

**PSI-5790: Aprendizado Profundo para Visão Computacional**  
**Primeiro período de 2023 Exercício-programa Prof. Hae**  
**Data de entrega: 11/06/2023 até 23:59 horas**

**Obs. 1:** Cada dia de atraso acarreta perda de 1 ponto no exercício. Não será aceita entrega após 18/06/2023, 23:59.

**Obs. 2:** Este EP deve ser resolvido individualmente. EPs iguais receberão nota zero.

No site abaixo:

<https://www.kaggle.com/datasets/rawatjitesh/avengers-face-recognition>

há um conjunto de 274 imagens faciais dos 5 atores de vingadores.



Figura 1: Exemplos de imagens faciais dos 5 atores de vingadores.

Para cada um dos 5 atores, há entre  $n=50$  e  $n=66$  imagens de faces, numeradas de 1 a  $n$ . Para uniformizar, vamos considerar somente as imagens de 1 a 50 de cada ator (vamos desprezar as imagens com numeração acima de 50). O objetivo deste exercício é achar uma técnica que consegue identificar o ator com alta probabilidade. Para isso, vamos usar 2-fold cross validation. No fold 1, vamos usar como dados de treino as imagens de 1 a 25 ( $5 \times 25 = 125$  imagens) e como dados de teste as imagens de 26 a 50 ( $5 \times 25 = 125$  imagens). Depois, no fold 2, vamos inverter as imagens de treino e de teste, isto é, treinar com as imagens de 26 a 50 e testar usando as imagens de 1 a 25.

Apresente as duas taxas de acerto dos dois folds, para cada um dos seus programas, juntamente com os tempos de treino e teste.

Você deve obrigatoriamente resolver este problema usando pelo menos duas técnicas diferentes. Opcionalmente, você pode implementar uma terceira técnica.

**Técnica 1 - obrigatória: Usar qualquer rede neural convolucional não projetado especificamente para reconhecer faces.**

*Nota:* Neste item, procure atingir a menor taxa de erro possível. Sugiro que implementem a técnica (c).

a) A primeira ideia seria usar simplesmente alguma estrutura de rede neural convolucional e treinar o modelo do zero para classificar os 5 atores.

b) Você pode pré-processar as imagens (tanto de treino quanto de teste) de alguma forma para aumentar a taxa de acerto.

c) Uma técnica melhor seria pegar uma rede já treinada (por exemplo, com pesos treinados para classificar as imagens de ImageNet) e efetuar “transfer-learning” e “fine tuning” para este problema. Aqui, não pode usar uma rede treinada especificamente para reconhecer faces. Acredito que este deve resultar numa baixa taxa de erro.

d) Uma outra ideia seria usar “self-supervised learning”. Para isso, você pode usar quaisquer conjuntos de imagens sem rótulos (inclusive de faces humanas) para treinar a tarefa-pretexto. A tarefa-jusante só pode ser treinada usando as 125 imagens de treino rotuladas acima.

e) Você pode misturar as técnicas acima ou escolher outras técnicas para melhorar a taxa de acerto, desde que não use conjunto de faces humanas rotuladas.

No fold 1, obtive 36 erros ao classificar as 125 imagens de teste (28,8%). No fold 2, obtive 39 erros ou 31,2% de erro. Média = 30% de erro.

**Técnica 2 - obrigatória: Usar módulo DeepFace.**

*Nota:* Neste item, procure atingir a menor taxa de erro possível.

Na internet, existem modelos prontos para reconhecer faces. Aparentemente, *DeepFace* de Sefik Serengil é uma das bibliotecas mais populares.

<https://pypi.org/project/deepface/>

<https://github.com/serengil/deepface>

<https://viso.ai/computer-vision/deepface/>

<https://sefiks.com/2020/05/01/a-gentle-introduction-to-face-recognition-in-deep-learning/>

[https://www.youtube.com/playlist?list=PLsS\\_1RYmYQQFdWqxQggXHynP1rqaYXv\\_E](https://www.youtube.com/playlist?list=PLsS_1RYmYQQFdWqxQggXHynP1rqaYXv_E)

Utilize essa biblioteca para resolver este EP.

No fold 1, obtive 1 erro ao classificar as 125 imagens de teste (0,8% de erro). No fold 2, obtive 2 erros ou 1,6%. Média = 1,2% de erro.

**Técnica 3 - opcional (vale +2 pontos): Usar qualquer outra técnica projetada especificamente para reconhecer faces.**

*Nota:* Neste item, procure atingir a menor taxa de erro possível, mas pode ser que não seja possível atingir uma taxa de erro baixa, menor que a obtida pela técnica 2.

a) O [[exercício-programa 2 de 2017](#)] consiste em reconhecer faces usando as técnicas clássicas Eigenface, Fisherface e LBPH [[OpenCV2](#)]. Poderia usar essas técnicas juntamente com algum algoritmo de aprendizado (incluindo as redes neurais convolucionais) e verificar a taxa de erro. Se você testar estas técnicas “clássicas”, quase certamente a taxa de erro será mais alta que as técnicas anteriores – não tem problema.

b) A apostila [[oneshot-ead.odt](#)] apresenta “one-shot learning”, que pode ser usada para reconhecimento facial. Você pode usar quaisquer de conjunto de faces rotuladas (exceto o conjunto usados neste exercício) para treinar uma rede one-shot learning do zero. Depois, utilize a rede treinada para resolver o problema deste EP, desta vez usando apenas as 125 imagens-treinos como exemplos de treino em 2 folds, conforme descrito acima. Evidentemente, será muito difícil atingir taxa de erro menor que DeepFace com modelos criados pelas grandes empresas – não tem problema.

c) Você pode pesquisar outras técnicas de reconhecimento facial na internet.

**Obs. 1:** Entregue dois ou três programas-fontes (ep-t1.ext, ep-t2.ext e ep-t3.ext onde ext = “py” ou “cpp”; não envie “ipynb”). Ou então, você pode enviar links para notebooks Google Colab – neste caso, assegure que [hae.kim@usp.br](mailto:hae.kim@usp.br) tenha acesso ao seu notebook.

**Obs. 2:** Entregue um documento relatorio.pdf de no máximo 4 páginas descrevendo o funcionamento dos seus programas e os resultados obtidos (taxa de acerto, desvio-padrão e tempo de execução). O envio do relatório é obrigatório (veja o anexo).

**Obs. 3:** Entregue um vídeo de no máximo 3 minutos explicando o funcionamento dos seus programas e os resultados obtidos. No vídeo deve aparecer em algum momento o seu rosto e um documento seu. De preferência, envie link para vídeo e não o vídeo em si, para não sobrecarregar o meu HD (assegure que [hae.kim@usp.br](mailto:hae.kim@usp.br) tenha acesso).

**Obs. 4:** Envie os arquivos e/ou links via edisciplinas.

## **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 [<https://plainlanguage.gov/guidelines/>], 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. Quebre o texto em seções com títulos claros. Seja breve. 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 realidade). 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*

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*

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.