February, 2025

CSC_5IA23_TA

Deep Learning Based Computer Vision

Project: Fire Detection

JIN Pin, LIU Zihan, LIU Ziyi, SUI Xiaotong

In this project, we developed a wildfire detection system based on deep learning, utilizing multiple pre-trained neural network models for feature extraction and employing a Multi-Layer Perceptron (MLP) for classification training. The entire process includes data preprocessing, feature extraction, supervised learning training, and model evaluation.

1 Data Preprocessing

1.1 Dataset Acquisition

To ensure accessibility, consistency, and reproducibility, we utilize `kagglehub` to automate the dataset downloading process. This approach eliminates potential discrepancies caused by manual downloads and guarantees that all experiments are conducted on the same dataset version.

1.2 Dataset Splitting

To facilitate effective model training and evaluation, we partition the dataset as follows:

- The original `valid` dataset is carefully divided into a training set (**train_new**) and a validation set (**valid_new**) following an **80/20** ratio. This ensures a substantial amount of data is allocated for training while reserving a portion for performance validation.
- The test set (**test**) remains unaltered and serves as the final benchmark for assessing model generalization. Keeping this dataset unchanged guarantees a fair comparison across different model variations.

2 Data Augmentation

2.1 Augmentation Techniques

Data augmentation plays a critical role in improving model robustness and performance. By artificially increasing the training data diversity, augmentation helps mitigate overfitting and enhances the model's ability to generalize to unseen data. We apply the following widely-used augmentation techniques:

- **Random Resized Cropping**: Adjusts the image size while randomly cropping different portions to improve spatial invariance.

- **Random Horizontal Flipping**: Flips the image along the horizontal axis with a given probability, making the model more resilient to left-right variations.
- **Random Rotation**: Rotates the image within a certain range to simulate different viewing angles and perspectives.
- **Color Jittering**: Modifies brightness, contrast, saturation, and hue to account for different lighting conditions, ensuring better adaptability.

2.2 Rationale for Selection

The selected augmentation techniques are carefully chosen to enhance the model's generalization capacity and improve robustness against variations.

- **Improving model generalization:** These techniques help the model adapt to variations in angles, lighting conditions, and wildfire scales, ensuring better performance on real-world scenarios.
- **Mitigating overfitting:** By artificially expanding the training dataset, augmentation reduces dependency on specific features and prevents the model from memorizing patterns instead of learning meaningful representations.

3 Transfer Learning and Feature Extraction

3.1 Methods

To leverage the advanced feature extraction capabilities of pretrained models, we employ several widely recognized Convolutional Neural Network (CNN) and Vision Transformer (ViT) architectures:

- **ResNet** [2]: A deep neural network incorporating residual connections to effectively mitigate the vanishing gradient problem while maintaining computational efficiency.
- **MobileNet** [3]: A lightweight neural network designed for resource-constrained environments, significantly reducing computational cost while preserving high accuracy.
- **EfficientNet** [4]: Optimized using a compound scaling strategy, this model strikes an optimal balance between computational efficiency and predictive performance.
- **Vision Transformer (ViT)** [1]: A self-attention-based model capable of capturing long-range dependencies more effectively than traditional CNNs, enabling superior performance in complex vision tasks.

3.2 Reasons for Choosing This Approach

These models collectively demonstrate efficient and flexible feature extraction capabilities, and they are able to achieve excellent classification performance while maintaining low computational cost, capturing both fine-grained local features and modeling long-range dependent information, thus enabling high-precision and high-performance image analysis under different hardware conditions. Using these models to extract features can provide a rich and sufficient selection of features for further subsequent classification.

Since the early layers of pretrained models have already learned to extract general image features, in the training that followed, we will freeze their weights, preventing their parameters from updating during training. This helps to avoid overfitting, reduces the number of trainable parameters, and improves training efficiency.

Additionally, as the final layer of pretrained models is designed for **ImageNet's 1000-class classification task**, but our project involves a **binary classification problem (Wildfire vs. No Wildfire)**, we **remove the final classification head** and replace it with a new **fully connected layer (FC)** for binary classification. In this way, we can effectively leverage existing deep learning architectures, enhancing the wildfire detection model's performance while reducing training time and computational costs.

4 Supervised Learning and MLP Classification

4.1 Methods

Following feature extraction, a **Multi-Layer Perceptron (MLP)** is employed for classification. This approach effectively leverages high-dimensional feature vectors extracted from pretrained CNN or ViT models to perform binary classification. The detailed process is outlined as follows:

- **Feature Vector Input**:
  - After processing input images, the pretrained CNN or ViT model outputs a **fixed-length high-dimensional feature vector**, which serves as the input for the MLP classifier.
- **MLP Architecture**:
  - The MLP consists of **two hidden layers** with **512 and 256 neurons**, respectively. Each hidden layer applies the **ReLU activation function** to introduce non-linearity and enhance model expressiveness.
  - The final output layer is a **fully connected (FC) layer** with a **single neuron**, employing the **Sigmoid activation function** to facilitate binary classification.
- **Loss Function and Optimization**:
  - The model is trained using **Binary Cross-Entropy (BCE) Loss**, ensuring that the predicted probabilities align accurately with the wildfire/non-wildfire classes.
  - The **Adam optimizer** is utilized for parameter updates, with an **initial learning rate set to 0.001**, balancing convergence speed and stability.
- **Mini-Batch Training and Overfitting Prevention**:
  - The training process employs a mini-batch strategy, with a batch size of 32, ensuring stable parameter updates while maximizing GPU acceleration.
  - The model is trained for 50 epochs, incorporating early stopping to mitigate overfitting and improve generalization.

4.2 Reasons for Choosing This Approach

- **Decoupling Feature Extraction and Classification Tasks**: Employing an MLP for classification instead of training a deep CNN end-to-end significantly reduces computational complexity while maintaining effective classification capabilities.
- **Efficient Adaptation to Binary Classification**:
  - Since wildfire detection is a **binary classification task (Wildfire vs. No Wildfire)**, the MLP's output layer incorporates the **Sigmoid activation function**, producing probability values between **0 and 1**, indicating the likelihood of an image containing a wildfire.
  - By defining an appropriate **classification threshold**, the model can effectively distinguish between wildfire and non-wildfire images, improving its practical applicability in real-world scenarios.

References

[1] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[2] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[3] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[4] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.