

PSI3472 Conceção e Implementação de Sistemas Eletrônicos Inteligentes
Segundo Semestre de 2020 **Exercício-programa**
Data de entrega: 18/10/2020 (domingo) até 23:59 horas

O objetivo deste exercício é colorir automaticamente imagens em níveis de cinza. Como o esforço computacional necessário para treinar uma rede capaz de colorir uma imagem genérica está fora das possibilidades deste exercício, vamos limitar o domínio do problema, colorindo apenas as faces de um banco de dados.

O site:

<https://fei.edu.br/~cet/facedatabase.html>

contém bancos de dados de faces humanas. Baixei 400 imagens JPG coloridas, numeradas de 001 a 200, com 360×260 pixels, de rostos frontais de expressão neutra (*ac.jpg) e sorridente (*bc.jpg). Redimensionei as imagens originais para 384×256 pixels, para que sejam divisíveis por 2 várias vezes, facilitando processá-las com redes convolucionais (coluna esquerda da figura 1).

No problema de colorir, normalmente o sistema de cores CieLAB é utilizado (veja o anexo A). Assim, converti as imagens RGB para CieLAB e gravei o componente L, obtendo as imagens em níveis de cinzas (*l.jpg, coluna do meio da figura 1). As 800 imagens assim obtidas (400 coloridas *c.jpg e 400 imagens em níveis de cinza das componentes L *l.jpg) estão em:

<http://www.lps.usp.br/hae/psi3472/ep-2020/feiFrontG2C.zip> OU

<https://drive.google.com/drive/folders/1boM6Ecv7Bbhg8ERJzrEHTR9vqBcH16Xv?usp=sharing>

O problema é, dada uma imagem em níveis de cinza (componente L do CieLAB) colorir-la, isto é, achar os componentes A e B do CieLAB que especificam as cores dos pixels. Para isso, utilize as imagens 001 a 100 para treino, 101 a 150 para validação e 151 a 200 para teste.

A coluna direita da figura 1 mostra algumas imagens de teste coloridas desta forma. As imagens coloridas automaticamente estão com qualidade visual razoável, exceto pela cor do cachecol verde que a rede coloriu como marrom. Note que não há como rede neural descobrir qual é a cor original do cachecol olhando somente a imagem em níveis de cinza.

Usei MSE para medir a diferença entre os componentes AB das imagens originais e os das imagens coloridas artificialmente. Obtive erro de teste de 0,00025 (com os valores de AB indo de -0,5 a +0,5 ou de 0,0 a 1,0). Procure obter erro menor ou igual a esse valor.

Você deve criar dois notebooks Colab:

1) psi3472_treina: Lê as imagens de treino e de validação e cria o modelo de rede psi3472.h5. Também deve imprimir os erros MSE de treino e de validação.

2) psi3472_testa: Lê as imagens de teste e o modelo psi3472.h5, faz predição e grava as imagens coloridas obtidas. Também deve imprimir o erro de teste.

Você deve gravar um vídeo de no máximo 45 segundos mostrando que o seu sistema funciona, juntamente com as taxas de erro e algumas imagens coloridas automaticamente.

Você também deve entregar um relatório de no máximo 3 páginas (veja anexo B).

		
		
		
Imagem colorida original	Imagem em níveis de cinza	Imagem colorida pela rede convolucional.

Figura 1: Imagens coloridas originais, em níveis de cinza (componente L do sistema CieLAB) e imagens coloridas automaticamente pela rede convolucional. Note que não há como rede descobrir que a cor do cachecol da terceira imagem é verde.

- Obs. 1:** Este exercício deve ser resolvido individualmente.
- Obs. 2:** Em princípio, deve usar o mesmo ambiente de programação utilizado em aula (Python, Keras, OpenCV, Colab). Se quiser utilizar outro ambiente, converse antes com o professor.
- Obs. 3:** Entregue o vídeo de no máximo 45 segundos e um relatório PDF de no máximo 3 páginas descrevendo o seu programa em edisciplinas. O seu programa será corrigido em princípio baseado apenas nesses materiais.
- Obs. 4:** Entregue também os links dos Notebooks Colab, os modelos e outros dados necessários para a execução do seu programa em edisciplinas (pode colocar os links dentro do seu relatório). Este material só será analisado se o professor ficar com alguma dúvida e não conseguir atribuir uma nota olhando somente o vídeo e o relatório. Você deve entregar os links de modo que o professor consiga rodar o seu programa, se necessário.
- Obs. 5:** Envie o material através de edisciplinas. Dentro do prazo, você pode substituir o material anterior por um novo.

Anexo A: Sistema de cores CieLAB

CieLAB (também conhecido como CIE L*a*b*) é um espaço de cores projetado para que uma certa quantidade de mudança nesse espaço corresponda aproximadamente à mesma quantidade de mudança na percepção visual humana. Por ter sido projetado para se aproximar à visão humana, CieLAB é utilizado nos problemas onde é importante que o erro de coloração medido no espaço das cores seja equivalente ao erro de coloração percebido por seres humanos [wikiLAB].

Esse sistema expressa a cor como três valores L , A e B . OpenCV utiliza lightness L de preto (0) a branco (100); A de verde (-127) a vermelho (+127); e B de azul (-127) a amarelo (+127) [OpenCV]. Nas matrizes em float32, OpenCV armazena pixel em CieLAB nesse formato ($0 \leq L \leq 100$, $-127 \leq A \leq +127$, $-127 \leq B \leq +127$). Mas matrizes em uint8, OpenCV faz a conversão abaixo para que os valores estejam entre 0 e 255:

$$L = L * 255 / 100, A = A + 128, B = B + 128.$$

OpenCV possui a função `cvtColor` que converte imagem RGB (ou BGR) de/para CieLAB:

```
lab=cv2.cvtColor(bgr,COLOR_BGR2LAB)
bgr=cv2.cvtColor(lab,COLOR_LAB2BGR)
lab=cv2.cvtColor(rgb,COLOR_RGB2LAB)
rgb=cv2.cvtColor(lab,COLOR_LAB2RGB)
```

Tome cuidado que OpenCV trabalha com sistema BGR enquanto que Pyplot trabalha com sistema RGB.

[wikiLAB] https://en.wikipedia.org/wiki/CIELAB_color_space

[OpenCV] https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html

Anexo B: Relatórios dos exercícios programados

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 do aluno, 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.

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? O problema foi resolvido de forma satisfatória?

Referências

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