Universidade Federal do Ceará
Departamento de Computação
Disciplina: Inteligência Artificial
Prof. João Paulo do Vale Madeiro

Trabalho Final de Inteligência Artificial – Graduação

Objetivo:
Desenvolver uma aplicação utilizando Redes Neurais Convolucionais (CNN) para resolver um problema de classificação binária ou multiclasse, implementando e comparando versões arquiteturais e explorando fundamentos técnicos da modelagem e treinamento.

Estrutura obrigatória:

- **Dataset e definição do problema:** Explique e justifique o problema; apresente os dados com visualizações claras (exemplos); descreva o pré-processamento e divisão treino/validação/teste (com código explicado).
- **Arquitetura da CNN:** Apresente no mínimo duas versões diferentes (ex: simples vs. ajustada); explique detalhadamente a composição (número de filtros, kernel, pooling, ativação, normalização, etc.).
- **Treinamento e validação:** Relate os parâmetros utilizados (learning rate, epochs, batch size, etc.); exiba gráficos de acurácia e perda para treino e validação.
- **Avaliação:** Calcule acurácia, F1-score e matriz de confusão; mostre os gráficos de treino/validação e compare o desempenho entre as versões.
- **Experimento com hiperparâmetros:** Alterar um parâmetro relevante (ex: learning rate, número de camadas) e discutir o impacto.
- **Discussão técnica:** Apontar erros comuns, limitações e possíveis melhorias técnicas (não apenas aumento de dados ou camadas).

Observações:

- O código deve ser escrito em Python, organizado e comentado, utilizando notebooks (Google Colab ou Jupyter Notebook);
- A avaliação acontecerá por meio da apresentação oral do trabalho, incluindo explicações técnicas do treinamento, validação e testes;
- A equipe deve indicar como cada participante contribuiu para o projeto (máximo de 3 integrantes).

Ferramentas e Diretrizes Técnicas:
Para o desenvolvimento do trabalho, o aluno poderá basear-se no livro *Deep Learning with Python* (François Chollet) e nas aulas ministradas. Na implementação, deverá utilizar a

biblioteca **Keras**, podendo, se desejar, empregar uma abordagem de **transfer learning**, ou seja, utilizar modelos de CNN pré-treinados (como VGG16, ResNet, MobileNet, etc.) com pesos carregados do ImageNet, e treinar apenas as camadas finais customizadas para o novo problema.

Sugestões de datasets (opcional):

- `https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia`
- `https://www.kaggle.com/alxmamaev/flowers-recognition`
- `https://www.kaggle.com/kritikseth/fruit-and-vegetable-image-recognition`
- `https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000`
- `https://www.kaggle.com/jessicali9530/stanford-cars-dataset`
- `https://www.kaggle.com/gpiosenka/100-bird-species`
- `https://www.kaggle.com/c/diabetic-retinopathy-detection/data`