Data Processing

Tong He
Applied Scientist, Amazon Web Services.
2018.12.18

Data Collection

Data Collection

- Format
- Label
- Directory Structure

Data Collection

- Format: Task Specific
    - Classification: the label
    - Detection: the boxes, and labels
    - Segmentation: the masks, and labels

Data Collection

- Labeling
    - Manual
        - Accurate, Expensive
    - Automatic
        - Somewhat accurate, cheap
        - SageMaker Ground Truth

Data Collection

- Labeling

Data Collection

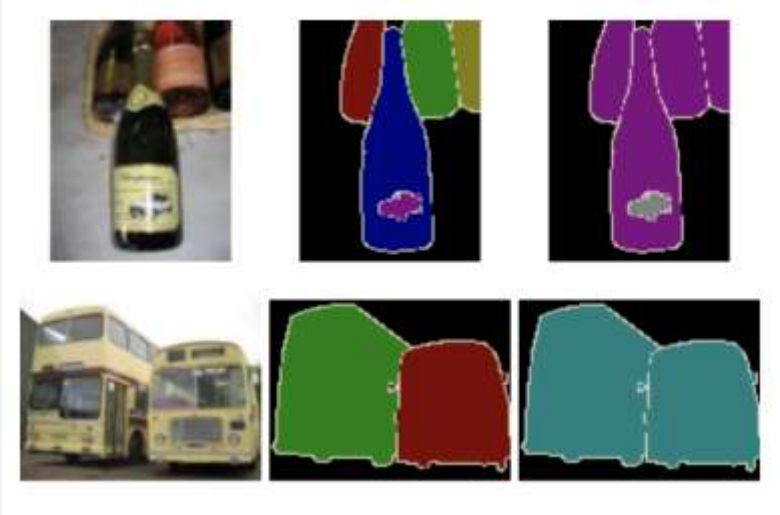
- Classification Directory Structure
    - ImageNet-Train/
        - Cat/
            - 001.jpg
            - 002.jpg
            - …
        - Dog/
        - …

Data Collection

- Detection Directory Structure
    - Pascal VOC/
        - Images
            - 001.jpg
            - 002.jpg
            - …
        - Annotation
            - 001.xml
            - …

Data Collection

- Segmentation Directory Structure
  - Pascal VOC/
    - Images
      - 001.jpg
    - Object Segmentation
      - 001.jpg
    - Class Segmentation
      - 001.jpg



aws

Data Loading with GluonCV

Data Loading with GluonCV

- GluonCV Interface
    - DataSet
        - Input: images, labels
        - Output: Arrays of images and labels in memory
    - Transformation
        - Data augmentation
    - DataLoader
        - Scheduling
        - Multi-threading

Data Loading with GluonCV

- DataSet
    - Task/Structure Dependent
        - Preset functions for certain structures
    - Very flexible
        - Class VisionDataset()
        - Users can override the class

Data Loading with GluonCV

- Transformation
  - Augmentation
    - Abundant choices
  - Flexible interface
    - Stack in sequence

```
transform_train = transforms.Compose([
    transforms.RandomResizedCrop(input_size),
    transforms.RandomFlipLeftRight(),
    transforms.RandomColorJitter(brightness=jitter_param, contrast=jitter_param,
                                 saturation=jitter_param),
    transforms.RandomLighting(lighting_param),
    transforms.ToTensor(),
    normalize
])
```


aws

Data Loading with GluonCV

- DataLoader
    - Load Schedule
        - Pool of threads
        - Pre-fetch
    - Training/Testing specific
        - Data Shuffling
        - Batch size

Data Loading with GluonCV

- GluonCV Interface
    - Pipeline
        - File -> Dataset -> Transformation -> DataLoader

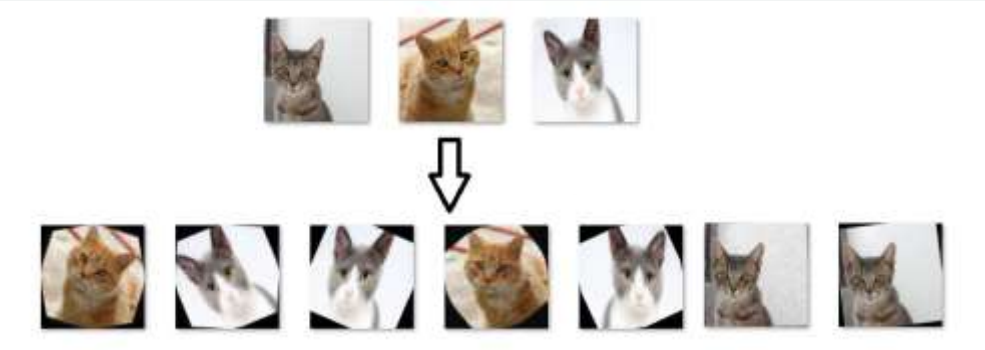
Data Transformation with GluonCV

Data Transformation with GluonCV

- Why Transformation?
  - Resize to fit model
  - Prevent overfitting
  - Enrich the dataset

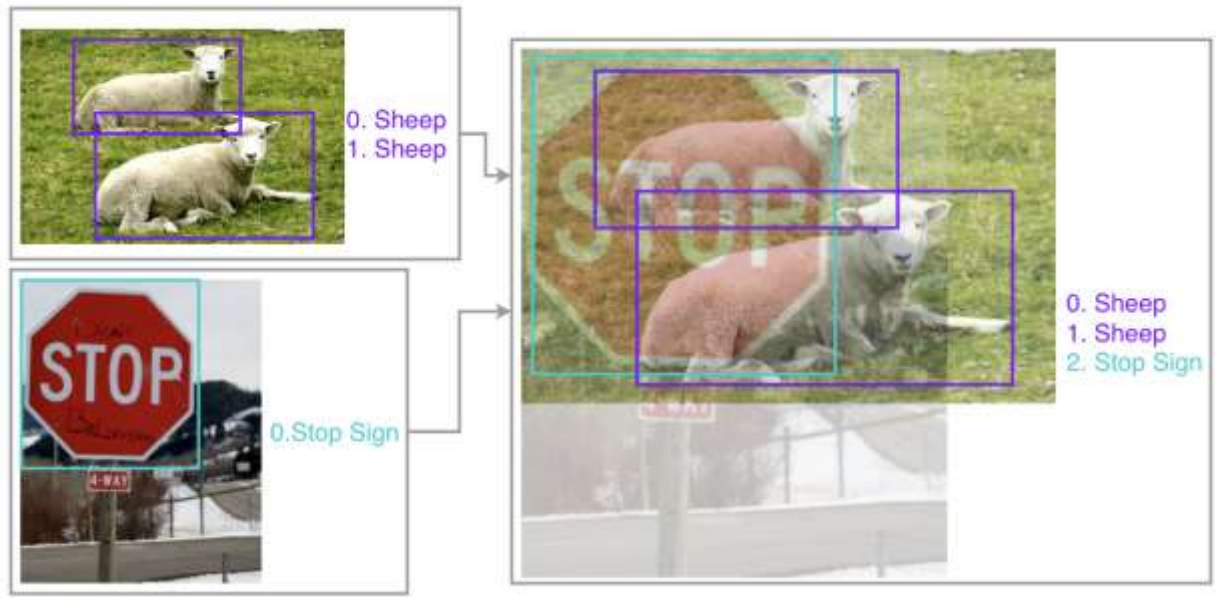
Data Transformation with GluonCV

- Popular Transformation
  - Resize
  - Crop
  - Flip
  - Rotation
  - Adding Noise
  - Normalization



Data Transformation with GluonCV

- Advanced Transformation
  - Mix-Up



Data Transformation with GluonCV

- Transformation for Inference
    - Crop
    - Normalization
    - No Randomization

Fast IO in GluonCV

Fast IO in GluonCV

- Hardware
    - RAM Disk > SSD >> HDD
        - ImageNet dataset: 140GB
        - RAM of p3.16xlarge: 768GB

Fast IO in GluonCV

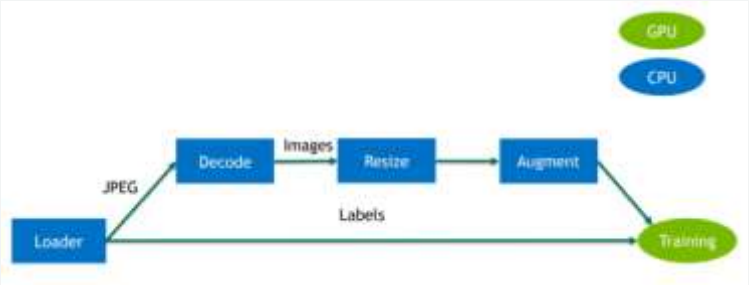
- Image Format: Raw Image
    - Support any kind of tasks
    - Read through DataLoader
    - Slow
    - Need to unzip on each new machine

Fast IO in GluonCV

- Image Format: RecordIO
    - Support classification and detection
    - Read through DataLoader or ImageRecordIter
    - Fast
    - One-time packing

Fast IO in GluonCV

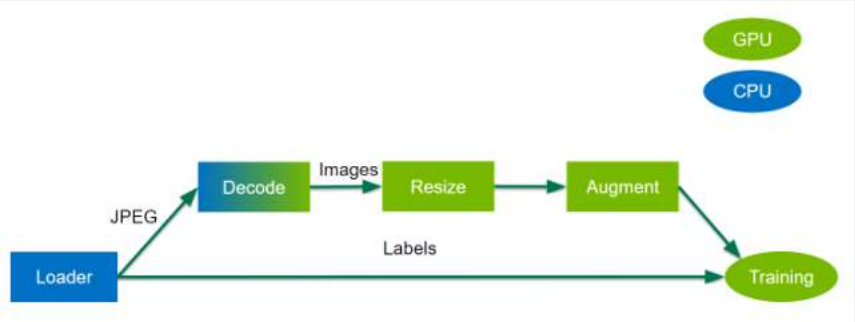
- Interface: ImageRecordIter
  - One function call for
    - DataSet
    - Transform
    - DataLoader
  - Less flexible
  - Very Fast



aws

Fast IO in GluonCV

- Interface: Nvidia DALI (with nvJPEG)
  - Combination of
    - DataSet
    - Transform
    - DataLoader
  - Flexible
  - Extremely Fast
  - In-Development



aws