig 5. Network, Convolutional Neural.

B. ROS - Gazebo

O ROS ou Robotic Operating System é um framework para desenvolvimento de robôs que envia informações orientado a nós. Junto com o simulador Gazebo 11 pode realizar testes sem a necessidade de realizar a montagem do hardware. A versão Noetic foi usada no Framework ROS, por ser a mais atual e possuir projetos prontos que auxiliaram no desenvolvimento do Drone Arara.



Fig 6. Logo, ROS Noetic.

Para simular o drone no Gazebo foram utilizados artigos e vídeos que auxiliaram no desenvolvimento do código, tópicos e simulação do Drone Arara tais como:

O artigo do site The Construct [7]

Este artigo do site The Construct mostra como gerar um Parrot Ardrone, publicar mensagens em tópicos, usar tele operação por teclado e simular no Gazebo.

O vídeo do YouTube [8]

O vídeo também é relacionado ao artigo do site The Construct para gerar o Parrot Ardrone, porém ensina a criar um arquivo '.sh' que impede que ocorra erro ao executar o comando para abrir o mundo no Gazebo.

WikiROS Noetic [9]

Neste site ensina a realizar a instalação do Framework ROS e o Gazebo, além de possuir tutoriais para movimentação de robôs.



Fig 7. Logo, Gazebo

III. MATERIAIS E MÉTODOS

A. Reconhecimento de Gestos

Vale ressaltar que o projeto atua a partir da classificação de três letras, A, C e L onde cada uma está relacionada ao movimento do drone para tal ação foram utilizadas as bibliotecas OpenCV juntamente com o modelo CNN. Desse modo, foi possível alcançar o reconhecimento em tempo real.

Utilizamos o modelo h5 e a linguagem Python para identificação dos gestos. Foi criado um def predictor para chamar o modelo e retornar as letras as quais foram classificadas, em seguida foi criado um while True para fazer a configuração do dos frames, foi preciso executar os gestos várias vezes para CNN reconhecer a letras, por mais que o modelo seja treinado, ele não estava executando a princípio com exatidão.

Caso a inteligência artificial e o drone fossem integrados pelo tópico '/writing' a CNN publicaria o caractere no tópico ao oposto que acontece com os testes no teleop, que são enviadas informações a partir do teclado.

Texto estruturado CNN:

```
Enquanto a inteligência artificial estiver
executando, verificar

Se for reconhecido um gesto em libras
    mostrar a letra na tela do result

Se não for reconhecida nenhuma letra
    continuar verificando
```

B. Simulação do Drone

Usando os artigos e vídeo mencionados na seção “Trabalhos Relacionados” (ROS-Noetic e Gazebo 11) usei partes do projeto Parrot Ardrone. A estrutura do drone ficou na pasta macaw_drone que possui parte do código de Parrot Ardrone, sjtu-drone para criar o drone quadrotor e a pasta spawn-robot-tools que gera as ferramentas do robô.

Desta forma foi possível gerar o mundo para teste e o drone quadrotor com uma câmera frontal e uma câmera apontada para baixo.

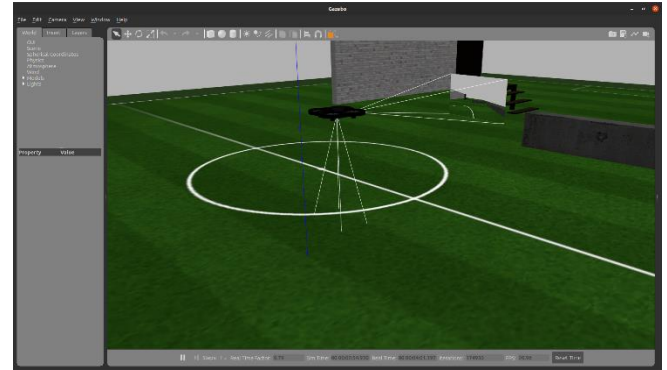


Fig 8. Voando, Drone.

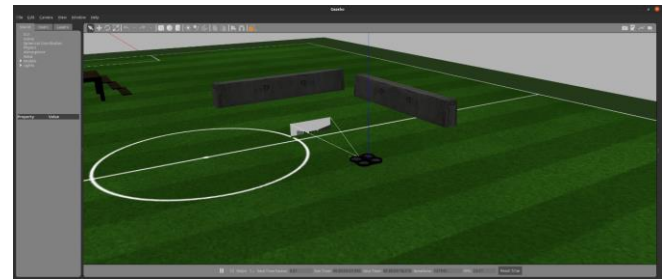


Fig 9. Parado, Drone.



Fig 10. Drone Quadrotor.

A fim de realizar a movimentação do drone criei um código em Python 3 que usa a regra da mão direita, já que, a mesma usa 3 eixos. Usando os eixos x, y e z, regra da mão direita, foi possível mover e rotacionar o drone para a esquerda, direita, frente, ré e alçar voo com o drone a partir do nó '/drone/takeoff'.

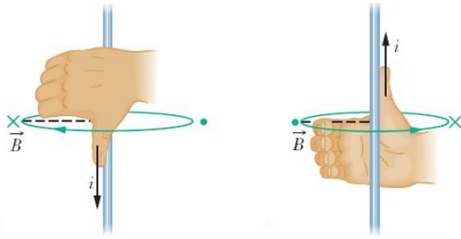


Fig 11. Rotacionar, Regra da Mão Direita.

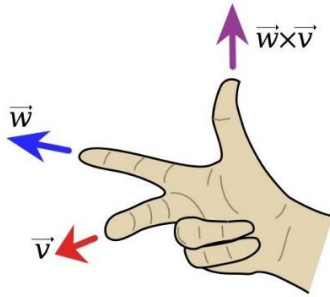


Fig 12. Movimentar, Regra da Mão Direita.

O desenvolvimento da simulação do drone foi necessário criar um Publisher e um Subscriber que não realizam só os movimentos do drone, mas também a interação da Inteligência Artificial com o drone, já que, os mesmos não foram integrados.

Com a finalidade de realizar a comunicação entre os códigos do Publisher e do Subscriber houve a necessidade de usar o nó `/writing` que publica a letra, variável do tipo String, selecionada no terminal e envia para o Subscriber. O código Publisher usa as teclas para:

- L – Alçar voo do drone.
- G – Pousa o drone.
- W – Mover para frente.
- S – Mover para trás.
- A – Girar para a esquerda.
- D – Girar para a direita.

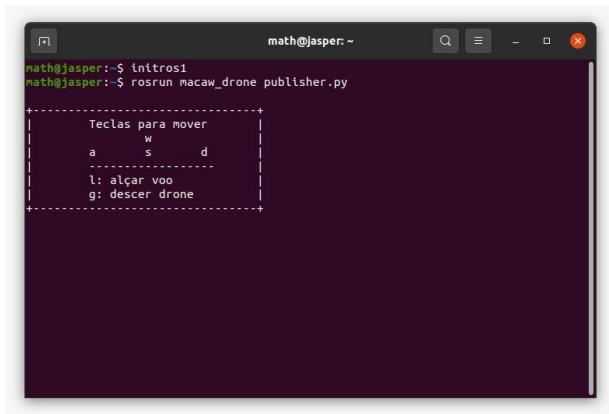


Fig 13. Publisher, Terminal.

A partir do código Subscriber, recebe a letra selecionada no nó `/writing`, traduz o tipo de mensagem `std_msgs/String`, determina em qual método deverá executar e realiza a publicação em um tópico determinado. Tais como:

- G – Publica uma mensagem vazia (Empty) no tópico `/drone/land`.
- L – Publica uma mensagem vazia (Empty) no tópico `/drone/takeoff`.
- W – Publica a tupla (1, 0, 0, 0) no canal `/cmd_vel`.
- S – Publica a tupla (0, 0, 0, 1) no canal `/cmd_vel`.
- A – Publica a tupla (1, 0, 0, 1) no canal `/cmd_vel`.
- D – Publica a tupla (1, 0, 0, -1) no canal `/cmd_vel`.

Texto estruturado Publisher:

Enquanto o código Publisher está executando

Imprime a tabela de movimentos

Entra num loop de verificação

Se um dos botões definidos (l, g, w, s, a, d) for selecionado = publicar a letra no tópico `/writing`

Se os botões CTRL + C forem usados = parar loop

Texto estruturado Subscriber:

Enquanto o código Subscriber está executando

Entra no loop de verificação do tópico `cmd_vel`

Se for igual a letra 'g' = executar método para pousar o drone (Publica no tópico `/drone/land`)

Se for igual a letra 'l' = executar método para subir o drone (Publica no tópico `/drone/takeoff`)

Se não for igual a letra 'g' ou 'l', verificar

Se a letra inserida está na lista de letras (w, s, a, d) = inserir os números para os movimentos e publicar no canal `/cmd_vel`

Se os botões CTRL + C forem usados = parar loop

Desta forma os nós de comunicação foram gerados a partir do `rqt_graph`:



Fig 14. RQT Graph.

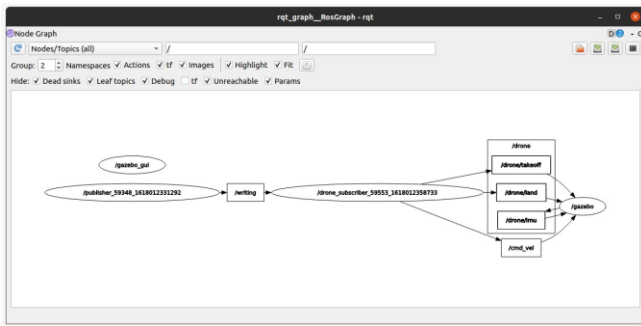


Fig 15. RQT Graph, Janela.

IV. RESULTADOS OBTIDOS

A. Testes dos Gestos

Conforme mostrado a imagem abaixo, podemos verificar no canto superior do lado esquerdo a letra alfabética que representa o sinal de libras.

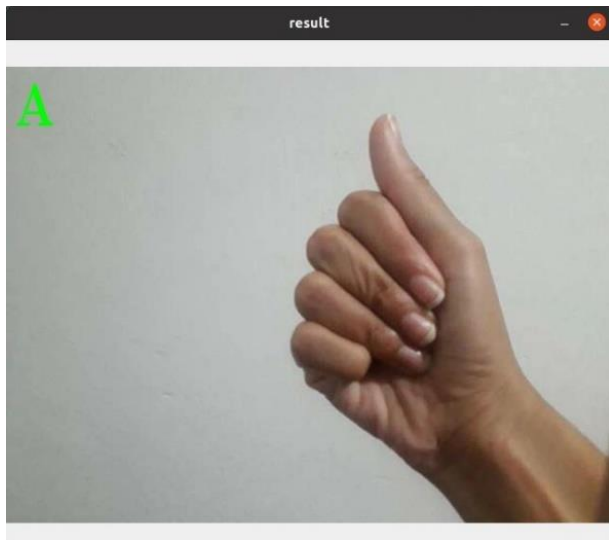


Fig 16. Letra A, Teste.

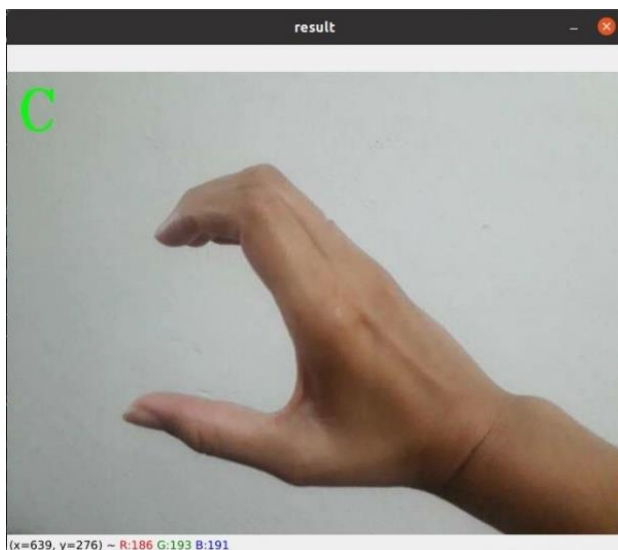


Fig 17. Letra C, Teste.

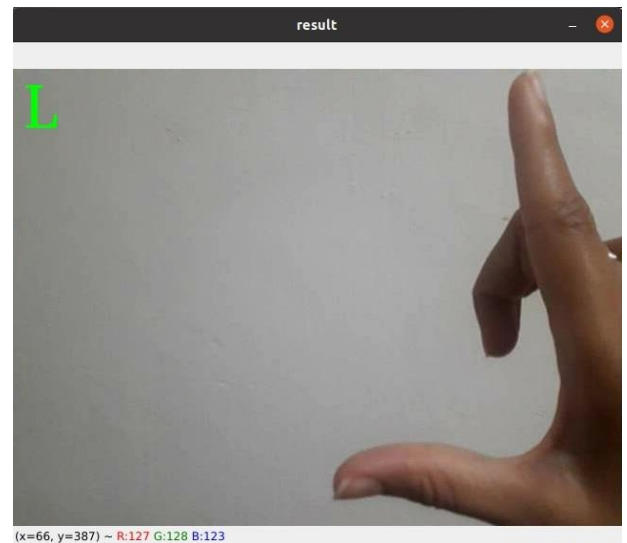


Fig 18. Letra L, Teste.

B. Testes do Drone

A letra 'l' publica uma mensagem vazia (Empty) no tópico '/drone/takeoff'.

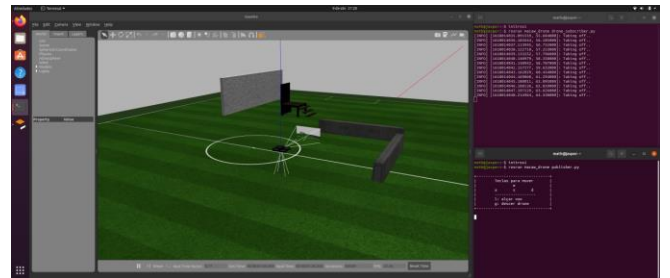


Fig 19. Subir drone, Teste.

A letra 'g' publica uma mensagem vazia (Empty) no tópico '/drone/land'.

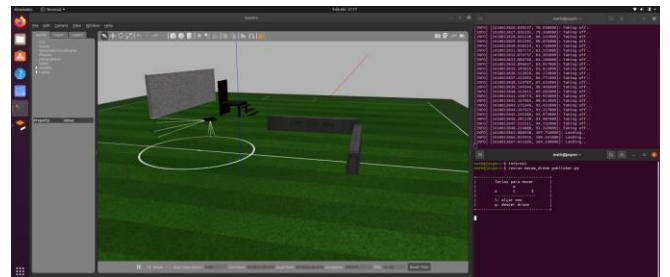


Fig 20. Pousar drone, Teste.

As letras ‘w’, ‘s’, ‘a’ e ‘d’ publicam tuplas específicas na movimentação no tópico ‘cmd_vel’.

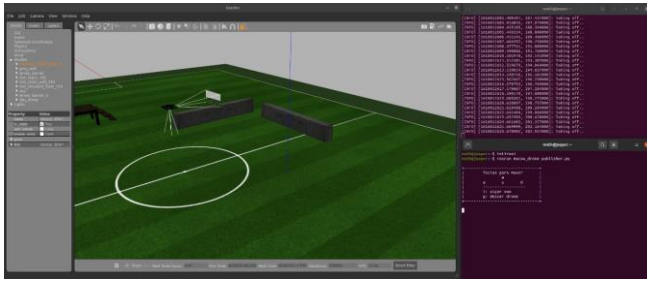


Fig 21. Movendo drone, Teste.

V. CONCLUSÃO

Considerando que a utilização e tomada de decisão do drone é realizada a partir de reconhecimento de sinais de mãos, não foi possível realizar a conexão da CNN com o drone, por conta disso, se fez necessário a criação de cada sistema de forma separada.

O resultado da CNN classifica as letras do alfabeto representada por gestos de Libras. Alguns problemas foram apresentados erros durante a conexão da CNN com o ROS, contudo, posteriormente será inserido uma conexão entre os sistemas.

Podemos verificar realização dos testes da simulação do drone foi necessário modificar o pacote baseado no projeto Parrot Ardrone, para que assim fosse possível gerar a comunicação entre o Publisher e o Subscriber. A simulação mostrou que o drone consegue realizar movimentos, tais como, decolar, pousar, girar e movimentar-se para diferentes direções. Entretanto necessita de melhorias em sua movimentação e que seu pouso ainda é muito abrupto.

Demonstrada as informações do artigo, é perceptível que o drone e a Inteligência Artificial necessitam de melhorias, além de integrar o reconhecimento de gestos com o drone a fim de ser usada em operações policiais reais assim como serem realizados testes reais para determinar sua precisão.

VI. REFERÊNCIAS

[1] “Número de policiais mortos no RJ é quase 4 vezes maior que em SP no primeiro semestre de 2017” [Online]

Acessível:

<https://g1.globo.com/rio-de-janeiro/noticia/numero-de-policiais-mortos-no-rj-e-quase-4-vezes-maior-que-em-sp-no-primeiro-semester-de-2017.ghtml>

[2] “Após 2017 com 134 assassinatos, PM é baleado no Rio; por que essa é a 1ª de muitas mortes” [Online]

Acessível:

<https://noticias.uol.com.br/cotidiano/ultimas-noticias/2018/01/03/rj-termina-2017-com-134-pms-mortos-por-que-esse-numero-nao-deve-cair-em-2018.htm>

[3] “PM morre em emboscada no Rio de Janeiro; 83 policiais tiveram morte violenta este ano” [Online].

Acessível:

<https://agenciabrasil.ebc.com.br/radioagencia-nacional/geral/audio/2018-10/pm-morre-em-emboscada-no-rio-de-janeiro-83-policiais>

[4] “11 sinais da precariedade sofrida pela PM no Rio” [Online].

Acessível:

<https://oglobo.globo.com/rio/confira-11-sinais-da-precariedade-sofrida-pela-pm-do-rio-22184900>

[5] “Número de PMs mortos em serviço em 2019 é o maior dos últimos 5 anos no RS” [Online]

Acessível:

<https://gauchazh.clicrbs.com.br/seguranca/noticia/2019/07/numero-de-pms-mortos-em-servico-em-2019-e-o-maior-dos-ultimos-cinco-anos-no-rs-cjxxcrjr00dc01rv42fvpodd.html>

[6] “cnn-libras” [GitHub]

Acessível:

<https://github.com/lucaaslb/cnn-libras>

[7] “How to Launch the Parrot Drone Simulation Locally” [Online]

Acessível:

<https://www.theconstructsim.com/how-to-launch-drone-simulation-locally/>

[8] “How to Launch the Parrot Drone Simulation Locally” [Online – Vídeo]

Acessível:

<https://www.youtube.com/watch?v=dwdVwwngMow>

[9] “ROS Noetic Ninjemys”

Acessível:

<http://wiki.ros.org/noetic>

[Projeto – GitHub] [infnet_project-drone](https://github.com/infnet-project-drone)

Acessível:

https://github.com/Math09/infnet_project-drone