



CS50AI with Python

10.Traffic

Mateus Schwede

Problemática

- Escreva IA para identificar qual placa de trânsito aparece em fotografia (imagem);
- Em carros autônomos, um dos principais desafios é visão computacional;
- Visão computacional permite que carro compreenda ambiente ao seu redor a partir de imagens digitais;
- Parte importante disso é capacidade de reconhecer e distinguir diferentes tipos de placas de trânsito, como placas de pare, limite de velocidade, e de preferência;
- Utilizado conjunto de dados rotulado (imagens já categorizadas pelo tipo de placa) German Traffic Sign Recognition Benchmark (GTSRB);
 - GTSRB contém milhares de imagens;
 - Representa 43 diferentes tipos de placas de trânsito.
- Será utilizada a biblioteca TensorFlow;
- Rede neural será treinada para aprender classificar imagens de placas conforme exemplos fornecidos pelo GTSRB.

Instruções

- Baixe código de <https://cdn.cs50.net/ai/2023/x/projects/5/traffic.zip> e descompacte-o;
- Baixe descompacte o dataset de <https://cdn.cs50.net/ai/2023/x/projects/5/gtsrb.zip>;
- Mova o diretório gtsrb para dentro do diretório do projeto traffic;
- Dentro do diretório traffic, execute "pip3 install -r requirements.txt" para instalar dependências do projeto: opencv-python para processamento de imagens, scikit-learn para funções relacionadas a ML e tensorflow para redes neurais.

Funcionamento

- Dataset:
 - Diretório gtsrb contém 43 subpastas numeradas de 0 a 42;
 - Cada subpasta representa categoria diferente de placa de trânsito, com imagens do respectivo tipo de placa;
- traffic.py:
 - função main recebe argumentos da linha de comando:
 - Diretório contendo dados e, opcionalmente, arquivo para salvar modelo treinado.
 - Dados e rótulos são carregados da pasta especificada usando função load_data, sendo divididos em conjuntos de treino e teste;
 - Função get_model é chamada para obter rede neural compilada;
 - Modelo é treinado com dados de treino, e avaliado com dados de teste;
 - Se fornecido nome de arquivo, modelo treinado é salvo em disco;
 - Funções load_data e get_model precisam ser implementadas.

Especificações

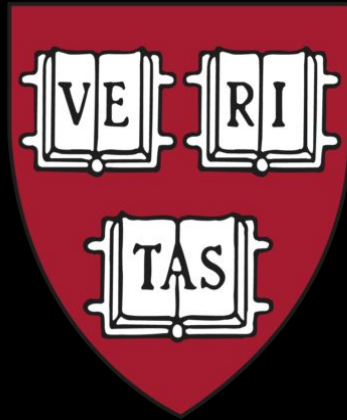
- Função `load_data`:
 - Recebe argumento diretório `data_dir` com dados armazenados, e retorna 2 listas (imagens e labels);
 - `data_dir` contém subpastas 0 até `NUM_CATEGORIES - 1`;
 - Cada subpasta representa categoria de placa de trânsito;
 - Cada subpasta contém vários arquivos de imagem;
 - Usar OpenCV (`cv2`) para ler cada imagem como `numpy.ndarray`;
 - Redimensionar todas imagens para largura `IMG_WIDTH` e altura `IMG_HEIGHT`, garantindo tamanho uniforme para rede neural;
 - Função retorna tupla(`images`, `labels`);
 - `images` é lista de arrays numpy com imagens redimensionadas;
 - `labels` é lista de inteiros, representando categoria de cada imagem.
 - Usar módulos `os.sep` e `os.path.join` para manipulação de caminhos, garantindo que código funcione em qualquer sistema operacional.

Especificações

- Função `get_model`:
 - Entrada da rede neural deve ter formato `(IMG_WIDTH, IMG_HEIGHT, 3)`, representando imagem colorida com 3 canais (RGB);
 - Camada de saída deve ter `NUM_CATEGORIES` unidades, uma para cada tipo de placa de trânsito;
 - Estrutura intermediária da rede é livre, podendo incluir:
 - Diferentes quantidades de camadas convolucionais e de pooling;
 - Variações no n^o e tamanho dos filtros nas camadas convolucionais;
 - Variações no tamanho do pooling;
 - Diferentes n^o s e tamanhos de camadas ocultas;
 - Uso de dropout para reduzir overfitting.
 - Criar arquivo separado chamado `README.md` para documentar processo de experimentação, contendo: o que foi testado, o que funcionou bem, o que não funcionou bem, e observações feitas durante testes.

Submissão

- Visual Studio Code online: <https://cs50.dev>
- Testar precisão da lógica do algoritmo: `check50 ai50/projects/2024/x/traffic`
- Testar estilização do código: `style50 traffic.py`
- Para submissão:
 - Em <https://submit.cs50.io/invites/d03c31aef1984c29b5e7b268c3a87b7b>, entre com GitHub e autorize CS50;
 - Instale pacote Git, Python 3 (e pip), instalando pacotes: `pip3 install style50 check50 submit50`
 - Submeta o projeto: `submit50 ai50/projects/2024/x/traffic`
- Verificar avaliação: <https://cs50.me/cs50ai>.



UB Social

Mateus Schwede

HBS ID 202400167108 - DCE ID @00963203