

PSI3472 Conceção e Implementação de Sistemas Eletrônicos Inteligentes
Segundo Semestre de 2018 **3º exercício-programa**

O objetivo deste exercício é usar rede convolucional para fazer regressão (em vez de classificação, como fizemos no semestre passado). Vamos desenvolver um programa que faz super-resolução de imagens de rostos (isto é, aumenta a resolução das imagens).

Como no primeiro semestre, 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 alinhados espacialmente. Renomeei algumas dessas imagens, para que todas tenham nomes com 4 caracteres, por exemplo, 1a.jpg torne-se 001a.jpg.

Reduzi a resolução dessas imagens para 38x32 pixels, depois cortei as linhas superiores e inferiores, resultando em 400 imagens 32x32. Normalizei essas imagens de forma que o pixel mais escuro se torne 0 e o mais claro se torne 255, gerando 400 imagens ????.png. Reduzi a resolução dessas imagens para 8x8 usando o método INTER_AREA do OpenCV, resultando em 400 imagens ????.png.

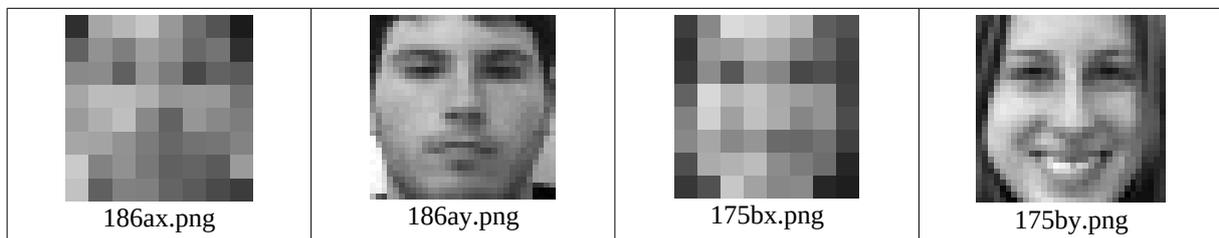


Figura 1: Imagens de entrada 8x8 (X) e as imagens de saída ideais 32x32 (Y).

O objetivo deste exercício é desenvolver um programa que recebe uma imagem 8x8 ????.png e aumenta a resolução para 32x32, gerando a imagem ????.png, de forma que a imagem processada seja a mais semelhante possível da imagem ideal ????.png. A semelhança será medida pela métrica MSE (média do erro quadrático).

Também criei 3 arquivos CSV (comma separated values), especificando quais pares de imagens devem ser usadas no treino, na validação e no teste:

treino.csv: 200 pares de imagens x y

valida.csv: 100 pares de imagens x y

teste.csv : 100 pares de imagens x y

Para ilustração, as seis primeiras linhas do treino.csv são:

001ax.png;001ay.png

001bx.png;001by.png

002ax.png;002ay.png

002bx.png;002by.png

005ax.png;005ay.png

005bx.png;005by.png

Esses arquivos estão em:

www.lps.usp.br/hae/psi3472/ep3-2018/feiFrontRed.zip

Faça três programas C++ treino.cpp, supres.cpp e mse.cpp.

O programa treino.cpp deve fazer o treino utilizando as pares de imagens listadas no arquivo treino.csv como amostras de treinamento, através de algum método de aprendizagem de máquina. Durante o treino, você pode ajustar os parâmetros da rede usando as pares de imagens listadas no arquivo valida.csv. Calcule MSE considerando preto=0 e branco=1. Você deve gravar o arquivo "rede.net" no diretório default com a rede obtida.

```
$ treino diretorio_onde_esta_banco_de_dados
```

(Usa banco de dados no diretório "diretorio_onde_esta_banco_de_dados" para gerar rede.net no diretório default.)

A saída de treino.cpp deve ser algo como:

```
MSE médio de validacao: 0.00603193
```

```
Tempo de processamento: 12.4 minutos.
```

Aula 7, exercício 3 (3 pontos): Escreva o programa leitura.cpp que recebe o nome do diretório do banco de dados e lê as imagens listadas em treino.csv, valida.csv e teste.csv em estruturas do tipo `vector< Mat_<FLT> >`, com nomes AX (treino 8x8), AY (treino 32x32), VX, VY (validação), QX, QY (teste ou query). Para se certificar de que a leitura está correta, mostre na tela as imagens de índice especificada pelo usuário.

```
>leitura diretorio_onde_esta_banco_de_dados 20
```

Deve mostrar as 6 imagens de índice 20.

Aula 7, exercício 4 (3 pontos): Escreva o programa leitura_tiny.cpp que, além de ler as imagens, faz super-resolução das imagens de entrada (AX, VX e QX) para 32x32 pixels e converte todas as imagens para o formato `tiny_dnn vector<vec_t>`. Vamos chamar as 6 estruturas obtidas de ax, ay, vx, vy, qx e qy. Para se certificar de que a conversão está correta, pegue as 6 imagens formato `tiny_dnn` de índice especificada pelo usuário, as reconverta para o formato OpenCV (`Mat_<FLT>` ou `Mat_<GRY>`) e as mostre na tela.

```
>leitura_tiny diretorio_onde_esta_banco_de_dados 20
```

Deve mostrar as 6 imagens de índice 20, só que fazendo conversão para o formato `tiny_dnn` e conversão reversa para OpenCV.

Nota: É melhor que, no `tiny_dnn`, os níveis de cinza estejam de -1 a +1 ou de -0.5 a +0.5 (do que de 0 a 1).

Aula 8 (10 pontos - Os exercícios desta aula tem peso 2): Faça os 3 programas de acordo com o enunciado deste documento. Escreva o relatório de acordo com o enunciado. Entregue via edisciplinas até a data indicada.

Depois do treino, supres.cpp deve carregar rede.net do diretório default e uma imagem 8x8 e gravar a imagem 32x32 obtida pela super-resolução.

```
$ supres ../diretorio/004bx.png 004bp.png
```

(Usa rede.net do diretório default para aumentar resolução da imagem 8x8 ../diretorio/004bx.png e grava a imagem 32x32 004bp.png no diretório default.)

O programa mse.cpp deve fazer superresolução de todas as imagens de treino, validação e teste e calcular a média dos MSEs obtidos para cada conjunto de imagens. MSE deve ser calculado considerando preto=0 e branco=1.

```
$ mse diretorio_onde_esta_banco_de_dados
```

(Usa rede.net do diretório default e banco de dados do diretório "diretorio_onde_esta_banco_de_dados" para calcular a média dos MSEs de treino.csv, valida.csv e teste.csv)

```
Media de MSE de treino.csv: 0.0042981
```

```
Media de MSE de valida.csv: 0.00607254
```

```
Media de MSE de teste.csv: 0.0062042
```

Todos os programas devem emitir uma mensagem amigável se os argumentos do programa não estiverem no formato esperado, assim como se o programa não encontrar algum arquivo necessário.



Figura 2: Da esquerda para direita: Imagem original 8x8; interpolação bilinear para 32x32; interpolação bicúbica para 32x32; super-resolução com rede neural convolucional; saída ideal 32x32.

Tabela 1: As médias de MSEs de diferentes métodos de reamostragem (considerando pixels de 0 a 1):

	treino.csv	valida.csv	teste.csv
nearest	0.0151048	0.0152761	0.0153826
linear	0.0128697	0.0129207	0.0131465
cubic	0.0112363	0.0111828	0.0115495
meu programa	0.0042981	0.00607254	0.0062042

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

Obs. 2: Entregue os 3 programas-fontes (treino.cpp, supres.cpp e mse.cpp), a rede obtida (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. 3: 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. 4: 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.

Referências:

- Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2), 295-307.

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"). 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.

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

Ambiente de desenvolvimento utilizado

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.