Final Project
Bwt ML-DL Track Batch 3

HybridNet: Multi-Label Image Classification with CNN-RNN

Project Overview

In real-world scenarios, images often contain multiple objects, scenes, actions, and attributes, requiring models to assign multiple labels to each image. While Convolutional Neural Networks (CNNs) have proven effective in single-label image classification tasks, traditional multi-label classification approaches tend to struggle with the inherent relationships between labels.

This project introduces **HybridNet**, a unified CNN-RNN framework that combines the strengths of CNNs for feature extraction and Recurrent Neural Networks (RNNs) for modeling label dependencies. By using this architecture, the model can effectively capture both the visual content of images and the relationships between multiple labels, resulting in more accurate multi-label image classification.

Key Features

- **CNN for Feature Extraction:** The CNN component extracts high-level features from input images, focusing on visual patterns and object characteristics.
- **RNN for Label Dependencies:** The RNN component models the relationships and dependencies between multiple labels, helping the system make more informed predictions.
- **Joint Image-Label Embedding:** The framework integrates image and label information seamlessly, enabling more robust multi-label classification.
- **End-to-End Training:** HybridNet allows for end-to-end training of both CNN and RNN components, optimizing the model for multi-label tasks.
- **Benchmark Performance:** Experimental results demonstrate that HybridNet achieves superior performance compared to existing models for multi-label image classification on benchmark datasets.

Requirements

To run this project, you need the following dependencies:

- **Python**: Version 3.6 or higher
- **PyTorch**: Version 1.6.0 or higher
- **torchvision**: Version 0.7.0 or higher
- **numpy**: Version 1.18.1 or higher
- **scikit-image**: Version 0.16.2 or higher
- **scipy**: Version 1.5.2 or higher
- **nltk**: Version 3.4.5 or higher

Installation

To install the required dependencies, simply run the following command:

```
pip install -r requirements.txt
```

Dataset

This project uses the **COCO dataset**, which consists of **82,081** training images and **40,137** validation images. Each image in the dataset is annotated with multiple labels from a set of **80** categories, making it a suitable benchmark for multi-label classification tasks.

You can download the COCO dataset from the official website. Ensure that the dataset is properly preprocessed and stored in the appropriate directories within your project.

- **Images Directory:** Store the images in data/images/.
- **Annotations Directory:** Store the corresponding annotations in data/annotations/.

Model Architecture

The HybridNet model architecture is composed of two primary components:

- **Convolutional Neural Network (CNN):** The CNN component processes the input image, extracting visual features such as edges, textures, and patterns. These features form a high-dimensional representation of the image, which is then passed to the RNN.
- **Recurrent Neural Network (RNN):** The RNN component models the dependencies between the different labels of the image. By learning a joint image-label embedding, the RNN helps to capture the semantic relationships between the labels and the content of the image.

This architecture enables the model to understand the relevance of each label concerning the image and other labels, significantly improving the accuracy of multi-label predictions.

The implementation of this architecture can be found in the model.py file.

Training the Model

To train the model, use the train.py script. The script allows for customizable hyperparameters, such as batch size, learning rate, and the number of epochs. You can specify these parameters through command-line arguments.

Example Training Command

```
python train.py --image_path data/images/img.jpg --vocab_path data/vocab.pkl --batch_size 256 --learning_rate 0.001 --num_epochs 10 --num_workers 4
```

This command will initiate the training process with a batch size of 256, a learning rate of 0.001, and a total of 10 epochs. The model will use the image data from the specified path and corresponding vocabulary.

Training Process

During training, the CNN will extract features from each image, while the RNN will learn the relationships between the labels. The model will be optimized end-to-end, adjusting weights in both the CNN and RNN layers based on the error gradients.

Evaluating the Model

After the model has been trained, you can evaluate its performance on the validation set. .

Results

HybridNet's performance on the COCO dataset demonstrates its ability to effectively model label dependencies and improve multi-label classification accuracy. The evaluation metrics provide a detailed view of the model's performance, with improvements over traditional methods in capturing multiple objects and actions in a single image.

How to Run the Project

To run the project, follow these steps:

1. Clone the Repository

```
git clone https://github.com/muhammadzain21/HybridNet.git
cd HybridNet
```

2. Prepare the Dataset

Download and preprocess the COCO dataset as outlined in the Dataset section. Ensure that images and annotations are placed in the correct directories.

3. Train the Model

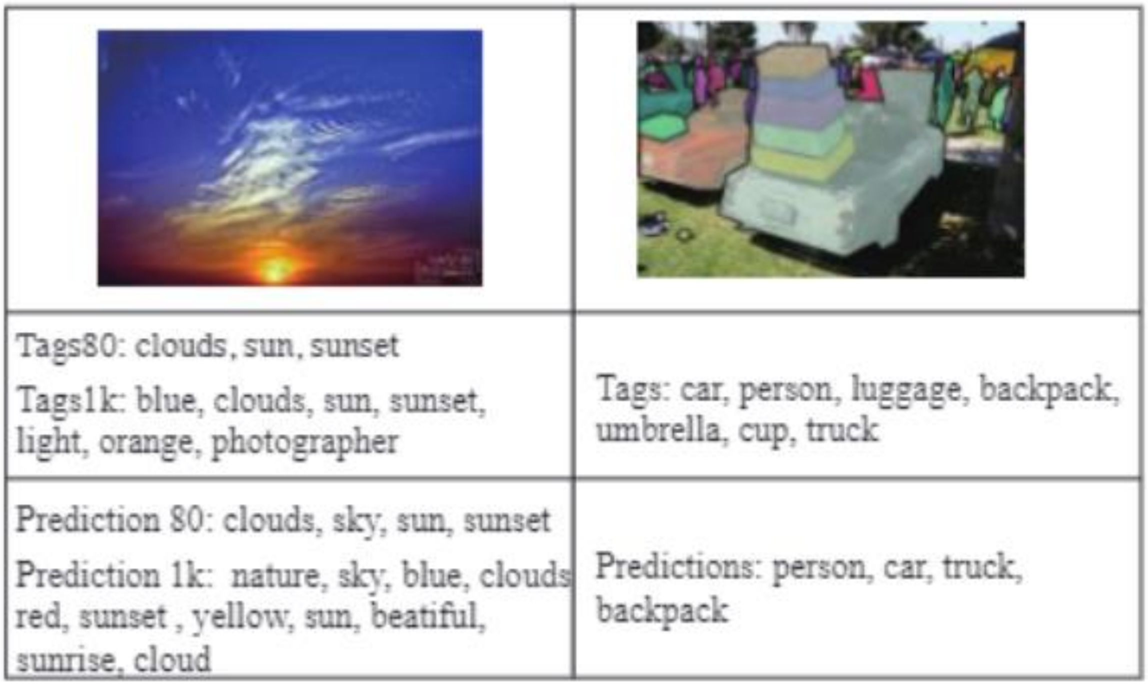
Use the following command to start training the model:

```
python train.py --image_path data/images/img.jpg --vocab_path data/vocab.pkl --batch_size 256 --learning_rate 0.001 --num_epochs 10 --num_workers 4
```

Adjust the parameters as needed for your hardware and training configuration.

4. Evaluate the Model

After training, you can evaluate the model on the validation set using the evaluation script. The results will be displayed on the console and saved to a log file.



Tags80: clouds, sun, sunset Tags1k: blue, clouds, sun, sunset, light, orange, photographer	Tags: car, person, luggage, backpack, umbrella, cup, truck
Prediction 80: clouds, sky, sun, sunset Prediction 1k: nature, sky, blue, clouds red, sunset , yellow, sun, beatiful, sunrise, cloud	Predictions: person, car, truck, backpack