

PSI-5796: Processamento e Análise de Imagens e Vídeos
Primeiro período de 2018 **1º exercício-programa**
Data de entrega: 15/04/2018 (domingo) até 24:00 horas
Este EP é igual a EP1-2017

Prof. Hae

- Obs. 1:** Cada dia de atraso acarreta uma perda de 1 ponto no exercício.
Obs. 2: Este EP deve ser resolvido individualmente. EPs iguais receberão nota zero. Também não pode entregar EP igual a algum EP entregue no ano passado.

O banco de dados abaixo contém imagens de placas de trânsito, algumas delas com sujeira:
http://inf-server.inf.uth.gr/~gpotamianos/traffic_sign_database.html
Desse banco de dados, peguei 44 imagens com a placa "proibido virar" (à esquerda ou à direita), renomeei-as de 00.jpg até 43.jpg, e deixei em:
http://www.lps.usp.br/hae/psi5796/ep1-2018/proibido_virar.zip



Faça um programa C/C++ ep1.cpp que lê uma dessas 44 imagens e detecta (localiza) a placa "proibido virar". A forma de chamar o programa deve ser:

```
linux$ ep1 ../37.jpg ~/diretorio/e37.png  
windows> ep1 ..\37.jpg c:\diretorio\e37.png
```

Se chamar o programa com parâmetros incorretos, o programa deve imprimir alguma mensagem amigável:

```
diretorio> ep1  
ep1: Detecta placa "proibido virar"  
sintaxe: ep1 ent.ext sai.ext  
Erro: Numero de argumentos invalido
```



Você pode escolher até 8 imagens (das 44 imagens) para servirem como "amostras de treinamento" ou "modelos" a serem procurados. Você pode editar as imagens escolhidas para criar as imagens-amostras de treinamento. Estas imagens devem estar no diretório default, podem ser lidos por seu programa e devem ser entregues juntos com o seu programa. Neste caso, deixe explícito no relatório quais foram as imagens escolhidas para servirem de "amostras" e como foram editadas. Para padronizar, nomeie estas imagens como am*.* (use o prefixo "am").

Deixei as 44 saídas geradas pelo meu programa em:

http://www.lps.usp.br/hae/psi5796/ep1-2018/saida_professor.zip

Você pode usar o fato de que em cada imagem aparece uma e somente uma placa "proibido virar". Note que este conhecimento facilita muito a resolução do problema. É muito mais difícil detectar todas as placas se você não sabe quantas placas aparecem em cada imagem.

Opcionalmente, você pode classificar (além de localizar) as placas em "proibido virar à direita" ou "proibido virar à esquerda", escrevendo letras D ou E na imagem de saída. Neste caso, o seu exercício pode receber até nota 12.

Obs. 1: Pode usar (se quiser) a biblioteca Cekeikon/OpenCV.

Obs. 2: Entregue o programa-fonte (ep1.cpp), os modelos e outros dados necessários (am*.* para a execução do seu programa, e um documento PDF (relatorio.pdf) com os comentários descrevendo o funcionamento do programa. O envio do relatório é obrigatório (veja o anexo).

(a) Se você fez o programa no ambiente usado na classe (Cekeikon/OpenCV, Windows/Linux), ou fez usando só OpenCV em Mac, basta entregar o programa-fonte ep1.cpp para poder corrigir o seu programa.

(b) Se você quiser fazer usar alguma biblioteca diferente ou programar em outro ambiente, converse antes com o professor.

Obs. 3: Compacte todos os arquivos como SeuNome_Sobrenome.ZIP e envie um email para:

- hae@lps.usp.br

Se você enviar mais de um email, considerarei somente o último email enviado, apagando os anteriores.

Anexo: Relatórios dos exercícios programas

O mais importante numa comunicação escrita é que o leitor entenda, sem esforço e inequivocamente, o que o escritor quis dizer. O texto ficar "bonito" é um aspecto secundário. Se uma (pseudo) regra de escrita dificultar o entendimento do leitor, essa regra está indo contra a finalidade primária da comunicação. No site do governo americano [1], há regras denominadas de "plain language" para que comunicações governamentais sejam escritas de forma clara. As ideias por trás dessas regras podem ser usadas em outros domínios, como na escrita científica. Resumo abaixo algumas dessas ideias.

(1) Escreva para a sua audiência. No caso do relatório, a sua audiência será o professor ou o monitor que irá corrigir o seu exercício. Você deve focar na informação que o seu leitor quer conhecer. Não precisa escrever informações que são inúteis ou óbvias para o seu leitor.

(2) Organize a informação. Você é livre para organizar o relatório como achar melhor, porém sempre procurando facilitar o entendimento do leitor. Seja breve. Quebre o texto em seções com títulos claros. Use sentenças curtas. Elimine as frases e palavras que podem ser retiradas sem prejudicar o entendimento. Use sentenças em ordem direta (sujeito-verbo-predicado).

(3) Use palavras simples. Use o tempo verbal o mais simples possível. Evite cadeia longa de nomes, substituindo-os por verbos (em vez de "desenvolvimento de procedimento de proteção de segurança de trabalhadores de minas subterrâneas" escreva "desenvolvendo procedimentos para proteger a segurança dos trabalhadores em minas subterrâneas"). Minimizar o uso de abreviações (para que o leitor não tenha que decorá-las). Use sempre o mesmo termo para se referir à mesma realidade (pode confundir o leitor se usar termos diferentes para se referir a uma mesma coisa). O relatório não é obra literária, não tem problema repetir várias vezes a mesma palavra.

(4) Use voz ativa. Deixe claro quem fez o quê. Se você utilizar oração com sujeito indeterminado ou na voz passiva, o leitor pode não entender quem foi o responsável (Ex: "Criou-se um novo algoritmo" - Quem criou? Você? Ou algum autor da literatura científica?). O site diz: "Passive voice obscures who is responsible for what and is one of the biggest problems with government writing."

(5) Use exemplos, diagramas, tabelas, figuras e listas. Ajudam bastante o entendimento.

O relatório deve conter pelo menos as seguintes informações:

Identificação

Seu nome, número USP, nome da disciplina, etc.

Breve enunciado do problema

Apesar do enunciado do problema ser conhecido ao professor/monitor, descreva brevemente o problema que está resolvendo. Isto tornará o documento compreensível para alguma pessoa que não tem o enunciado do EP à mão.

Técnica(s) utilizada(s) para resolver o problema

Descreva quais técnicas você usou para resolver o problema. Se você mesmo inventou a técnica, descreva a sua ideia, deixando claro que a ideia foi sua. Se você utilizou alguma técnica já conhecida, utilize o nome próprio da técnica (por exemplo, filtragem Gaussiana, algoritmo SIFT, etc.) juntamente com alguma referência bibliográfica onde a técnica está descrita. Use elementos gráficos como imagens intermediárias e diagramas, pois ajudam muito a compreensão. Não "copie-e-cole" código-fonte, a não ser que seja relevante. Use preferencialmente o pseudo-código.

Ambiente de desenvolvimento utilizado

1 <https://plainlanguage.gov/guidelines/>

Em qual plataforma você desenvolveu o programa? Como o professor/monitor pode compilar o programa? Você utilizou que bibliotecas?

Operação

Como o professor/monitor pode executar o programa? Que argumentos são necessários para a execução do programa? Há parâmetros que devem ser configurados? Quais arquivos de entrada são necessários? Quais arquivos de saída são gerados?

Resultados Obtidos

Descreva os resultados obtidos. Qual é o tempo de processamento típico? O problema foi resolvido de forma satisfatória?

Referências

Descreva o material externo utilizado, como livros/artigos consultados, websites visitados, etc.