GluonCV

Deep Learning Toolkit for Computer Vision

https://gluon-cv.mxnet.io/

Muhyun Kim, AI Labs
TeaHyun Kim, ML AoD, Solutions Architect

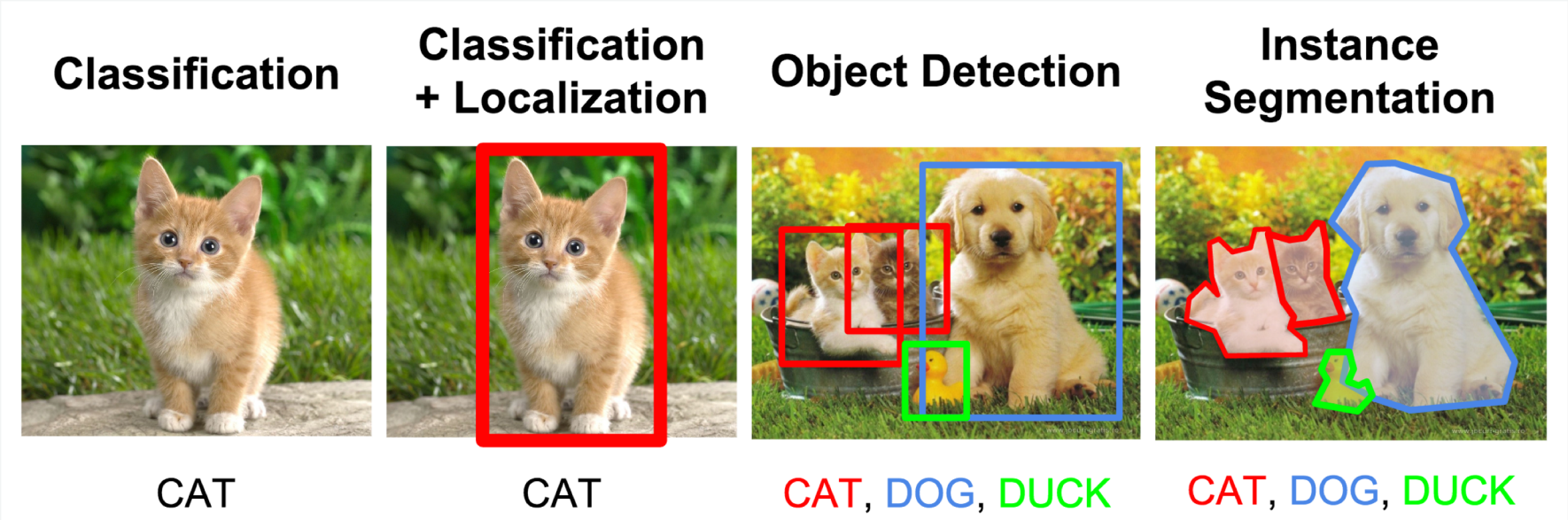
August 13, 2018

Agenda

- Computer Vision
- GluonCV
- Data Augmentation
- Transfer Leaning
- Data Preparation using RecordIO

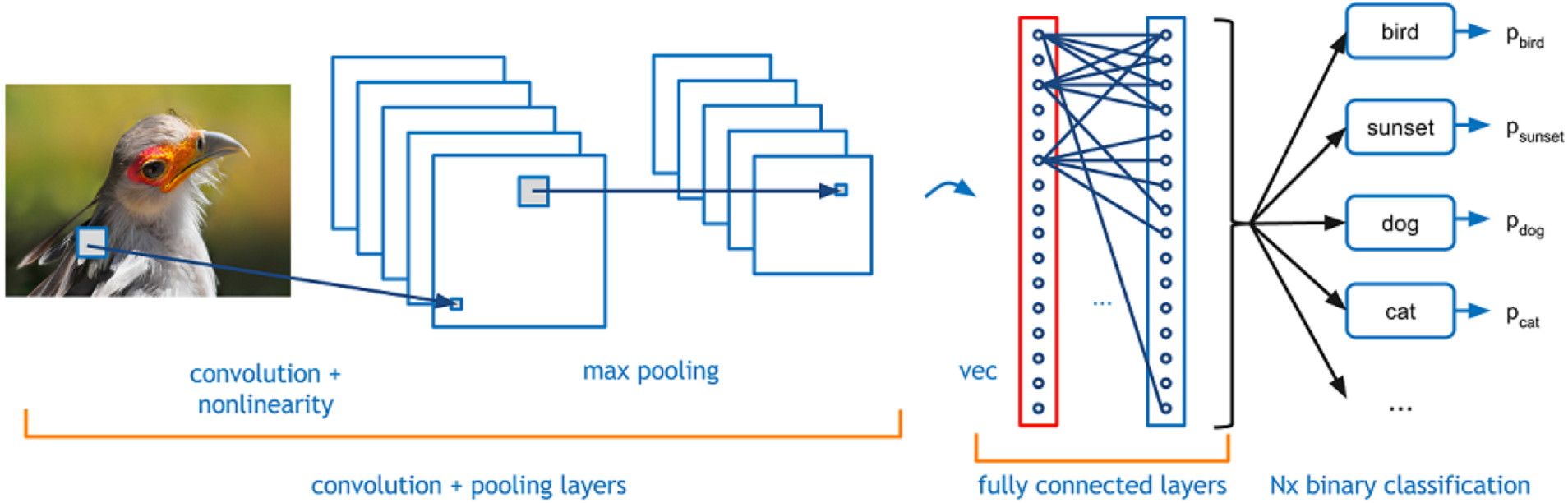
© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Computer Vision Problems (NOT ALL)



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Convolutional Neural Network for Image Classification



Source: https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

What needed to solve computer vision problems

Training Data

ML Algorithm

Image Credit : http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/

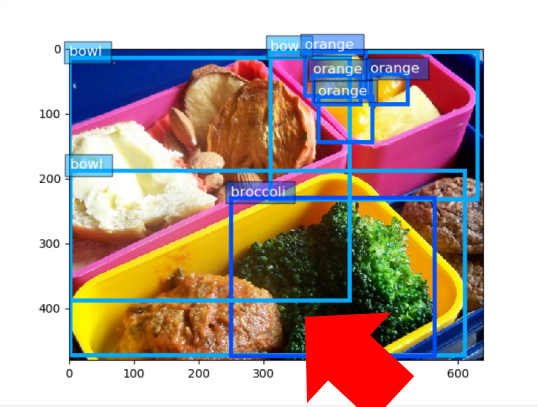
© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Dataset for Supervised Learning for each CV Task

Classification

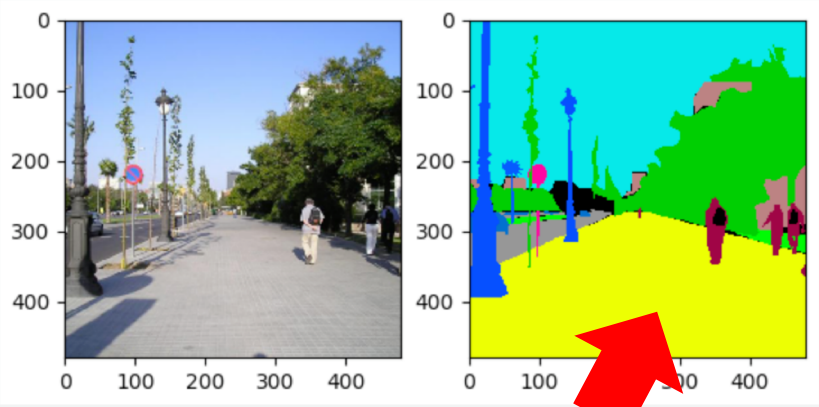
Class

Object Detection



Object Name with Bounding Box

Segmentation



Pixel level segmentation

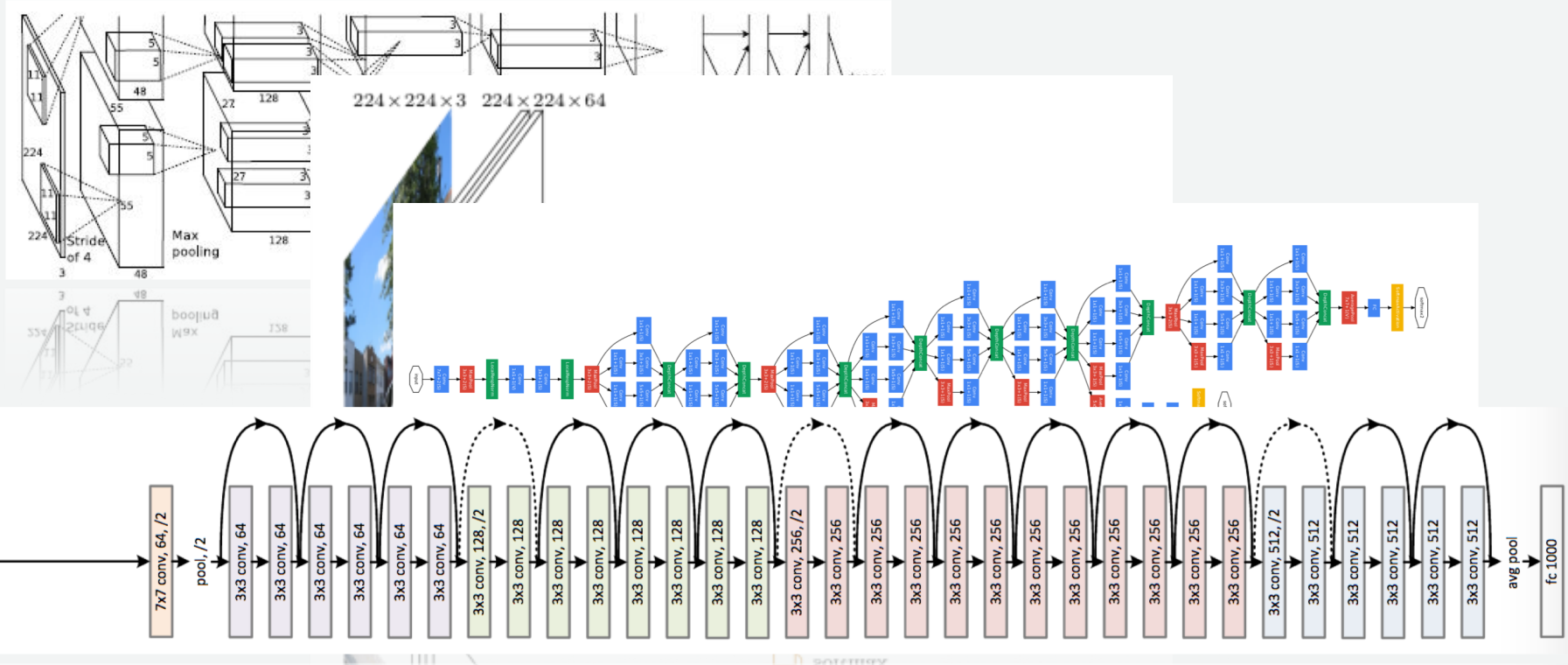
© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Public Vision Datasets

- **ADE20K** : A scene-centric containing 20 thousands images annotated with 150 object categories
- **COCO** : Large-scale object detection, segmentation, and captioning dataset
- **PASCAL VOC** : A Collection of datasets for object detection
- **ImageNet** : Millions of images and thousands of objects for image classification

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

ML Algorithms. You are not alone



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

But …

- Re-implementation of papers with the same accuracy is not easy task

- There are open source implementation, but
  - Quality of open source or 3rd party implementation of papers varies a lot
  - A lot of details vary from one implementation to another
  - Implementations focus on a single task applying only to a particular data set
  - Stopped support on the code

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Introducing GluonCV

Our ML Experts created a Computer Vision toolkit using Gluon to solve these problems

for

Experienced users and Newcomers to CV field

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Gluon CV Toolkit

https://gluon-cv.mxnet.io/index.html

GluonCV provides implementations of the state-of-the-art (SOTA) deep learning models in computer vision.

It is designed for engineers, researchers, and students to fast prototype products and research ideas based on these models. This toolkit offers four main features:

1. Training scripts to reproduce SOTA results reported in research papers
2. A large number of pre-trained models
3. Carefully designed APIs that greatly reduce the implementation complexity
4. Community supports

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Gluon CV Toolkit – Model Zoo

https://gluon-cv.mxnet.io/model_zoo/index.html

Image Classification – CIFAR, ImageNet

- ResNet V1, V2
- WideResNet

Object Detection

- SSD
- Faster RCNN

Semantic Segmentation

- FCN
- PSPNet

```
from gluoncv import model_zoo

# load a ResNet model trained on CIFAR10
cifar_resnet20 = model_zoo.get_model('cifar_resnet20_v1', pretrained=True)

# load a pre-trained ssd model
ssd0 = model_zoo.get_model('ssd_300_vgg16_atrous_voc', pretrained=True)

# load ssd model with pre-trained feature extractors
ssd1 = model_zoo.get_model('ssd_512_vgg16_atrous_voc', pretrained_base=True)

# load ssd model without initialization
ssd2 = model_zoo.get_model('ssd_512_resnet50_v1_voc', pretrained_base=False)

cifar_resnet20 = model_zoo.cifar_resnet20_v1(pretrained=True)
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

What’s included

1. Reimplementation of important papers in recent years
2. Detailed documentation and well-documented examples
3. Pre-trained models that can be used directly
4. Performance metrics, to help choose between different models
5. Consistent interface to reduce the barrier to entry when switching from one model to another
6. Regular re-training and continuous integration to ensure code correctness

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Let’s Install – On Your Own PC

1. Install mxnet, gluoncv and jupyter

```
pip install --upgrade pip
pip install --pre mxnet
pip install gluoncv
pip install jupyter
```

2. Run jupyter notebook

```
jupyter notebook
```

3. Create a new notebook by “New” > ”Python*”

4. Verify if GluonCV is installed by importing it

```
import gluoncv
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Let’s Install – On SageMaker

1. Create Notebook instance, and Open it
2. Create a new notebook by “New” > ”conda_mxnet_p36”
3. Execute the commands in a Code cell

```
!pip install --pre mxnet
!pip install gluoncv
```

4. Verify if GluonCV is installed by importing it

```
import gluoncv
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Your 1st Deep Learning Computer Vision Task

1. Find any street image on Internet, and Upload it to your notebook
2. Put the below code into two cells and Run them

jupyter Untitled1 Last Checkpoint: a minute ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted conda_mxnet_p36

Run Code

In [ ]:

```
from gluoncv import model_zoo, data, utils
net = model_zoo.get_model('ssd_512_resnet50_v1_voc',pretrained=True)
x, img = data.transforms.presets.ssd.load_test('street.png', short=512)
class_IDs, scores, bounding_boxs = net(x)
```

In [ ]:

```
utils.viz.plot_bbox(img, bounding_boxs[0],scores[0], class_IDs[0], class_names=net.classes)
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Closer Look at the Code

```
from gluoncv import model_zoo, data, utils
```

importing necessary modules

```
net = model_zoo.get_model('ssd_512_resnet50_v1_voc',pretrained=True)

x, img = data.transforms.presets.ssd.load_test('street.jpg', short=512)

class_IDs, scores, bounding_boxs = net(x)

utils.viz.plot_bbox(img, bounding_boxs[0], scores[0], class_IDs[0], class_names=net.classes)
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Closer Look at the Code

```
from gluoncv import model_zoo, data, utils
```

importing necessary modules

```
net = model_zoo.get_model('ssd_512_resnet50_v1_voc', pretrained=True)
```

Loading pre-trained Object Detection Model

```
x, img = data.transforms.presets.ssd.load_test('street.jpg', short=512)

class_IDs, scores, bounding_boxs = net(x)

utils.viz.plot_bbox(img, bounding_boxs[0], scores[0], class_IDs[0], class_names=net.classes)
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Closer Look at the Code

```
from gluoncv import model_zoo, data, utils
```

importing necessary modules

```
net = model_zoo.get_model('ssd_512_resnet50_v1_voc', pretrained=True)
```

Loading pre-trained Object Detection Model

```
x, img = data.transforms.presets.ssd.load_test('street.jpg', short=512)
```

Loading an image, and Transforming it to tensor by applying normalizations

```
class_IDs, scores, bounding_boxs = net(x)
```

```
utils.viz.plot_bbox(img, bounding_boxs[0], scores[0], class_IDs[0], class_names=net.classes)
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Closer Look at the Code

```
from gluoncv import model_zoo, data, utils
```

importing necessary modules

```
net = model_zoo.get_model('ssd_512_resnet50_v1_voc', pretrained=True)
```

Loading pre-trained Object Detection Model

```
x, img = data.transforms.presets.ssd.load_test('street.jpg', short=512)
```

Loading an image, and Transforming it to tensor by applying normalizations

```
class_IDs, scores, bounding_boxs = net(x)
```

Inferencing on the image

```
utils.viz.plot_bbox(img, bounding_boxs[0], scores[0], class_IDs[0], class_names=net.classes)
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Closer Look at the Code

```
from gluoncv import model_zoo, data, utils
```

importing necessary modules

```
net = model_zoo.get_model('ssd_512_resnet50_v1_voc', pretrained=True)
```

Loading pre-trained Object Detection Model

```
x, img = data.transforms.presets.ssd.load_test('street.jpg', short=512)
```

Loading an image, and Transforming it to tensor by applying normalizations

```
class_IDs, scores, bounding_boxs = net(x)
```

Inferencing on the image

```
utils.viz.plot_bbox(img, bounding_boxs[0], scores[0], class_IDs[0], class_names=net.classes)
```

Visualizing the result with bounding box

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

How to load a model

- Loading a model by name

```
gluoncv.model_zoo.get_model
gluoncv.model_zoo.get_[model name]
```

- Loading a model directly from Gluon Model Zoo

```
from mxnet.gluon.model_zoo import vision
resnet18 = vision.resnet18_v1()
```

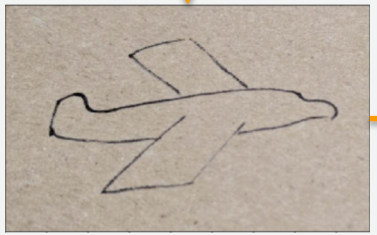
© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

How to load images for inference

plan-draw.jpg

```
img = image.imread(im_fname)
```

IMAGE FILE READ



(548,882,3)

TRANSFORM

(3,32,32)

```
img.expand_dims(axis=0)
```

DIMENSION EXPANSION

(1,3,32,32)

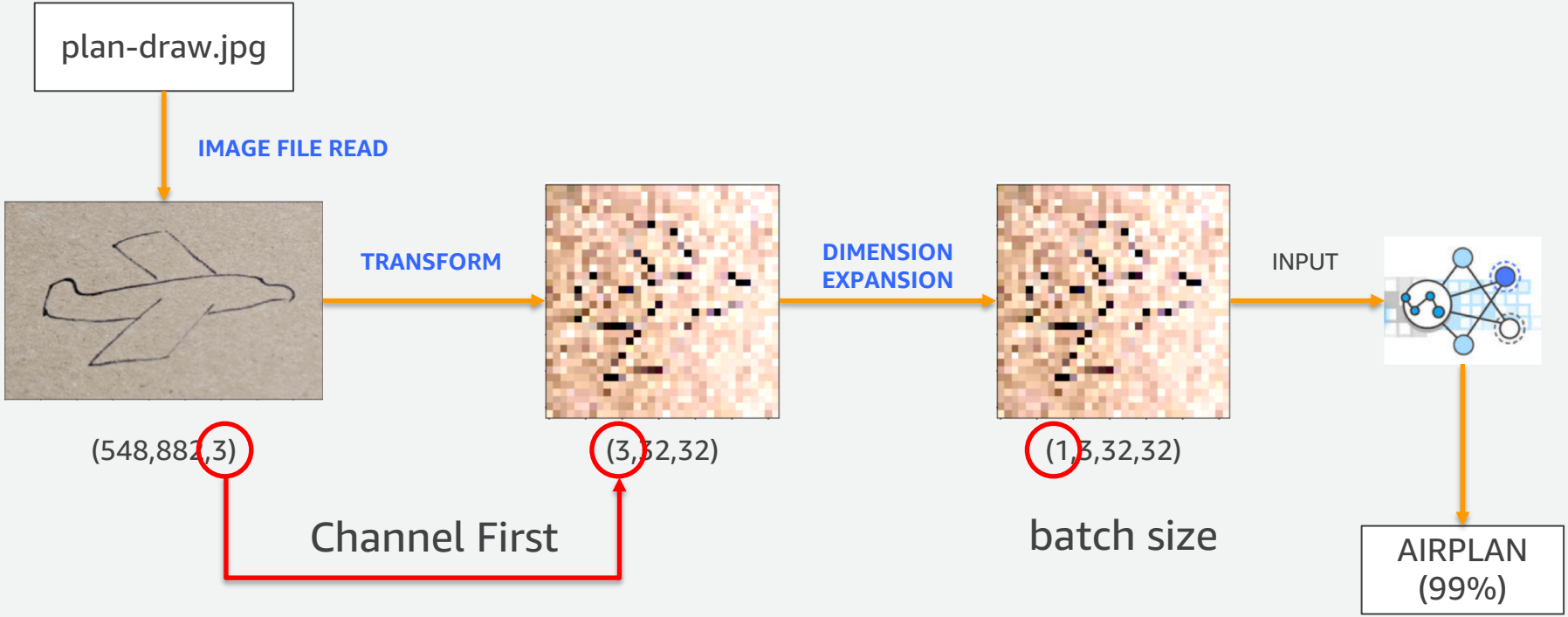
INPUT

AIRPLAN (99%)

```
transform_fn = transforms.Compose([
    transforms.Resize(32),
    transforms.CenterCrop(32),
    transforms.ToTensor(),
    transforms.Normalize(
        [0.4914, 0.4822, 0.4465], [0.2023, 0.1994, 0.2010])
])
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

How to load images for inference



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

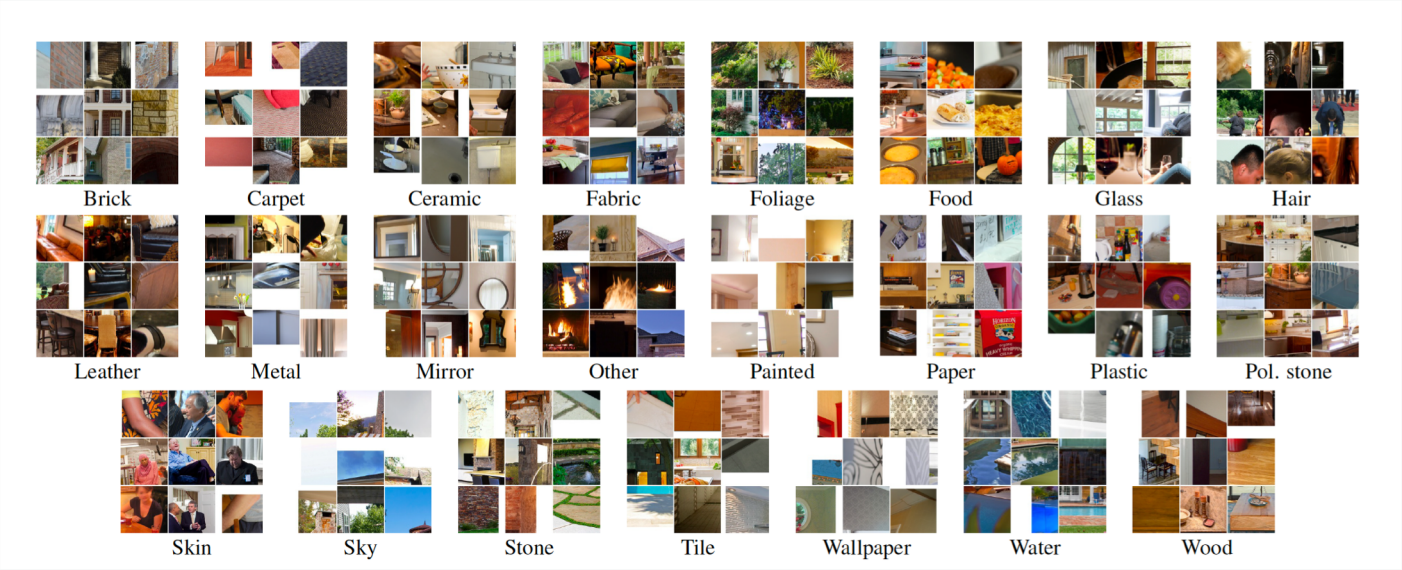
Lab 1: Inference on Image

```
!wget https://gluon-cv.mxnet.io/_downloads/demo_cifar10.ipynb
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

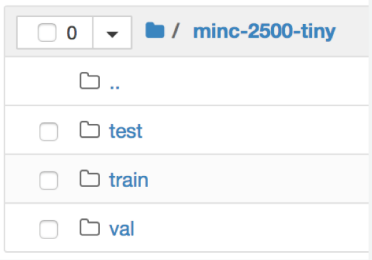
You have your own image dataset for classification

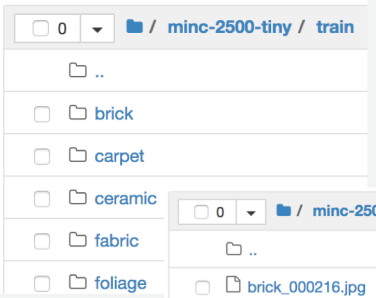
- MINC (Materials in Context Database) by Cornell
- MINC-2500 with 23 classes and 2500 images

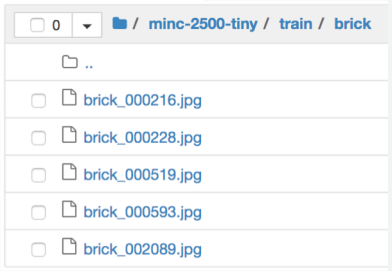


© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

How to load Training Data using DataLoader



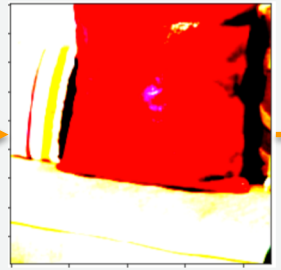




FILES

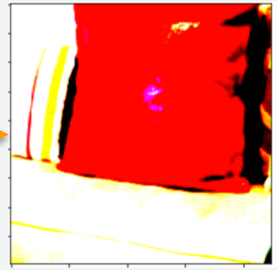
N * (362,362,3)

DataLoader



(N, 3, 224, 224)

Mini-Batch



(B, 3, 224, 224)

```
transform_train = transforms.Compose([
    transforms.RandomResizedCrop(224),
    transforms.RandomFlipLeftRight(),
    transforms.RandomColorJitter(brightness=jitter_param, contrast=jitter_param,
                                 saturation=jitter_param),
    transforms.RandomLighting(lighting_param),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])

train_data = gluon.data.DataLoader(
 gluon.data.vision.ImageFolderDataset(train_path).transform_first(transform_train),
 batch_size=batch_size, shuffle=True, num_workers=num_workers)
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Lab 2: Image Dataset for Training

```
!wget https://gluon-cv.mxnet.io/_downloads/transfer_learning_minc.ipynb
```

If you don't have a GPU, set num_gpus from 1 to 0
Set batch_size = 32

Finish up through "**Data Augmentation**" cell
Take a look at directories and files under *./minc-2500-tiny*

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Quiz

```
jitter_param = 0.4
lighting_param = 0.1

transform_train = transforms.Compose([
    transforms.RandomResizedCrop(224),
    transforms.RandomFlipLeftRight(),
    transforms.RandomColorJitter(brightness=jitter_param, contrast=jitter_param,
                                 saturation=jitter_param),
    transforms.RandomLighting(lighting_param),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])
```

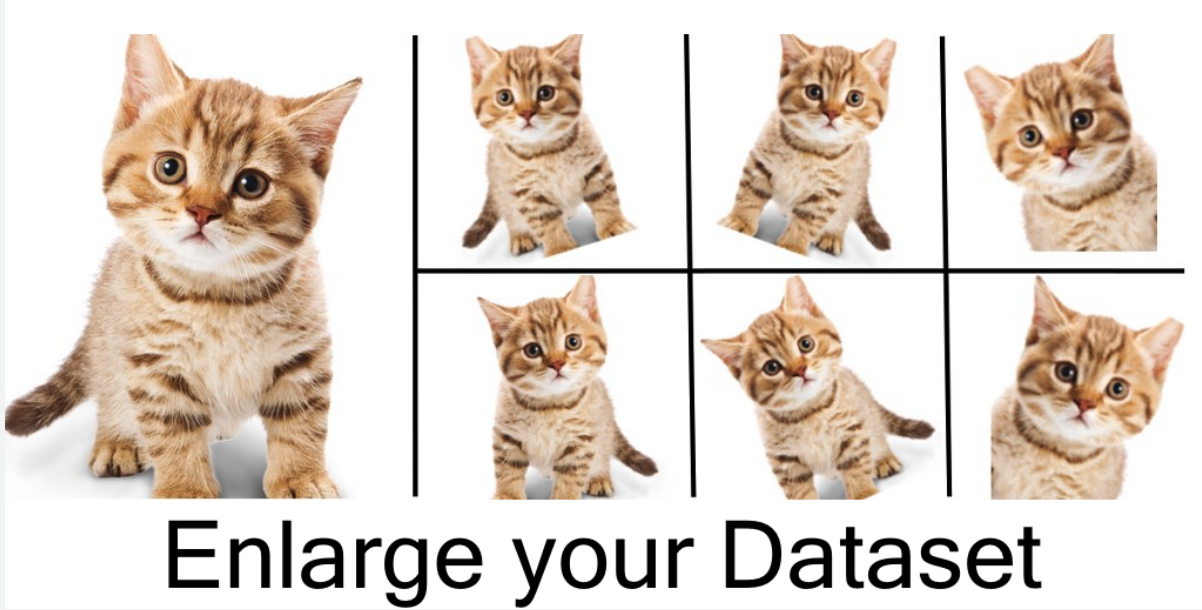
Why are Train Transform and Test Transform different?

```
transform_test = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Data Augmentation for training on small set of data

https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Data Augmentation for training on small set of data

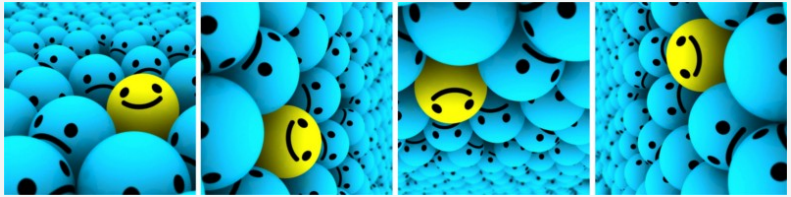
https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

Flip

Crop

Scale

Rotation



Gaussian Noise

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Data Augmentation API in Gluon

https://mxnet.incubator.apache.org/api/python/gluon/data.html
https://mxnet.incubator.apache.org/tutorials/gluon/data_augmentation.html

RandomResizedCrop	Crop the input image with random scale and aspect ratio.
CenterCrop	Crops the image src to the given size by trimming on all four sides and preserving the center of the image.
RandomFlipLeftRight	Randomly flip the input image left to right with a probability of 0.5.
RandomFlipTopBottom	Randomly flip the input image top to bottom with a probability of 0.5.
RandomBrightness	Randomly jitters image brightness with a factor chosen from [max(0, 1 - brightness), 1 + brightness].
RandomContrast	Randomly jitters image contrast with a factor chosen from [max(0, 1 - contrast), 1 + contrast].
RandomSaturation	Randomly jitters image saturation with a factor chosen from [max(0, 1 - saturation), 1 + saturation].
RandomHue	Randomly jitters image hue with a factor chosen from [max(0, 1 - hue), 1 + hue].
RandomColorJitter	Randomly jitters the brightness, contrast, saturation, and hue of an image.
RandomLighting	Add AlexNet-style PCA-based noise to an image.

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

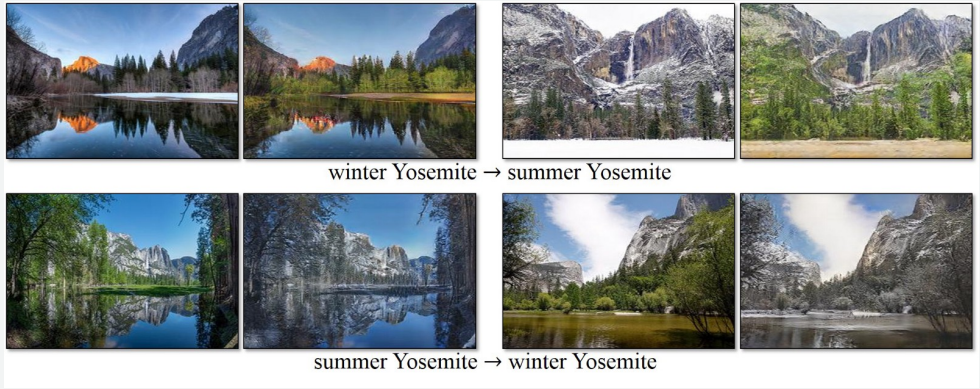
How to apply data augmentation in Gluon

```
from mxnet.gluon.data.vision import MNIST, transform
from mxnet import gluon
transform = transforms.Compose([
    transforms.Resize(300),
    transforms.RandomResizedCrop(224),
    transforms.RandomBrightness(0.1),
    transforms.ToTensor(),
    transforms.Normalize(0, 1)])
data = MNIST(train=True).transform_first(transform)
data_loader = gluon.data.DataLoader(data, batch_size=32, num_workers=1)
for data, label in data_loader:
    # do something with data and label
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Advanced Data Augmentation Method

CycleGAN to change seasons of scene



Neural Style Transfer

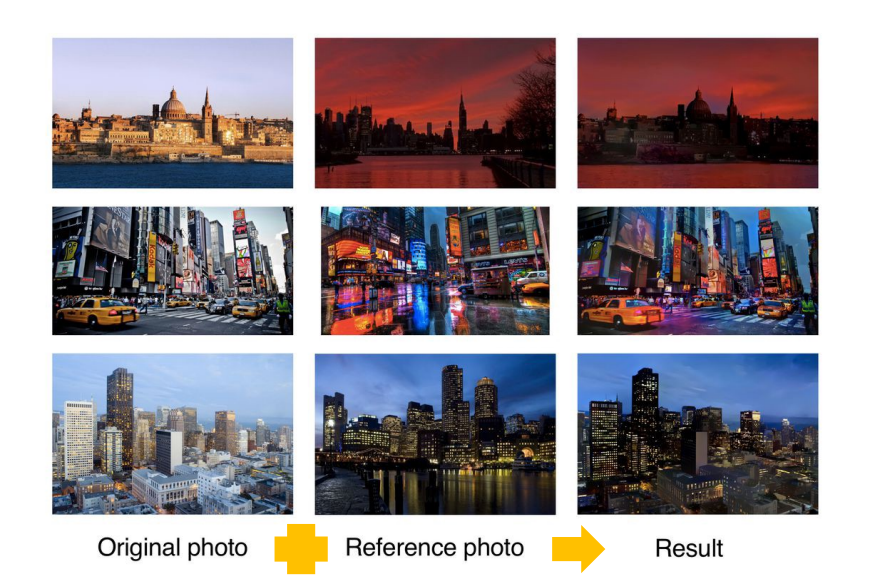


Image Credit: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Quiz: Which augmented images are meaningful?

Original Image

1

2

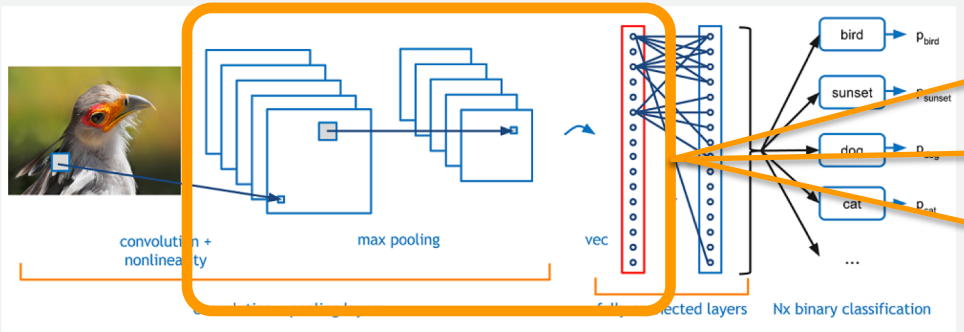
3

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Full Training vs. Transfer Learning

http://cs231n.github.io/transfer-learning/

- **Full Training**: Learning only from your training data
- **Transfer Learning**: Learning from your training data up on the pre-trained model (weights)
    - ConvNet as *fixed* feature extractor
    - Fine-tuning the ConvNet



SVM

Logistic Regression

DNN

Feature Extractor

Classifier

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Transfer Learning – Fine Tuning

1. Load the pre-trained model

```
model_name = 'ResNet50_v2'
finetune_net = gluon.model_zoo.vision.get_model(model_name, pretrained=True)
```

2. Re-define the output layer for the new task

```
with finetune_net.name_scope():
    finetune_net.output = nn.Dense(classes)

finetune_net.output.initialize(init.Xavier(), ctx=ctx)
finetune_net.collect_params().reset_ctx(ctx)
```

3. Train the network

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Lab 3: Transfer Learning

```
!wget https://gluon-cv.mxnet.io/_downloads/transfer_learning_minc.ipynb
```

Resume where you are done in the previous lab

If your notebook instance is not of P2 or P3,
set '*batch_size*' to 32 explicitly.

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

File IO can be a performance blocker if

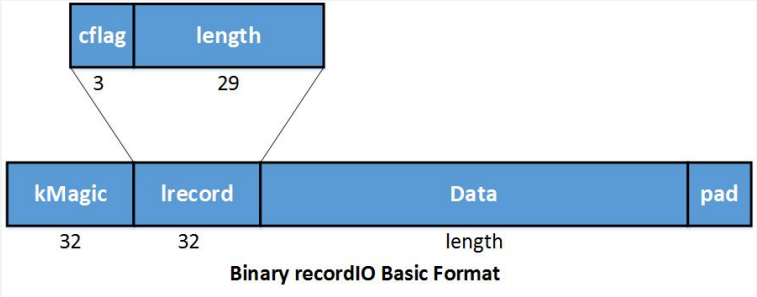
Training data is composed of tons of raw image files

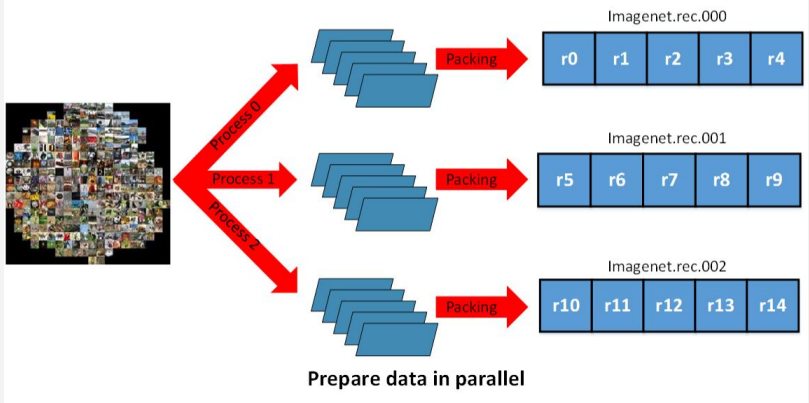
© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

ImageRecord format for your dataset

http://mxnet.incubator.apache.org/architecture/note_data_loading.html

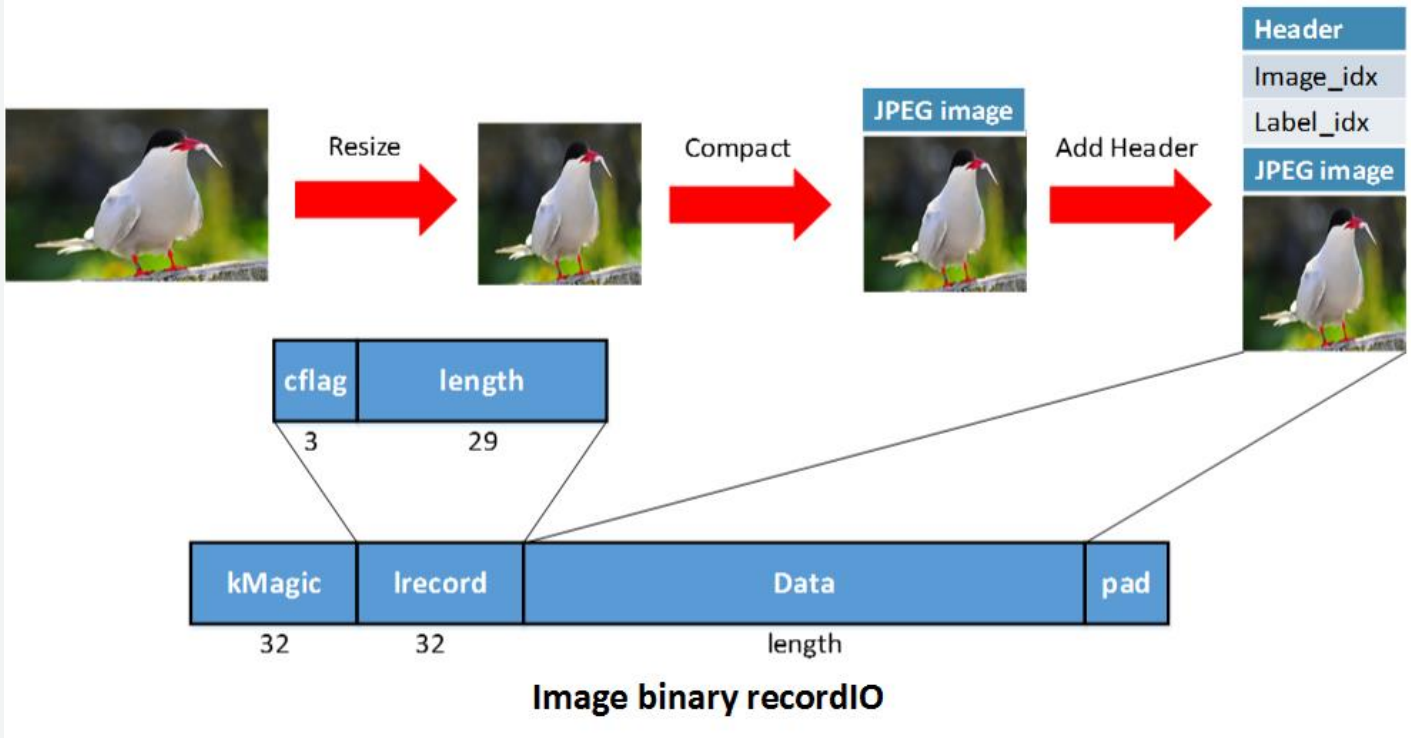
- Small file size.
- Parallel (distributed) packing of data.
- Fast data loading and online augmentation.
- Quick reads from arbitrary parts of the dataset in the distributed setting





© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Image RecordIO Format



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Lab 4 : Creating RecordIO Files

https://gluon-cv.mxnet.io/build/examples_datasets/recordio.html

- Use image files created in Lab 2 (DO NOT TRY TO DOWNLOAD ImageNet files)
- Download *im2rec.py* from http://bit.ly/2vHlHLh, and install opencv-python

```
pip install opencv-python
```

```
./train
├── brick
├── carpet
├── ceramic
├── fabric
├── foliage
├── food
├── glass
├── hair
├── leather
├── metal
├── mirror
├── other
├── painted
├── paper
├── plastic
├── polishedstone
├── skin
├── sky
├── stone
├── tile
├── wallpaper
├── water
└── wood
```

Step 1. Generate a *.lst* file

```
python im2rec.py ./train_rec ./train/ --recursive --list --
num-thread 8
```

Step 2. Generate a *.idx and .rec* files

```
python im2rec.py ./train_rec ./train/ --recursive
--pass-through --pack-label --num-thread 8
```

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Lab 4 : Creating RecordIO Files

https://gluon-cv.mxnet.io/build/examples_datasets/recordio.html

```
!wget https://gluon-cv.mxnet.io/_downloads/recordio.ipynb
```

In Jupyter notebook, read .rec and .idx file and visualize samples by following cells of "**Read with GluonCV**" section

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

END OF GluonCV

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.