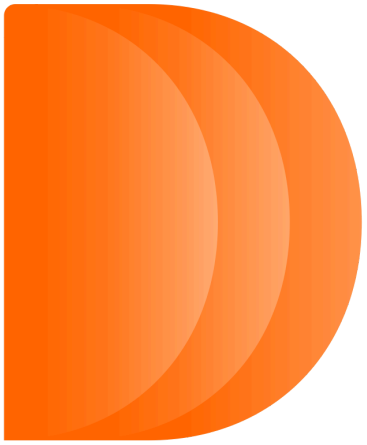

dtLabs

Tecnologias que avançam a humanidade

AVALIAÇÃO PRÁTICA

RECADO AO CANDIDATO

Toda a equipe da DAEDALUS gostaria de te parabenizar pela conquista de ter chegado até aqui! Acreditamos que, independentemente do resultado desta avaliação, possamos manter contato para uma possível futura oportunidade de trabalho em conjunto.

De todos os nossos membros,

Uma boa prova!

INSTRUÇÕES DA AVALIAÇÃO

Este exame tem como intuito avaliar suas competências referentes ao desenvolvimento de software. Além disso, iremos avaliar suas habilidades referentes aos assuntos de Machine Learning, Deep Learning e Visão Computacional.

Todas as questões respondidas devem ser enviadas para o seguinte e-mail:
jobs@dt-labs.ai

PRAZO PARA ENTREGA

Após receber este documento, você possui o prazo de 3 (três) dias para resolver as questões propostas na próxima página.

OBSERVAÇÕES

- **Todas as questões devem** ser resolvidas usando a ferramenta Colab do Google.
- O Colab possui GPU gratuito para uso. Verifique aqui como usar.
- **Você possui duas opções de envio das suas soluções.** A **primeira** é através do link do seu Colab. A **segunda** forma é baixando o arquivo do Colab como um Jupyter Notebook (formato .ipynb). Ambas as formas são aceitas. Caso você queira enviar o link do seu Colab, **não esqueça** de deixar no modo de compartilhamento "Todos com o link".
- Caso você prefira desenvolver o código em seu computador local, não há problema. Entretanto, **é necessário que a solução esteja no Colab, como mencionado anteriormente.**
- **Todas as questões devem** ter o output das células visíveis. Verifique isso antes de enviar a sua solução.
- **Não serão aceitas** questões em formato .py.
- **Não serão aceitas** questões sem o output das células do Jupyter Notebook.
- **Todas as questões devem ser feitas em Python**.
- Você será desclassificado do processo seletivo se incluir qualquer uma das soluções e perguntas em qualquer rede social, GitHub, BitBucket e plataformas similares.
- Tempo de resolução será levado em consideração na avaliação.
- Ao final, você **deve** gravar um vídeo explicando o algoritmo que você desenvolveu. Você será avaliado pela explicação também.
- Você será avaliado(a) pela qualidade do código escrito.
- Você será avaliado(a) pelo desenvolvimento dos algoritmos.

QUESTÃO 1

Uma tarefa que lidamos comumente é trabalhar com nuvens de pontos (point clouds). As point clouds são dados que podem ser extraídos de formas diferentes: um par de câmeras (geometria estéreo), sensor LiDAR (Light Detection And Ranging), sensores de profundidade e etc. Nessa tarefa, você receberá um conjunto de 30 scans extraídos de um LiDAR, estes dados são parte do KITTI DATASET. Utilize as point clouds de cada scan para estimar a trajetória final do veículo, de forma que essa trajetória se inicie no primeiro scan. Para estimar a trajetória percorrida pelo carro você deve usar o algoritmo Iterative Closest-Points (ICP). Contudo, **você não pode utilizar bibliotecas de terceiros para isso. Você deve fazer sua própria implementação do algoritmo ICP**. Em seu código, você deve mostrar que sua implementação está correta. Além disso, estamos anexando a ground-truth que é um arquivo .npy, que pode ser aberto com a biblioteca NumPy. Ao carregá-lo, você terá um array de tamanho (30, 4, 4). Cada linha, primeiro índice, representa uma matriz de transformação em coordenadas homogêneas para cada uma das 30 posições do carro.

Link para o dataset- Link para a Ground-Truth

OBSERVAÇÕES:

- Use a ground-truth para comparar seu resultado. Com ela, você pode verificar se você está chegando numa resposta próxima da realidade.

- Como descrito no corpo da questão, não é permitido usar bibliotecas que já possuam o algoritmo do ICP implementado. Portanto, você **não pode** usar a implementação das bibliotecas como: Open3D ou PCL. **Contudo**, você pode usar o algoritmo dessas bibliotecas como ground-truth para você verificar se sua implementação está correta.

- Você **pode** usar bibliotecas como SciPy, NumPy, MatPlotLib.

- Caso você não tenha experiência com transformações de corpo rígido, veja essa playlist com vídeo-aulas.

- É **obrigatório** que você demonstre o caminho percorrido, em todos os eixos (XYZ), pelo carro. Ou seja, você **deve** mostrar um plot 3D da trajetória estimada pelo veículo. Ou, se possível, a matriz de transformação final, em coordenadas homogêneas, do carro.

- Para carregar a nuvem de pontos, que estão nos arquivos .obj, você pode utilizar a biblioteca Trimesh.

Veja o exemplo abaixo de como carregar a point cloud:

```
import trimesh

point_cloud = trimesh.load("000000_points.obj").vertice
```

QUESTÃO 2

Um dos desafios do século XXI, no campo do Machine Learning, foi desenvolver modelos de reconhecimento facial que funcionassem para uma grande variedade de pessoas. Originalmente, podemos pensar que cada pessoa fosse um indivíduo único, sendo assim uma classe. Logo, os primeiros modelos foram baseados em modelos de classificação.

Contudo, havia um grande inconveniente nesse processo: toda vez que uma pessoa nova surgia para ser cadastrada no banco de dados, era necessário treinar o modelo novamente, uma vez que você incluiu uma nova classe no seu problema. Antes haviam N classes, com a introdução de uma nova pessoa N + 1. Conforme o sistema fosse aumentando, acabava complicando o procedimento para colocar o sistema em produção novamente. Uma saída para esse problema, que acabou virando um campo específico de Deep Learning, é o Aprendizado de Métrica, no inglês **Metric Learning**. O método consiste em, invés de trabalharmos com um problema de classificação, treinamos o modelo para aprender uma métrica de distância entre duas entidades. A saída de um modelo que foi treinado mediante o método de **Metric Learning** é um vetor descritor. Esse vetor é responsável por computar as características, ou features, que descrevem a face de uma pessoa. O método de **Metric Learning** tornou-se promissor, possibilitando a inclusão de novas pessoas no banco de dados, de forma que não fosse necessário re-treinar o modelo.

Contudo, apesar de todos nós possuímos características únicas em nossas faces, a pandemia do Covid-19 trouxe um grande problema para os modelos de reconhecimento facial, as máscaras faciais. Cobrindo parte importante do nosso rosto, do qual os modelos de deep learning extraíam características para computar o vetor descritor. Com isso, os modelos de reconhecimento facial apresentavam um alto índice de falhas. Algumas soluções foram encontradas para contornar esse problema.

Nesse tarefa, você precisará fazer o seguinte:

1. Treinar um modelo de rede neural nesse conjunto de dados.

   a. O conjunto de dados possui uma grande variedade de celebridades.

2. Uma vez treinado o modelo, você deve criar um banco de dados com as imagens das celebridades com o output do vetor descritor do seu modelo.

3. Após o treinamento da rede neural você deve incluir essa pessoa no banco de dados.

4. Uma vez concluídas as etapas (1), (2) e (3), você deve usar esta imagem, que é a pessoa do item (3) usando máscara facial, e fazer o reconhecimento facial dela.

OBSERVAÇÕES

- Caso você tenha problemas em usar a GPU do Colab, pode utilizar uma rede pré-treinada para criar o vetor descritor de características dos rostos.

- É **obrigatório** exibir a imagem que será usada para inferência, isto é, a imagem do item (4), e exibir qual é o score da métrica que você usará e, também, qual foi a pessoa com a qual ela foi identificada no banco de dados.

- Caso você nunca tenha trabalhado com **metric learning**, aqui está um post detalhado sobre o assunto. LINK.

- Para atingir a pontuação completa da questão, é importante que seu algoritmo consiga fazer o reconhecimento da pessoa do item 4.