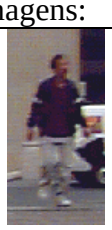
PSI-5796 Algoritmos para Processamento e Análise de Imagens
Segundo Período de 2011 2º exercício-programa Prof. Hae
Data de entrega: 28/agosto/2011 (domingo) até 24:00 horas

Deixei no site dois grupos de imagens que peguei do site do MIT, todos de tamanho 128x64:

- www.lps.usp.br/~hae/psi5796/ep1-2011/pedestre.zip
  - per00001.ppm até per00924.ppm
- www.lps.usp.br/~hae/psi5796/ep1-2011/carro.zip
  - carl0001.ppm até carl0516.ppm e carr0001.ppm até carr0516.ppm

Alguns exemplos de imagens:

				
per00001.ppm	per00003.ppm	carl0001.ppm	carr0001.ppm	carr0002.ppm

O objetivo é fazer um sistema que classifique uma imagem como pedestre ou carro. Para isso, faça um treinamento supervisionado (isto é, sabendo quais são as imagens de pedestres e quais são as imagens de carros) com as 400 primeiras imagens de pedestres (per00001.ppm até per00400.ppm) e as 400 primeiras imagens de carros (carl0001.ppm até carl0200.ppm e carr0001.ppm até carr0200.ppm). Depois, o sistema deve classificar todas as imagens (tanto as vistas como as não-vistas) como um carro ou um pedestre, baseado no conteúdo da imagem (isto é, sem usar a informação presente no nome da imagem).

Faça dois programas em C/C++ **treina.cpp** e **classifica.cpp**. O primeiro programa deve ler as 800 imagens de treino e gerar um arquivo "treina.txt" que contém a regra de decisão:

```
C:>treina
```

O segundo programa deve ler a regra de decisão "treina.txt" e classificar todas as imagens (vistas e não-vistas no treino):

```
C:>classifica
```

Responde:

```
carl0001.ppm: carro
...
carl0516.ppm: carro
carr0001.ppm: carro
...
carr0516.ppm: carro
per00001.ppm: pedestre
per00002.ppm: pedestre
...
per00024.ppm: pedestre
```

Nota: Para facilitar o desenvolvimento e os testes, sugiro quebrar o programa ***treina*** em dois sub-programas ***features*** e ***aprendiz*** com os argumentos:

`C:>features features.txt +1 per000??.ppm`

Extrai as características das imagens `per000??.ppm` e os armazena no arquivo `features.txt` com o rótulo +1 que indica que são imagens de pedestres. Note que ?? pode substituir por qualquer sequência de dois caracteres.

`C:>features features.txt -1 car?00??.ppm`

Extrai as características das imagens `car?00??.ppm` e os acrescenta no final do arquivo `features.txt` com o rótulo -1 que indica que não são imagens de pedestres.

`c:>aprendiz features.txt decisao.xml`

Cria o critério de decisão ***decisao.xml*** usando alguma técnica de aprendizagem de máquina a partir das características em ***features.txt***.

Também sugiro que o programa ***classifica*** aceite os seguintes argumentos:

`c:>classifica decisao.xml per00???.ppm`

Classifica os arquivos ***per00???.ppm*** usando o critério de decisão em ***decisao.xml***.

Nota: O programa argument.cpp abaixo imprime todos os argumentos do programa.

```
#include <cekeikon.h>
int main(int argc, char** argv)
{ for (int i=0; i<argc; i++)
    printf("Argumento %d=%s\n",i,argv[i]);
}
```

Se executar:

`c:\>argument per0000?.ppm`

no diretório que contém o banco de dados dos pedestres, o programa responderá:

```
Argumento 0=argument
Argumento 1=per00001.ppm
Argumento 2=per00002.ppm
Argumento 3=per00003.ppm
Argumento 4=per00004.ppm
Argumento 5=per00005.ppm
Argumento 6=per00006.ppm
Argumento 7=per00007.ppm
Argumento 8=per00008.ppm
Argumento 9=per00009.ppm
```

Observações:

1: Cada dia de atraso acarreta uma perda de 1 ponto no exercício.

2: Este EP deve ser resolvido individualmente. Não serão aceitos EPs iguais ou em grupo.

3: Pode usar (ou não) as funções das bibliotecas Proeikon, Cekeikon e OpenCV.

4: Programas muito lentos terão nota descontada. Programas que cometem muitos erros de classificação terão nota descontada.

5: Se você fez os programas utilizando Devcpp/Proeikon ou TDM-GCC/Cekeikon, entregue apenas os programas fontes `treina.cpp` e `classifica.cpp`; o arquivo de saída `saida.txt` obtido rodando o seu programa e um documento `coment.pdf` ou `coment.doc` com os comentários que achar convenientes. Neste documento, descreva (em português) o seu método, a seqüência de operações efetuadas e inclua (se quiser) as imagens intermediárias que possibilite entender o seu método. Também inclua a taxa de acerto em % das imagens vistas e não-vistas. Informe também o tempo de processamento e o modelo do seu computador. Se você fez os programas utilizando compilador ou biblioteca diferente, deve entregar também os programas executáveis. **Cuidado:** Alguns servidores de emails possuem anti-vírus que bloqueia envio/recepção de emails com arquivo EXE embutido. Renomeie EXE como EEE.

6: Compacte todos os arquivos como **SeuNome_Sobrenome.ZIP** e envie um email colocando como assunto **"PSI5796 EP2"** para o endereço abaixo:

- **hae@lps.usp.br**

7: Procure enviar um único email para entregar o seu EP. Se você enviar dois ou mais emails, vou apagar todos os emails antigos, considerando somente o email enviado por último.