Scalable Video Processing Using Spark

Siddhant Garg & Sridhama Prakhya

COMPSCI 532: Systems for Data Science
Content

- Motivation and Introduction
- Components
  - Spark
  - CLIP
  - MongoDB
- Experiments and Results
- Conclusion and Future Work
Motivation: Raw video datasets
Motivation

- **Video Processing**
  - Frame sampling
    - 9K frames for 5-minute video @ 30 FPS
  - Machine Learning inference
    - Batch inference
  - Takes too long for large datasets
    - 1 month for 500K videos on a single machine

- **Analysis**
  - Retrieve videos from specific classes
  - Retrieve similar videos based on shared features
Introduction

- **Parallelize video processing using Spark**
  - Automatically partition data across machines
  - Run feature extraction on each machine in parallel
  - Machine Learning model
    - CLIP model
    - ImageNet classes (1K)
    - Kinetics (700 action classes)

- **MongoDB database**
  - Storing extracted tags for each video
  - Efficient and simple for analyzing extracted information
Overview

Spark Partitions

Parallel Video Processors

Frame Sampling + ML Analyzer

Frame Sampling + ML Analyzer

Frame Sampling + ML Analyzer

Useful Information

MongoDB
Database storage

- **MongoDB**
  - NoSQL database
    - Horizontal scalability
    - Fast retrieval for semi-structured data (tag arrays)

- **SparkSQL for insertion and querying**
  - Simple retrieval query based on single tag
  - Conditional retrieval using multiple tags
Frame Sampling
+
ML Analyzer
Video Processing

Duration: 4 mins
Frame rate: 30 FPS

TOTAL FRAMES: 7200

Source: https://www.youtube.com/watch?v=u3dqiq0uFdg
Video Processing

Total Frames: 7200

Frame Sampling @ 1 FPS

Sampled Frames: 240
Video Processing

Sample Frames: 240

[jeep, parkour, shooting off fireworks, freight car, bulldozing]
Frame Sampling
+
ML Analyzer
Machine Learning analyzer

- CLIP model
  - Text Transformer encoder
  - Image Transformer encoder
  - Trained using natural language supervision
  - Support zero-shot predictions

CLIP Zero-Shot Prediction

A photo of a {object}

Text Encoder

Image Encoder

\( P_1 \)
\( P_2 \)
\( P_3 \)
\( \ldots \)
\( P_N \)

\( I \)
\( I \cdot P_1 \)
\( I \cdot P_2 \)
\( I \cdot P_3 \)
\( \ldots \)
\( I \cdot P_N \)

A photo of a dog.
Experiments and Results
Inference Durations

Inference time by varying number of workers and videos

Number of videos

Inference time (minutes)

1 worker

# GPU : 1 Tesla T4, 15GB
Inference Durations

Inference time by varying number of workers and videos

- **1 worker**
- **2 workers**

Number of videos:

- 100
- 200
- 300
- 400
- 500

Inference time (minutes):

- 0
- 20
- 40
- 60
- 80
- 100

**# GPU:** 1 Tesla T4, 15GB
Inference Durations

Inference time by varying number of workers and videos

- 1 worker
- 2 workers
- 3 workers

Number of videos vs. Inference time (minutes)

# GPU: 1 Tesla T4, 15GB
Inference Durations

Inference time by varying number of workers and videos

- 1 worker
- 2 workers
- 3 workers
- 4 workers

Number of videos

Inference time (minutes)

# GPU : 1 Tesla T4, 15GB
Query Results for ‘playing cello’ (video-1)

Source: https://www.youtube.com/watch?v=j7MZfgP3j7k
Query Results for ‘playing cello’ (video-2)

Cello/Playing cello

Cello/Playing cello

Bartending

Source: https://www.youtube.com/watch?v=Fzlh69nLx5Y
Thank you