

1. Instruções do Teste

Objetivo:

Migrar o código Python fornecido para C++, garantindo eficiência e organização. A interface gráfica deve ser implementada em **Qt** ou **C#**.

Tarefas:

1. **Migrar o Código Python para C++:**
 - Exportar o modelo Faster R-CNN utilizando **ONNX** ou **TorchScript** e realizar a inferência em C++.
 2. **Implementar a Interface Gráfica:**
 - **Opção 1:** Usar **Qt** para C++.
 - **Opção 2:** Usar **C#** com Windows Forms ou WPF.
 3. **Estruturar o Código:**
 - Organizar o código em classes modulares para promover clareza, reutilização e manutenção.
-

2. Critérios de Avaliação

a. Organização e Estrutura de Código

- Modularização e separação de responsabilidades.
- Uso correto de classes e padrões de design, quando aplicável.

b. Correção e Funcionalidade

- A solução final deve replicar corretamente a funcionalidade original.
- A interface gráfica deve permitir o carregamento e exibição de vídeos com detecção de objetos.

c. Documentação e Apresentação

- Criar um **README.md** com:
 - Descrição do projeto e instruções de execução.
 - Detalhes da migração e desafios enfrentados.
 - Justificativa da escolha entre **Qt** ou **C#**.
- Gravar um **vídeo explicativo(máximo de 20 minutos)** demonstrando:
 - O funcionamento da aplicação.
 - A abordagem adotada.
 - Comparações de desempenho entre Python e C++.

d. Performance Básica

- Avaliar a redução no tempo de inferência e no consumo de recursos em relação à versão Python.
-

3. Extra (Opcional): Otimização Avançada

Tarefas Extras:

1. **Otimização de Memória:**
 - Implementar técnicas avançadas de gerenciamento de memória, como o uso de *smart pointers* e controle eficaz de alocação/desalocação.
2. **Melhoria de Desempenho (40% ou mais):**
 - Aplicar paralelismo ou outras otimizações específicas para melhorar a performance em relação à versão Python.

CrITÉrios de Avaliação do Extra:

- **Impacto das Otimizações:** Apresentar métricas antes e depois das melhorias.
 - **Clareza do README e Vídeo:** Explicações detalhadas das técnicas aplicadas e impacto obtido.
-

4. Expectativas Técnicas

Exportação do Modelo:

O modelo Faster R-CNN deve ser exportado para **ONNX** ou **TorchScript** e utilizado no código C++.

Interface Gráfica:

A interface deve manter as seguintes funcionalidades:

- Carregar e exibir vídeos.
- Realizar a detecção de objetos e exibir os resultados sobrepostos no vídeo.

Avaliação de Performance:

Medição do desempenho da aplicação em C++ em comparação com a versão original em Python.

5. Entrega

O candidato deve enviar:

1. **Código-fonte completo.**
2. **README.md** com as informações relevantes, podendo ser:
 - a. Instruções para execução do projeto.
 - b. Desafios enfrentados e soluções implementadas.
 - c. Desempenho
3. **Vídeo demonstrativo e explicativo(máximo de 20 minutos).**

OBS: Pequenos detalhes fazem diferença. Fique à vontade para incluir extras ou melhorias que considere relevantes.

Boa sorte! 😊

Dúvidas: JFS6@CIN.UFPE.BR