**PSI-3471: Fundamentos de Sistemas Eletrônicos Inteligentes**
**Primeiro semestre de 2018 2º exercício-programa Prof. Hae**
**Data de entrega: 03/07/2018 (terça-feira) até 24:00 horas**
[Este EP é semelhante a EP2 de PSI-5796 (pós-graduação), 2018]

**Obs. 1:** Cada dia de atraso acarreta perda de 1 ponto no exercício.
**Obs. 2:** Este EP deve ser resolvido em duplas ou individualmente. EPs iguais receberão nota zero. Também não pode entregar EP igual a algum EP entregue nos outros cursos.

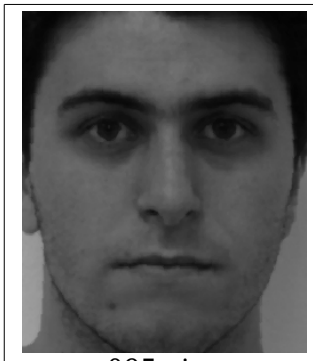
O objetivo deste exercício é exercitar-se no uso de rede neural convolucional. Para isso, vamos resolver dois problemas:

a) Identificar se um rosto é masculino ou feminino.
b) Identificar se um rosto está com expressão neutra ou sorridente.

Para isso, vamos usar o banco de dado de faces humanas abaixo:

http://fei.edu.br/~cet/frontalimages_spatiallynormalized_part1.zip
http://fei.edu.br/~cet/frontalimages_spatiallynormalized_part2.zip

Baixei os dois "ZIPs" e descompactei-os, resultando em 400 imagens JPG em níveis de cinza, com 300x250 pixels, de rostos frontais de expressão neutra (*a.jpg) e sorridente (*b.jpg), alinhados espacialmente. Renomeei algumas dessas imagens, para que todas tenham nomes com 4 caracteres, por exemplo, 1a.jpg torne-se 001a.jpg. Dessas imagens, metade são rostos masculinos e metade são femininos; metade está com expressão neutra e metade está sorridente.


095a.jpg


095b.jpg


096a.jpg


096b.jpg

Acrescentei seis arquivos csv (comma-separated values): mf_treino.csv, mf_valida.csv, mf_teste.csv (a serem usados na identificação masculino/feminino) e ns_treino.csv, ns_valida.csv, ns_teste.csv (a serem usados na identificação neutro/sorridente). Os arquivos ??_treino.csv contém os nomes de 200 imagens a serem usadas durante o treinamento de aprendizagem de máquina, com rótulos 0=masculino e 1=feminino ou 0=neutro e 1=sorridente. Trechos desses arquivos:

```
001a.jpg;0
001b.jpg;0
...
096a.jpg;1
096b.jpg;1
...
```

Os arquivos ??_valida.csv e ??_teste.csv são semelhantes e contêm, cada um, os nomes de 100 imagens (50 masculinos e 50 femininos ou 50 neutros e 50 sorridentes) a serem usadas para validação e teste. O "zip" abaixo contém todos esses arquivos:

www.lps.usp.br/hae/psi3471/ep2-2018/frontalImages_spatiallynormalized_with_csv.zip

Faça quatro programas C/C++ mf_treino.cpp, mf_teste.cpp, ns_treino.cpp, ns_teste.cpp. O programa ??_treino.cpp deve fazer o treino (usando algum método de aprendizagem de máquina) utilizando as imagens listadas no arquivo ??_treino.csv. Durante o treino, você deve utilizar as imagens listadas no arquivo ??_valida.csv para verificar a taxa de erro de validação do seu método. Você deve gravar o arquivo "??_rede.net" no diretório default com a rede que apresentar a menor taxa de erro de validação ou com a rede obtida no final da otimização.

Depois do treino, ??_teste.cpp deve carregar ??_rede.net e classificar as imagens ??_valida.csv e ??_teste.csv. Imprima a taxa de erro de teste e os nomes das imagens classificadas incorretamente. A forma de chamar os programas deve ser:

```
$ mf_treino diretorio_onde_esta_banco_de_dados
(Usa banco de dados no diretório "diretorio_onde_esta_banco_de_dados" para gerar mf_rede.net no diretório default)
$ mf_teste diretorio_onde_esta_banco_de_dados
(Lê mf_rede.net do diretório default e classifica as imagens testes no diretório "diretorio_onde_esta_banco_de_dados")

$ ns_treino diretorio_onde_esta_banco_de_dados
$ ns_teste diretorio_onde_esta_banco_de_dados
```

Se chamar um programa com parâmetros incorretos ou se algum arquivo não existir no diretório especificado, o programa deve imprimir alguma mensagem amigável.

A saída de ??_treino.cpp deve ser algo como:

```
A menor taxa de erro de validacao: 2%
Tempo de processamento: 4 minutos.
```

A saída de ??_teste.cpp deve ser algo como:

```
Taxa de erro de validacao: 2%
Arquivos de validacao classificados incorretamente:
xxxb.jpg
yyya.jpg
Taxa de erro de teste: 3%
Arquivos de teste classificados incorretamente:
lllb.jpg
mmmb.jpg
nnna.jpg
```

Para classificação masculino/feminino, o meu programa obteve erro de 1%, tanto em validação como em teste, com tempo de treino de 6 minutos em Linux (em Windows demora mais, por não usar a biblioteca TBB). Para classificação neutro/sorridente, o meu programa obteve erro de 3% na validação e 6% no teste, com tempo de treino de 1,5 minutos. A alta taxa de erro na classificação neutro/sorridente pode ser explicada olhando as imagens classificadas incorretamente. Há rostos supostamente neutros que na verdade estão sorrindo, assim como há rostos supostamente sorridentes que na verdade estão meio carrancudos.

**Obs. 1:** Os programas mf_treino.cpp e ns_treino.cpp são praticamente iguais. Você só precisa alterar alguns parâmetros de um programa para outro para otimizar as taxas de acerto. De forma semelhante, os programas mf_teste.cpp e ns_teste.cpp são praticamente iguais.

**Obs. 2:** Pode usar (se quiser) a biblioteca Cekeikon/OpenCV/Tiny_dnn.

**Obs. 3:** Entregue os 4 programas-fontes (??_treino.cpp e ??_teste.cpp), as 2 redes obtidas (??_rede.net) e um documento PDF (relatorio.pdf) com os comentários descrevendo o funcionamento do programa, o tempo de processamento e as taxas de erro obtidas. O envio do relatório é obrigatório (veja o anexo).

(a) Se você fez o programa no ambiente usado na classe (Cekeikon/OpenCV/Tiny_dnn, Windows/Linux), basta entregar os programas-fontes para poder corrigir o seu programa.

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

**Obs. 4:** Compacte todos os arquivos como nome_sobrenome.zip (trabalho individual) ou nome1_sobrenome1_nome2_sobrenome2.ZIP (trabalho em dupla) e envie o arquivo através de edisciplinas. Dentro do prazo, você pode substituir o arquivo anterior por um novo. Só o último arquivo entregue será corrigido.

**Obs. 5:** Se o trabalho foi feito em dupla, envie um único trabalho em nome de qualquer um dos dois integrantes do grupo. Isto é, não envie dois trabalhos iguais.

## 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 enfocar 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"). Minimize 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*

Nomes dos integrantes do grupo, números 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? Qual foi a taxa de erro obtida? O problema foi resolvido de forma satisfatória?

*Referências*

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