AI BUILDERS

Deep learning for image analysis

Titipat Achakulvisut

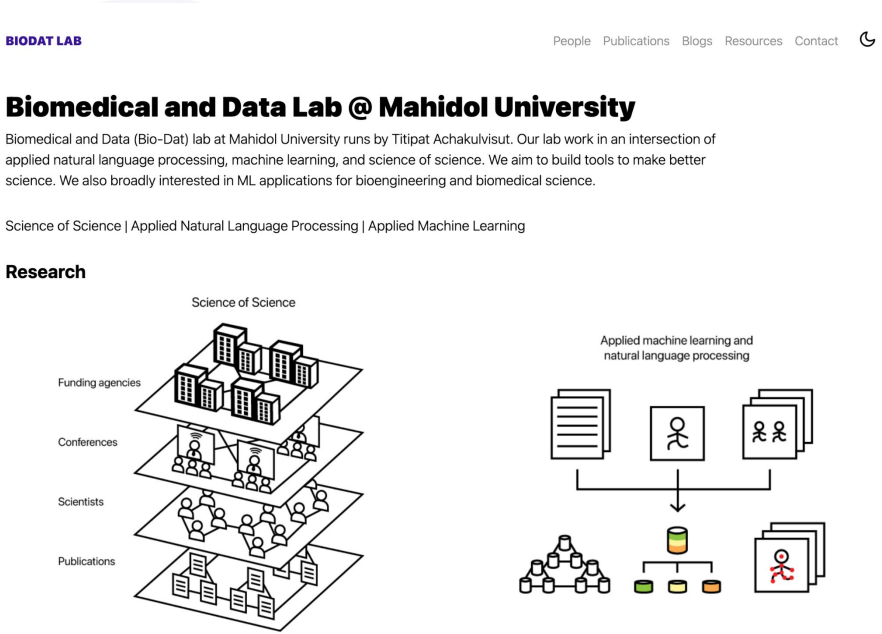
Department of Biomedical Engineering,Mahidol University

Kukkik Oparad

425Degree

AI BUILDERS

Who we are?



Titipat Achakulvisut

Department of Biomedical Engineering, Mahidol University

Kukkik Oparad

425Degree

AGENDA

What we'll Learn in these 3 weeks

- Application of deep learning for image analysis
- Transfer learning for image classification
- Transfer learning for object detection
- Transfer learning for semantic segmentation
- How Convolutional Neural Network (CNN) works?
- Tips and tricks: Augmentation, ...

AI BUILDERS

Image

Image

ภาพ (Image)

- คือ Array ของตัวเลขในขนาด 2 มิติหรือ 3 มิติ ประกอบด้วย ความกว้าง (width), ความสูง (height), ความลึก (depth)
- ปกติในหลายๆ libraries จะจัดวางแบบ (depth, height, width)
- และเวลาเทรนโมเดลด้วย Pytorch เราจะกำหนดให้ batch size อยู่ด้านหน้าสุดตามด้วยภาพ ดังนี้ (batch size, depth, height, width) เช่น (32, 3, 224, 224) แปลว่า batch size = 32, depth = 3, height = 224, width = 224

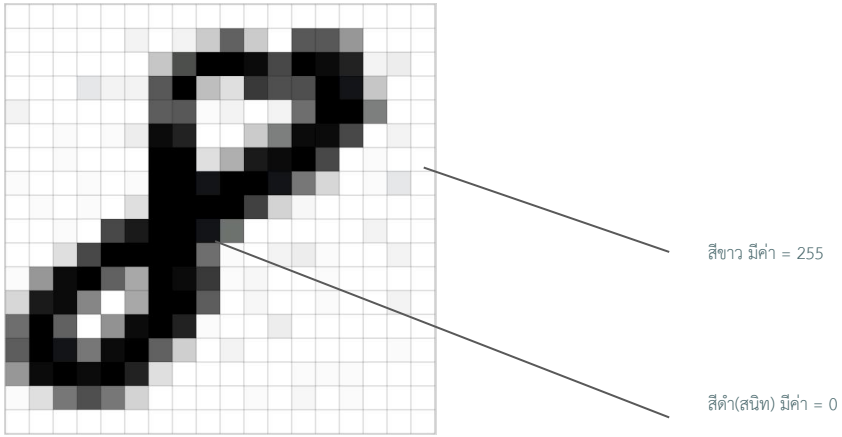
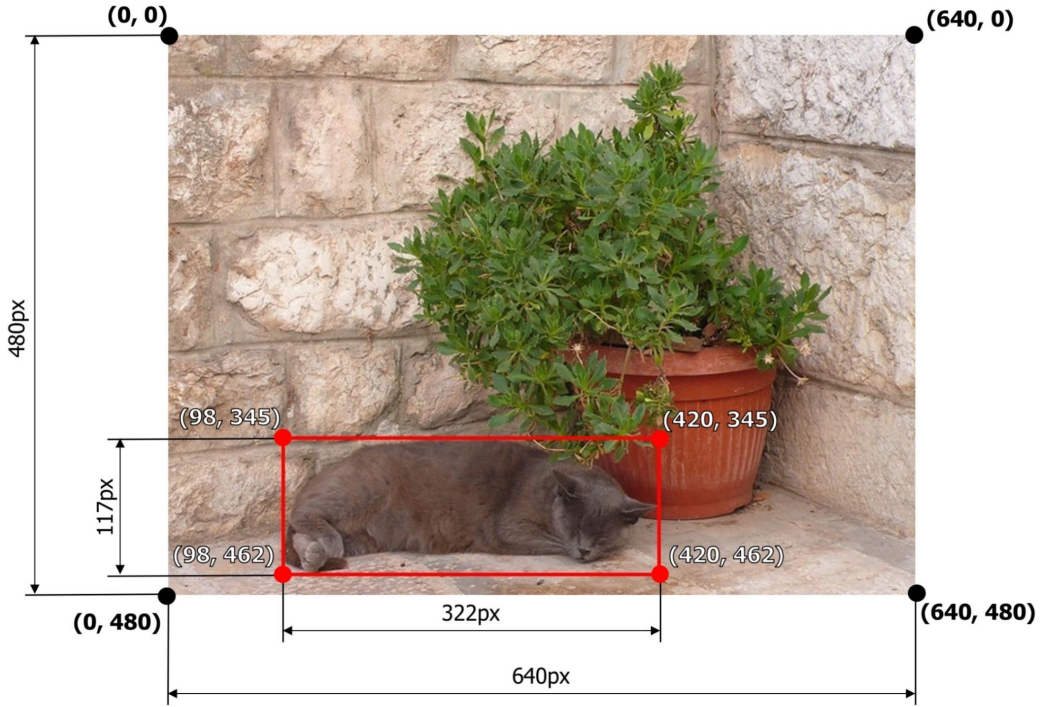


Image positions



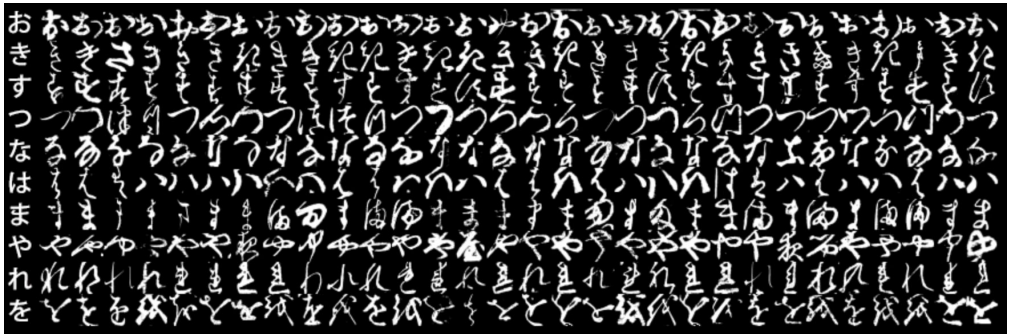
An example image with a bounding box from the COCO dataset

Reference: https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/

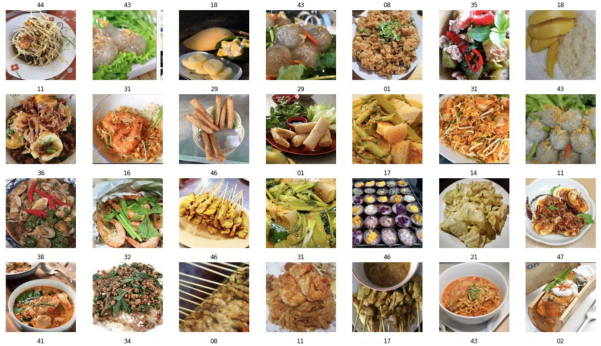
AI BUILDERS

Application of deep learning for image analysis

Image classification

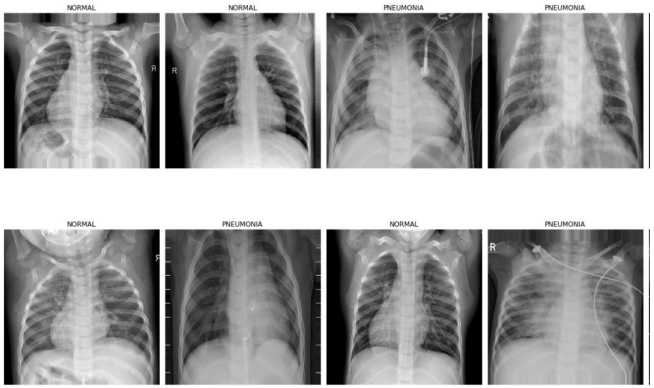


Kuzushiji-MNIST



FoodyDudy (Gemmy, AIB1)

Dog breed classification



Normal / Pneumonia Chest X-ray

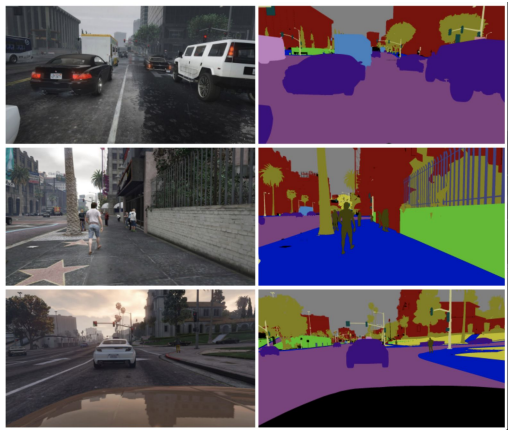
Object detection (ตรวจจับวัตถุ)

Microsoft COCO dataset
เป็นหนึ่งใน dataset ทางด้าน Object detection ที่ใหญ่ที่สุดอันหนึ่งในปัจจุบัน

Object tracking: คล้ายกับ object detection แต่ว่า สร้าง unique ID สำหรับการตรวจจับวัตถุตอนเริ่มต้นด้วย

รถขับเคลื่อนอัจฉริยะ: Tesla Object detection (รถ, ไฟจราจร, เลน, ...) และ Semantic Segmentation (แยกถนนออกจากฉากอื่นๆ)

Semantic Segmentation

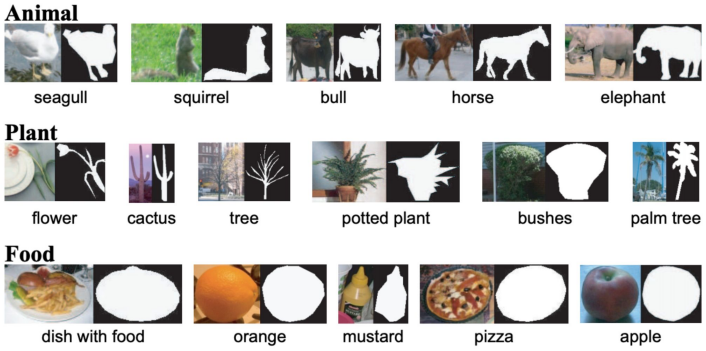


GTA5 dataset

NVIDIA Semantic Segmentation

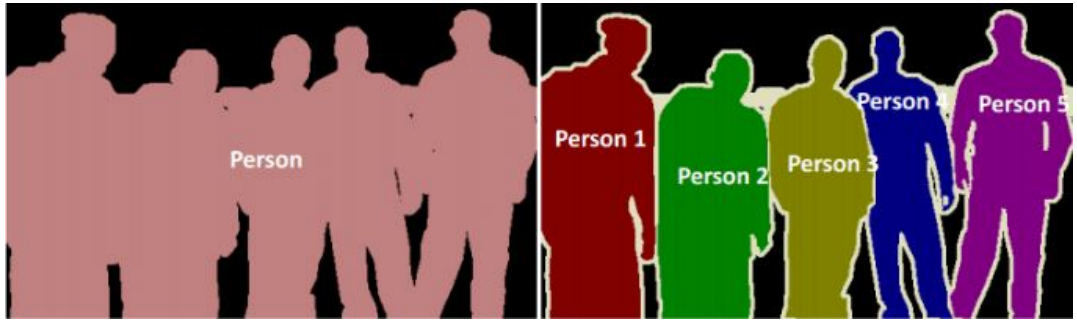
SUN RGB-D dataset (https://rgbd.cs.princeton.edu/)

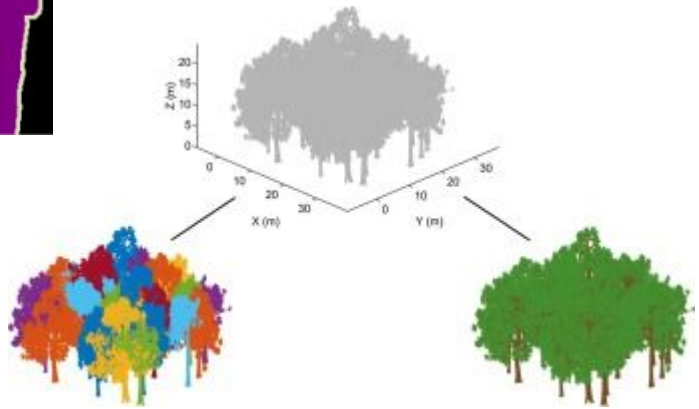
iMat Fashion dataset



LabelMe database (Russell et al.)

Instance Segmentation



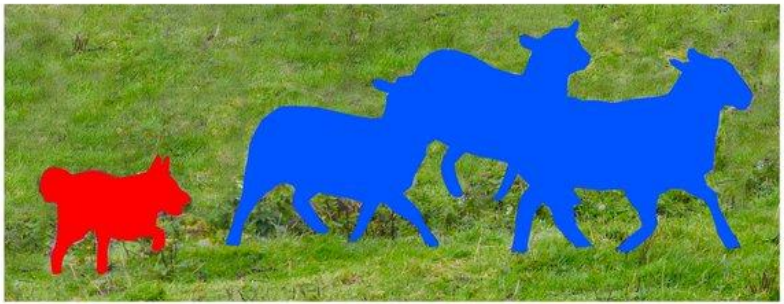


Instance segmentation: มีการทำงานคล้ายกับ Semantic Segmentation

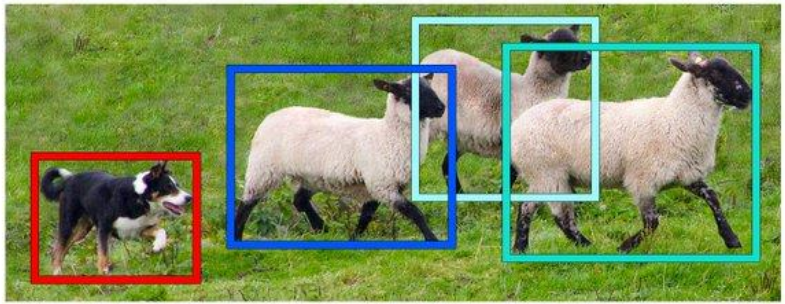
แต่ว่าสามารถระบุ ID ของวัตถุในภาพด้วย

สรุป

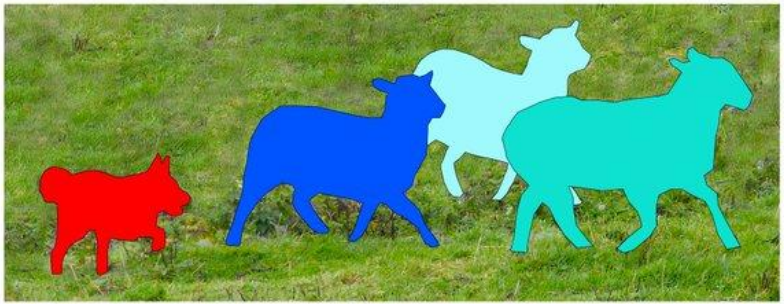
Image Recognition



Semantic Segmentation



Object Detection



Instance Segmentation

Comparing AI Vision Tasks

(https://ai-pool.com/d/could-you-explain-me-how-instance-segmentation-works)

Visual Question Answering (VQA)

Vehicles and Transportation

Q: What sort of vehicle uses this item?
A: firetruck

Brands, Companies and Products

Q: When was the soft drink company shown first created?
A: 1898

Objects, Material and Clothing

Q: What is the material used to make the vessels in this picture?
A: copper

Sports and Recreation

Q: What is the sports position of the man in the orange shirt?
A: goalie

Cooking and Food

Q: What is the name of the object used to eat this food?
A: chopsticks

Geography, History, Language and Culture

Q: What days might I most commonly go to this building?
A: Sunday

People and Everyday Life

Q: Is this photo from the 50's or the 90's?
A: 50's

Plants and Animals

Q: What phylum does this animal belong to?
A: chordate, chordata

Science and Technology

Q: How many chromosomes do these creatures have?
A: 23

Weather and Climate

Q: What is the warmest outdoor temperature at which this kind of weather can happen?
A: 32 degrees

ภาพ (Image) + คำถาม (Natural language) ⟶ ดึงคำตอบออกมา

AI BUILDERS

Deep learning models and tools

การสร้างโมเดล

General Pipeline

กำหนดข้อมูล Input, Output
X : input, features
Y : output

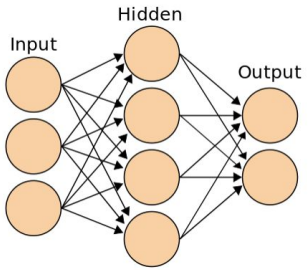
กำหนด **โมเดล**

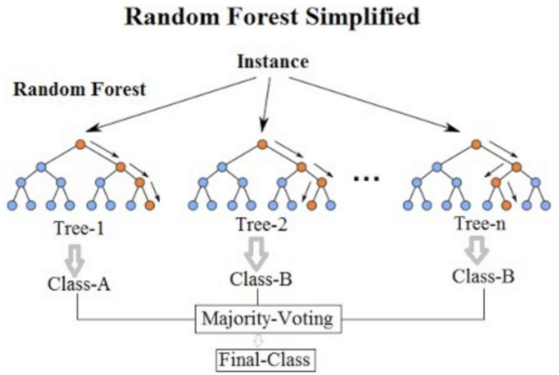
สร้าง **Cost function**

โมเดล

Optimize **Cost Function (Gradient descent)**

เทสโมเดลโมเดลกับ**ชุดข้อมูลที่ไม่เคยเห็น**





โมเดลในที่นี้อาจจะเป็นโมเดลอะไรก็ได้เช่น Neural Network (ที่ใช้สอนใน AI Builders เป็นหลัก) หรือว่าอาจจะเป็นโมเดลอื่นๆ เช่น Random forest, Support Vector Machine (SVM), ...

AI BUILDERS

การเขียนโค้ด

Dataset
กำหนดชุดข้อมูล
และ transformation
แบ่งเป็น train, validation, test set

Dataloader
โหลดข้อมูลมาเป็น batch
สำหรับใส่เข้าไปในโมเดล

สร้างโมเดลและ Loss function

เทรนโมเดล
ใส่ข้อมูล คำนวณ Loss และอัพเดทโมเดล

เทรนโมเดล

การเลือก Tools หรือ Library สำหรับสร้างโมเดล

Low level

- **Pytorch**
- Tensorflow
- Jax

Mid level

- Pytorch Ignite
- Tensorflow **Keras**
- **Pytorch Lightning**

High level

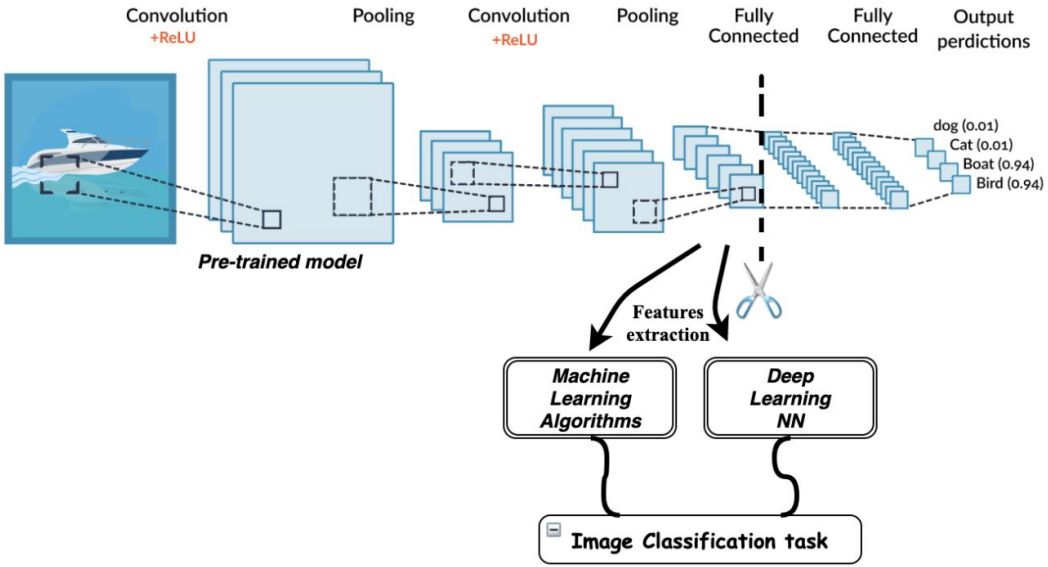
- **FastAI**
- Pytorch Lightning Flash
- Icevision (wrapper for FastAI, Lightning Flash)

AI BUILDERS

Transfer learning for image classification

Transfer learning

- **Transfer learning** เป็นหนึ่งในเทคนิคของ Machine learning ที่นำโมเดลที่สร้างมาสำหรับงานประเภทหนึ่งถูกนำมาใช้กับงานอีกประเภทหนึ่ง
- หลักการโดยคร่าวๆของ Transfer learning คือการนำโมเดลที่ถูกเทรนมาแล้ว (Pre-trained model) มาทำการดัดแปลงพารามิเตอร์บางส่วนจากโมเดลเดิมและเทรนโมเดลสำหรับข้อมูลใหม่ของเรา
- เทคนิคนี้ถูกสร้างมาในการแก้ไขปัญหาทาง Computer vision หรือภาพถ่ายทางการแพทย์ที่บางครั้งข้อมูลตั้งต้นอาจจะมีไม่มาก
- หลังจากนั้นเทคนิคนี้ก็ถูกนำมาประยุกต์ใช้ใน Natural Language Processing ด้วยในการ
  สร้างโมเดลภาษาจากข้อมูลขนาดใหญ่



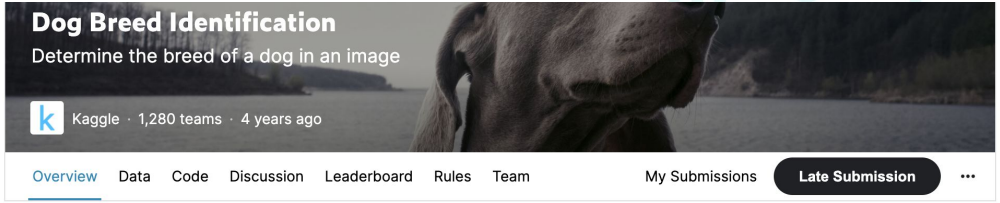
Ref: https://medium.com/@lorenzofamiglini/transfer-learning-with-deep-learning-machine-learning-techniques-b4052befe7e2

Transfer learning: ImageNet Competition

- หนึ่งในโมเดลที่ถูกนำมาใช้สำหรับ Pretrain model ถูกพัฒนามาจากการแข่งขัน ImageNet
- นักวิจัยเขียนโมเดลเพื่อแยกประเภทของภาพจากชุดข้อมูล ImageNet ทุกปี โดยมีโมเดล Deep learning มากมายที่สามารถแบ่งประเภทของภาพด้วยความแม่นยำสูงมากๆ เช่น VGG, MobileNet, DenseNet, ...
- ชุดข้อมูล ImageNet ประกอบด้วย 14,197,122 รูปภาพที่แบ่งประเภทของภาพกว่า 20,000 แบบ (classes)
- จำนวนภาพที่มี bounding box อีกกว่า 1,034,908 ภาพ
- โมเดลที่ถูกเทรนเรียบร้อยถูกเผยแพร่ใน Deep learning library ต่างๆ มากมายเช่น pytorch, tensorflow

Dog breed identification

- **Dog breed identification dataset** ประกอบด้วยภาพหมาใน training set ทั้งหมด 10.2k ภาพ และภาพหมาใน testing set ทั้งหมด 10.4k ภาพ
- ชุดข้อมูลประกอบด้วยพันธุ์หมา (breed) ทั้งหมด 120 พันธุ์ ยกตัวอย่าง เช่น borzoi, basenji, maltese_dog, bluetick, golden_retriever, irish_water_spaniel, ...



Overview

Description

Evaluation

Who's a good dog? Who likes ear scratches? Well, it seems those fancy deep neural networks don't have *all* the answers. However, maybe they can answer that ubiquitous question we all ask when meeting a four-legged stranger: what kind of good pup is that?

In this playground competition, you are provided a strictly canine subset of ImageNet in order to practice fine-grained image categorization. How well you can tell your Norfolk Terriers from your Norwich Terriers? With 120 breeds of dogs and a limited number training images per class, you might find the problem more, err, ruff than you anticipated.

labels.csv (482.06 kB)

Detail Compact Column

id	breed
10222 unique values	scottish_deerhound 1%, maltese_dog 1%, Other (9979) 98%
007b5a16db9d9ff9d7ad39982703e429	wire-haired_fox_terrier
007b8a07882822475a4ce6581e70b1f8	redbone
007ff9a78eba2aebb558afea3a51c469	lakeland_terrier
008887054b18ba3c7601792b6a453cc3	boxer

- train
  - 000bec180eb18c76...
  - 001513dfcb2ffafc82...
  - 001cdf01b096e06d7...
  - 00214f311d5d2247d...
  - 0021f9ceb3235effd...
  - 002211c81b498ef88...
  - 00290d3e1fdd2722...
  - 002a283a315af96ea...

Ref: https://www.kaggle.com/c/dog-breed-identification

ตัวอย่าง

- นำ Transfer learning มาเพื่อแยกพันธุ์น้องหมา (Dog breed identification)
- เขียนแอพพลิเคชั่นเพื่อแยกพันธุ์น้องหมา

Transfer learning using FastAI

```
dblock = DataBlock(
    blocks=(ImageBlock, CategoryBlock), #x - image; y - single class
    get_items=get_image_files, #get image
    splitter=GrandparentSplitter(valid_name='valid_mini'), #use parent folder as train-valid split
    get_y=parent_label, #use parent folder as label
    batch_tfms=aug_transforms(size=224)
    )
dls = dblock.dataloaders('FoodyDudy/images/', bs=64)
```

```
/usr/local/lib/python3.7/dist-packages/torch/_tensor.py:1051: UserWarning: torch.solve is deprecated in favor
be removed in a future PyTorch release.
torch.linalg.solve has its arguments reversed and does not return the LU factorization.
To get the LU factorization see torch.lu, which can be used with torch.lu_solve or torch.lu_unpack.
X = torch.solve(B, A).solution
should be replaced with
X = torch.linalg.solve(A, B) (Triggered internally at  ../aten/src/ATen/native/BatchLinearAlgebra.cpp:766.)
  ret = func(*args, **kwargs)
```

```
dls.train.show_batch(max_n=9,nrows=3)
```

กำหนดชุดข้อมูล

```
learn = cnn_learner(dls, resnet34, metrics=accuracy)
learn.fine_tune(epochs=0, freeze_epochs=1, base_lr=2e-3)
```

```
Downloading: "https://download.pytorch.org/models/resnet34-b627a593.pth
  0%|          | 0.00/83.3M [00:00<?, ?B/s]
```

epoch	train_loss	valid_loss	accuracy	time
0	2.153407	0.816388	0.758333	03:00

ดูผลการทำนายคร่าวๆ ใน validation set ว่าทำนายผิดจากอะไรเป็นอะไรบ้าง

```
learn.show_results() #true label - บน; prediction - ล่าง
```

ใส่ชุดข้อมูล, สถาปัตยกรรมของโมเดล, วิธีการวัดผล

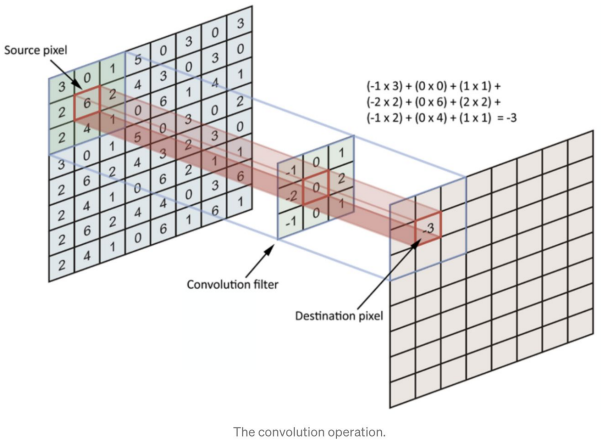
AI BUILDERS

How convolutional neural network (CNN) works?

CNN ทำงานอย่างไร?

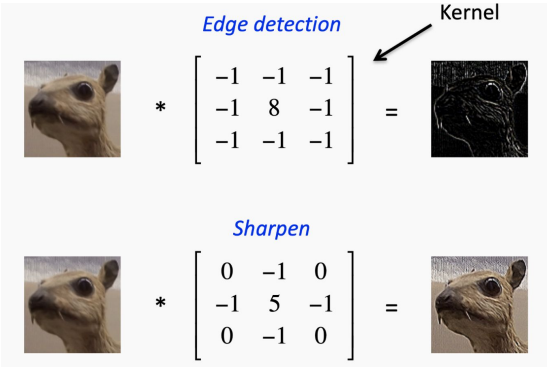
อธิบายการทำงานของฟิลเตอร์ (Filters)

- ปกติภาพสีมีทั้งหมด 3 dimensions: ความลึก/depth (เช่น RGB), ความกว้าง/width, ความสูง/height
- ปกติค่าของแต่ละ pixel มีค่าระหว่าง 0 ถึง 255 (0 = ดำ, 255 = ขาว)
- Filter หรือ Kernel ขนาด n x n มีหน้าที่วิ่งไปตามภาพ (ทั้งแกน x, y) เพื่อเปลี่ยนหน้าตาของภาพผ่านกระบวนการ convolution (คูณและบวกเลข)



The convolution operation.

Convolution filter

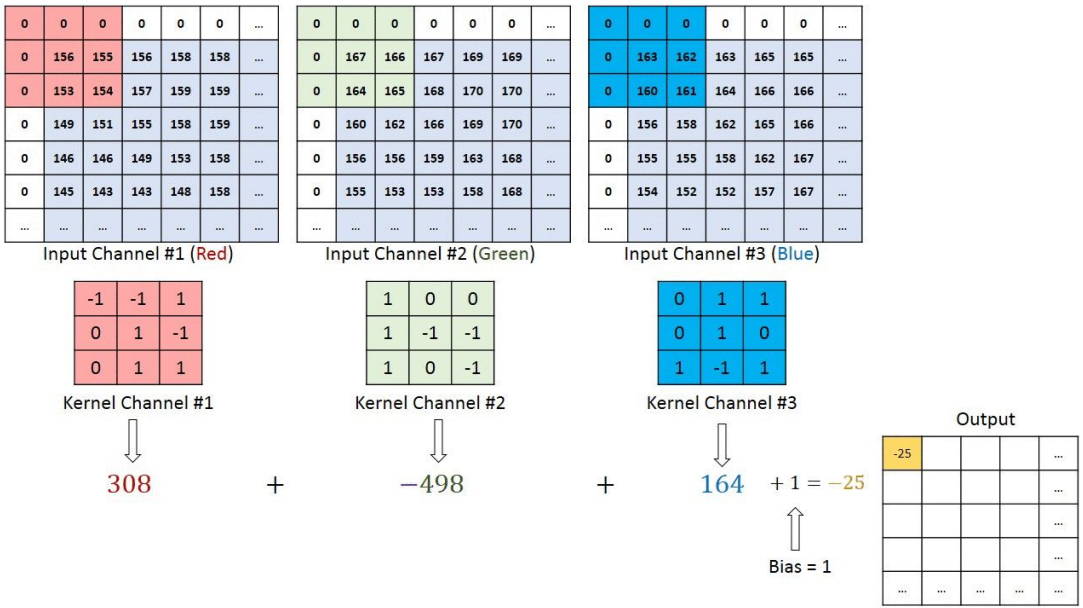


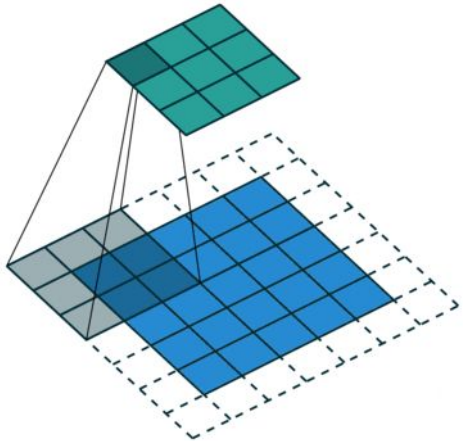
Kernel size = 3

Example of filters

Reference: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

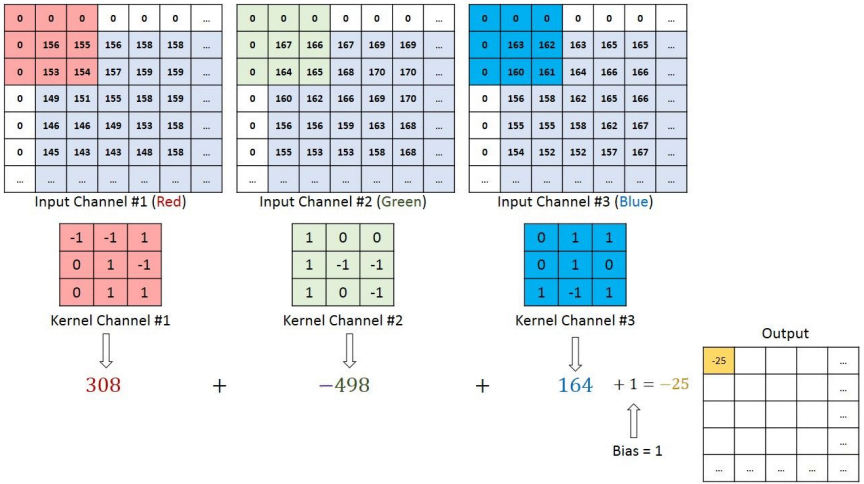
อธิบายการทำงานของฟิลเตอร์ (Filters)

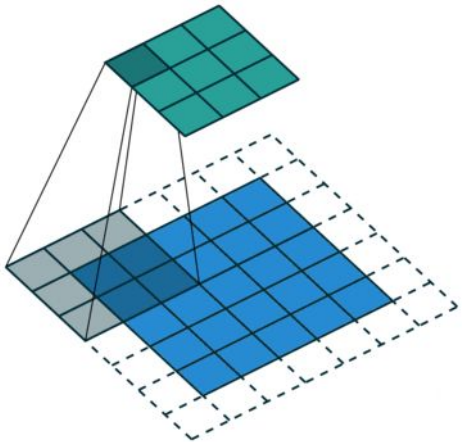




Reference: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

อธิบายการทำงานของฟิลเตอร์ (Filters)



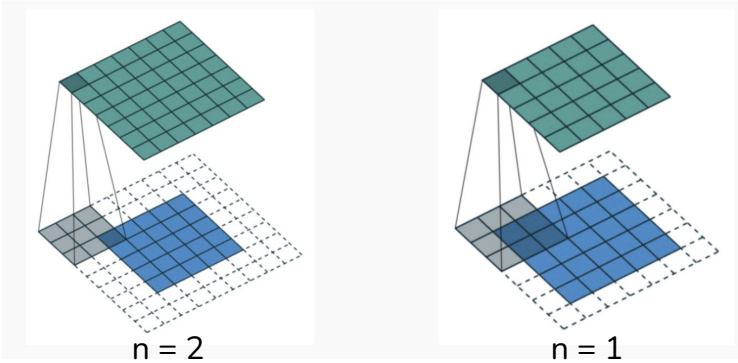


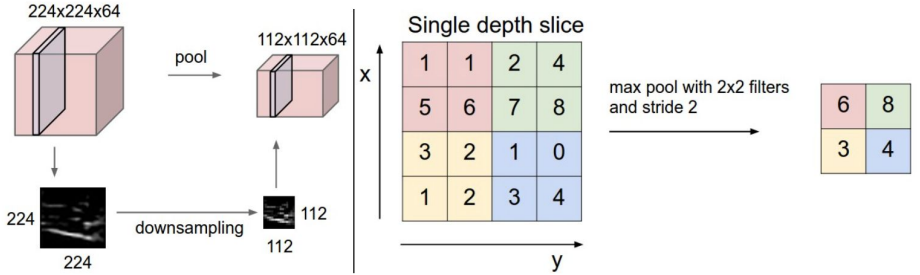
Reference: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

Convolutional layer & Max Pooling layer

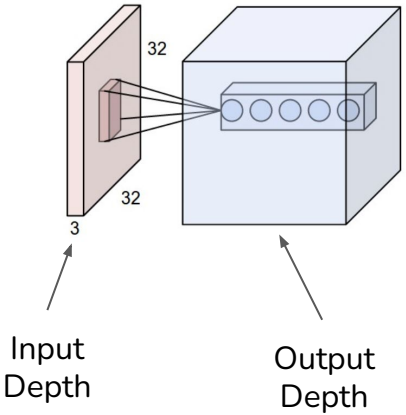
Kernel size = 3

Padding





Max pooling layer



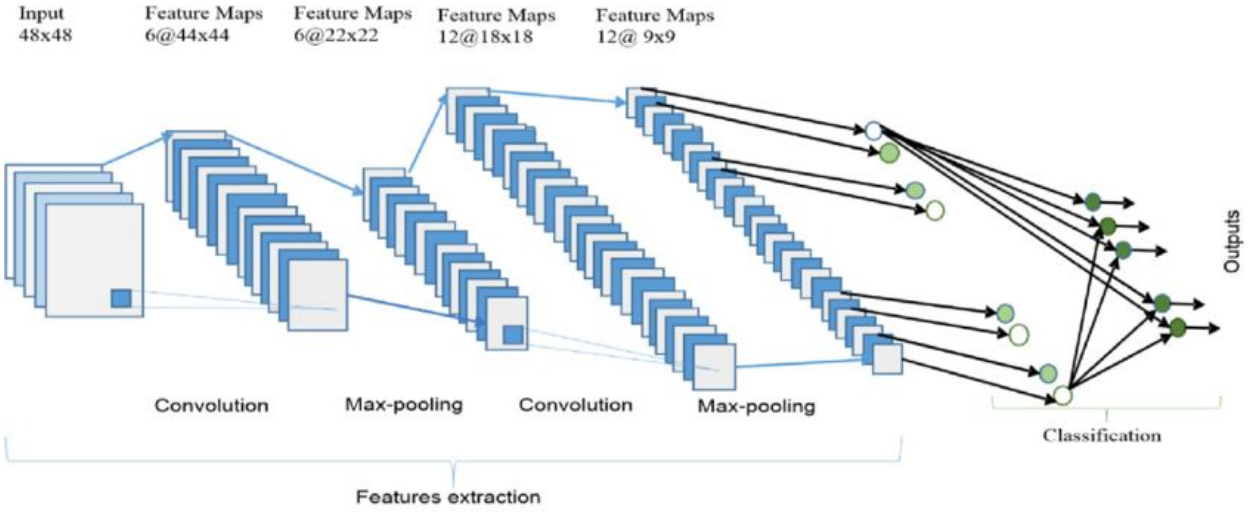
Convolutional layer

- CNN ประกอบด้วยส่วนสำคัญนอกจาก Dense layer สองอย่างคือ Convolutional layer และ Max pooling layer
- นอกจากนั้นอาจจะต้องคำนึงถึง padding ด้วยเนื่องจากภาพที่ได้จาก Convolutional layer อาจจะมีขนาดไม่เท่าเดิม
- วิธีคำนวณ image size: (((W - K + 2P)/S) + 1)

  W = Input size, K = Filter size, S = Stride, P = Padding vv

Reference: https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac

Convolutional Neural Network (CNN)



- Convolutional neural network (CNN) ประกอบด้วย Convolutional layer, Max pooling layer, และ fully connected layer (dense layer)
- โมเดลอื่นๆใช้องค์ประกอบของ Convolutional layer, Max pooling layer เข้ามาสร้างโมเดลที่มีขนาดใหญ่มากยิ่งขึ้นสำหรับข้อมูลขนาดใหญ่ขึ้น

Ref: A State-of-the-Art Survey on Deep Learning Theory and Architectures

AI BUILDERS

Tips and tricks

Image augmentation

Transforms on PIL Image and torch.*Tensor

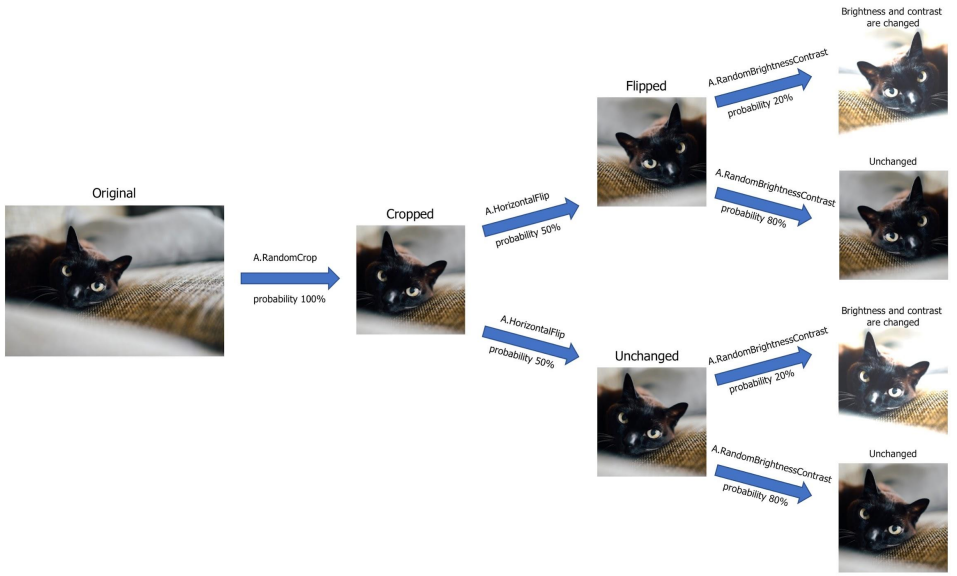
`CenterCrop` (size)	Crops the given image at the center.
`ColorJitter` ([brightness, contrast, ...])	Randomly change the brightness, contrast, saturation and hue of an image.
`FiveCrop` (size)	Crop the given image into four corners and the central crop.
`Grayscale` ([num_output_channels])	Convert image to grayscale.
`Pad` (padding[, fill, padding_mode])	Pad the given image on all sides with the given "pad" value.
`RandomAffine` (degrees[, translate, scale, ...])	Random affine transformation of the image keeping center invariant.
`RandomApply` (transforms[, p])	Apply randomly a list of transformations with a given probability.
`RandomCrop` (size[, padding, pad_if_needed, ...])	Crop the given image at a random location.

ตัวอย่างของ Image augmentation

- Image augmentation เป็นวิธีหนึ่งในการขยายรูปแบบของข้อมูลเพื่อ โมเดลเห็นในระหว่างการเทรนโมเดล
- ยกตัวอย่างเช่น
  - RandomHorizontalFlip จะทำการกลับภาพจากซ้ายไปขวา
  - RandomVerticalFlip ทำการกลับภาพจากบนลงล่าง
  - RandomRotation ทำการหมุนภาพไปในองศาต่างๆแบบสุ่ม
- เป็นการเพิ่ม distribution ของภาพที่ต่างไปจากเดิมเล็กน้อย โดยโมเดลยังสามารถเรียนรู้จากข้อมูลเหล่านี้ได้

https://pytorch.org/vision/master/transforms.html

Image augmentation (cont.)



References: https://albumentations.ai/docs/getting_started/image_augmentation/

Synthetic data augmentation

NVIDIA DRIVE Sim
https://www.youtube.com/watch?v=RVFIDEuNtt0

- นำภาพจากอินเตอร์เน็ตมาช่วยในการ generate ภาพเพิ่มเติม
- เช่นกรณีของ OCR สามารถนำตัวอักษรในการเขียนแบบต่างๆเข้ามาสร้างชุดข้อมูลใหม่
- กรณีของ object detection, semantic segmentation อาจจะนำเทคโนโลยี VR เข้ามาช่วยในการสร้างข้อมูลเพิ่มเติม (เช่น สร้างโลกเสมือนเพื่อเทรนข้อมูลเพิ่มเติม ก่อนนำไปใช้ในชีวิตจริง)

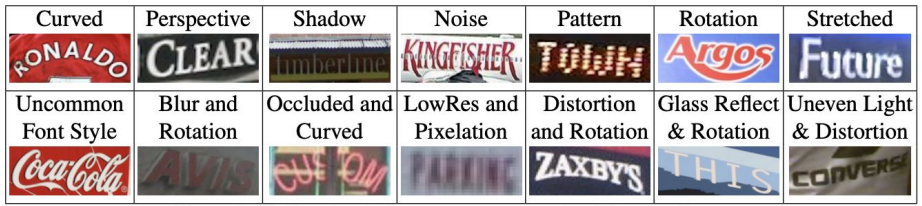


Figure 2. Challenging text appearances encountered in natural scenes

Data Augmentation for Scene Text Recognition
https://arxiv.org/pdf/2108.06949.pdf

AI BUILDERS

Deep learning for object detection and semantic segmentation

Titipat Achakulvisut

Department of Biomedical Engineering,Mahidol University

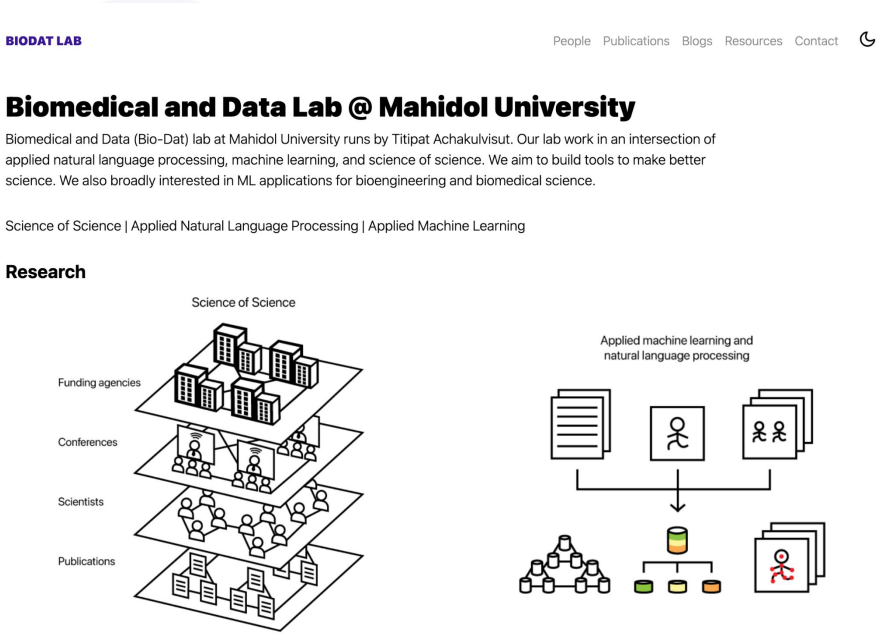
Kukkik Oparad

425Degree

NLP ไม่ได้มีแค่ตัดคำ
Object detection ไม่ได้มีแค่ YOLO
ความผิดคือเราแค่เคยทำทั้ง 2 อย่าง

AI BUILDERS

Who we are?



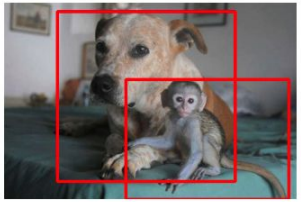
Titipat Achakulvisut

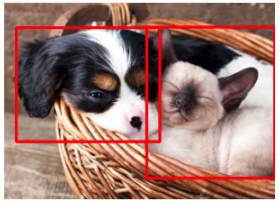
Department of Biomedical Engineering, Mahidol University

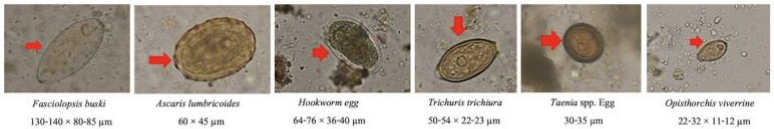
Kukkik Oparad

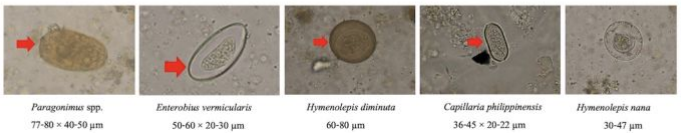
425Degree

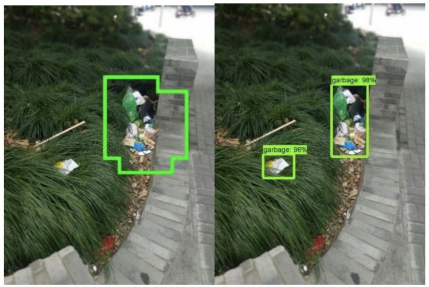
Object detection: Motivation







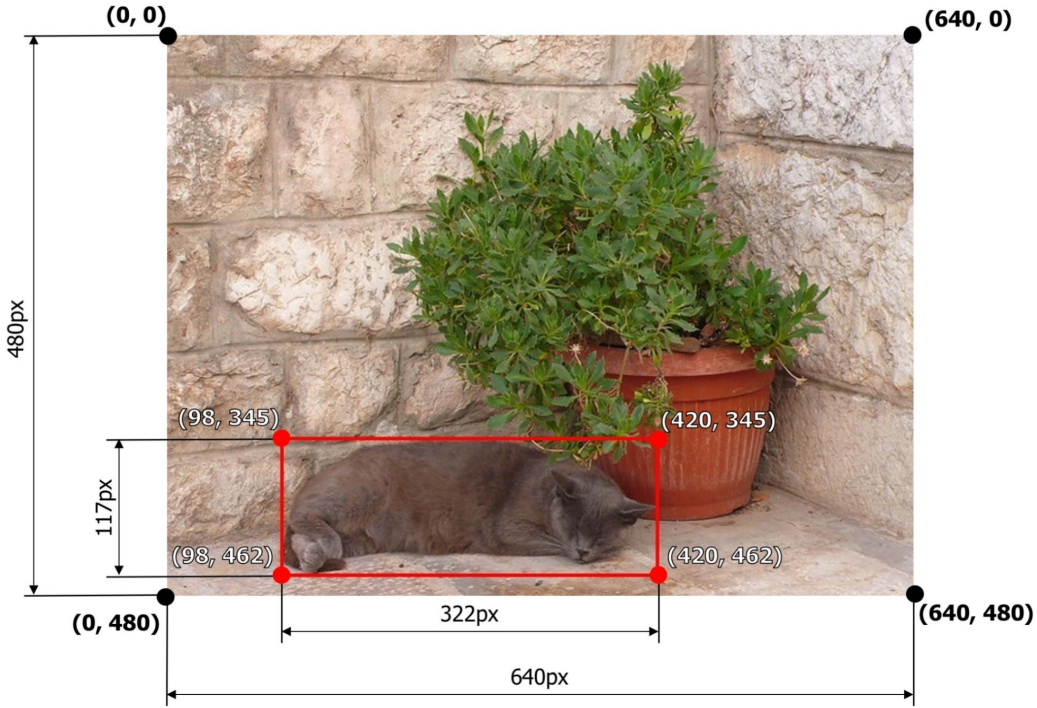




https://icip2022challenge.piclab.ai/dataset/

Wang et al. Autonomous garbage detection for intelligent urban management

Object detection

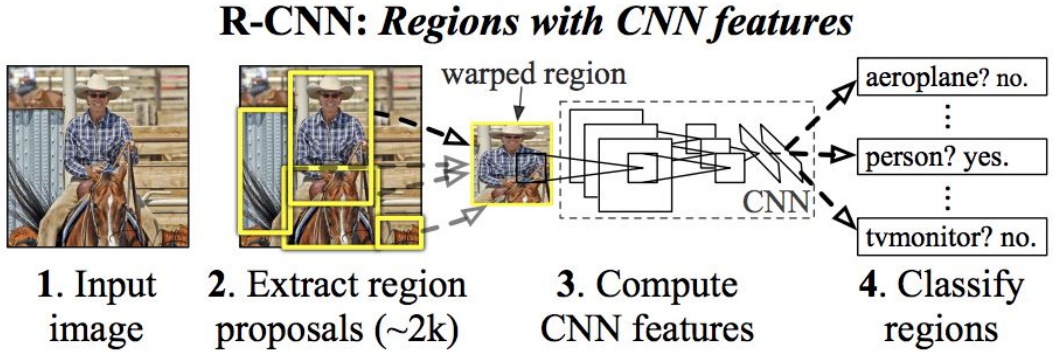


An example image with a bounding box from the COCO dataset

- Object detection มีความหลากหลายมากๆในเชิงเทคนิคและแนวคิดในการสร้างโมเดล แต่อยู่ภายใต้หลักการคล้ายๆกัน
- เราสามารถใช้จุดอย่างต่ำ 4 จุดในการทำนาย bounding box (xmin, ymin, xmax, ymax)
- จากนั้นสามารถใช้ probability ในการทำนายว่าจะมีวัตถุอะไรใน bounding box นั้นๆคล้ายกับ classification [p_cat, p_dog, p_monkey, ...]
- ส่วนมากเราจะทำนายด้วยว่าความน่าจะเป็นที่จะมีวัตถุใน bounding box นั้นมั้ย

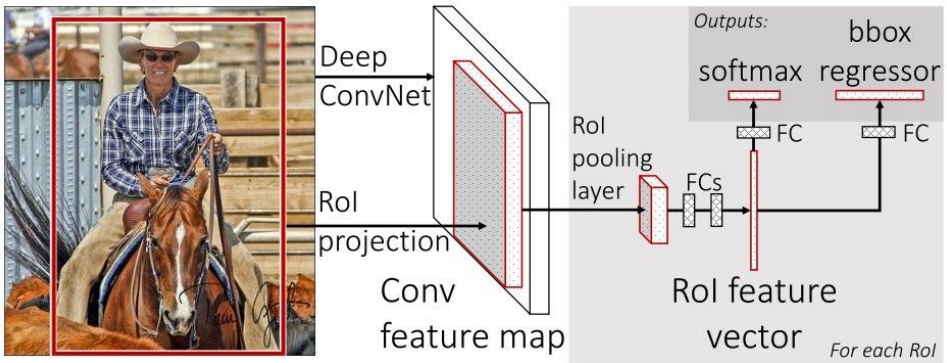
Reference: https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/

Object detection: RCNN



เป็น Rule-based ก็ได้หรือใช้

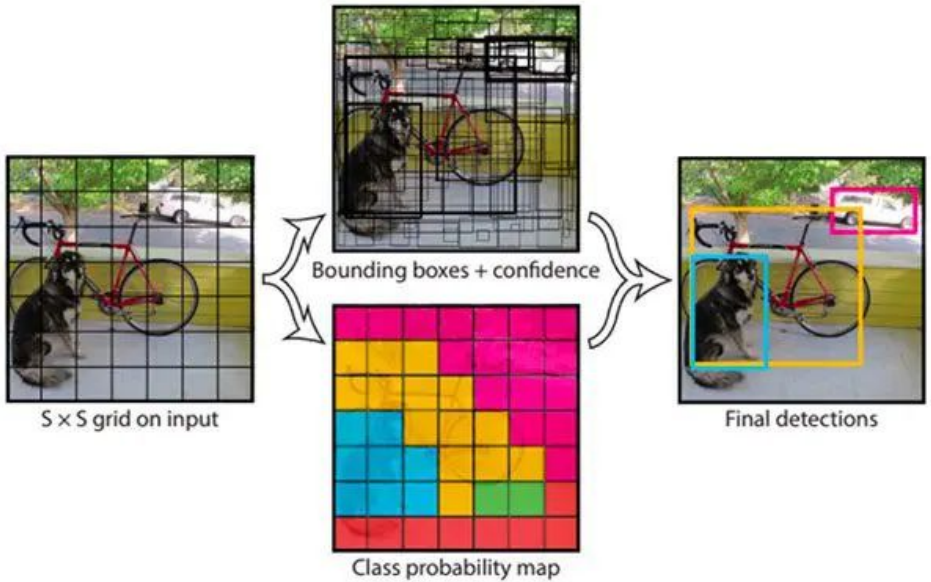
Neural network ในการทำนายก็ได้



https://www.researchgate.net/figure/The-framework-of-R-FCN-RPN-region-proposal-network-RoI-region-of-interest-ConvNets_fig3_333048961

https://medium.com/ai%C2%B3-theory-practice-business/object-detection-in-deep-learning-part2-855b78689f13

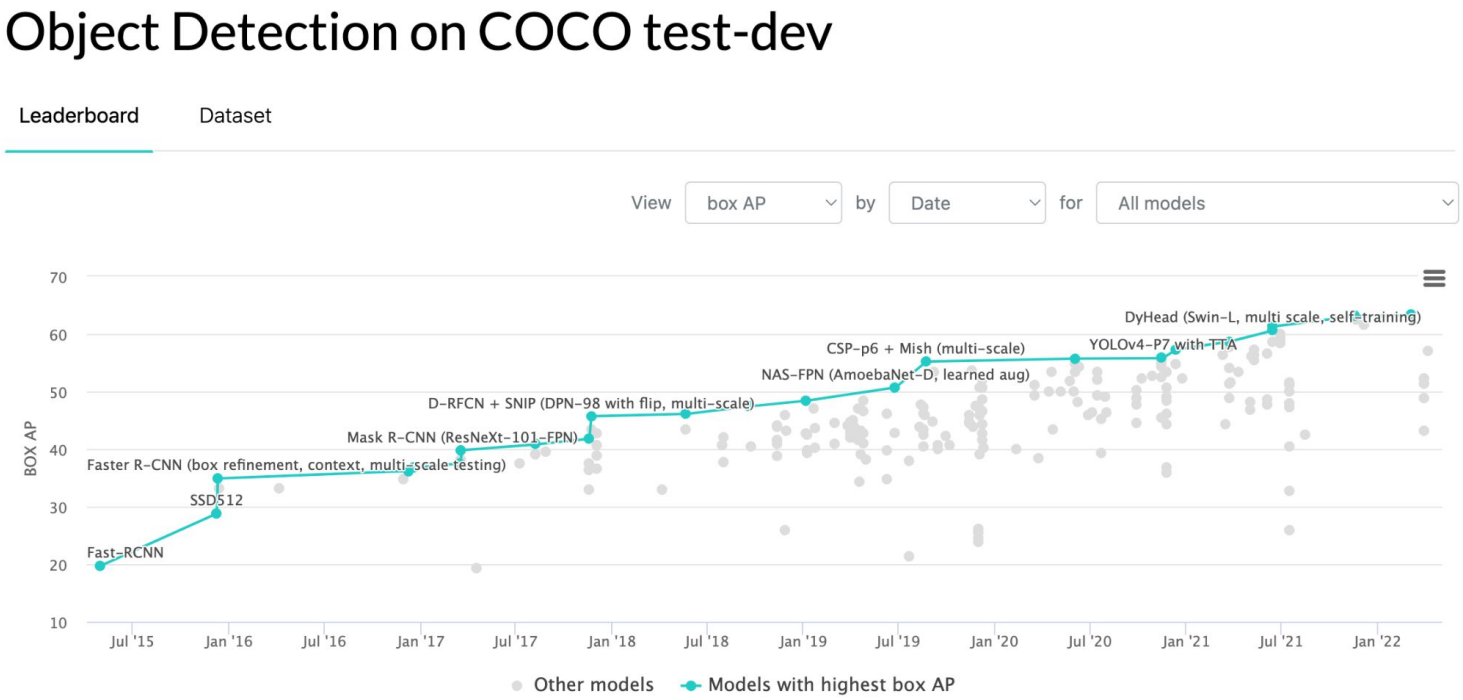
Object detection: YOLO



YOLO แบ่งการทำนายวัตถุในภาพเป็น Grid (เช่น 7x7) แล้วใช้การทำนายวัตถุร่วมกับ bounding box ในแต่ละ grid ในการหา bounding box และชนิดของวัตถุ

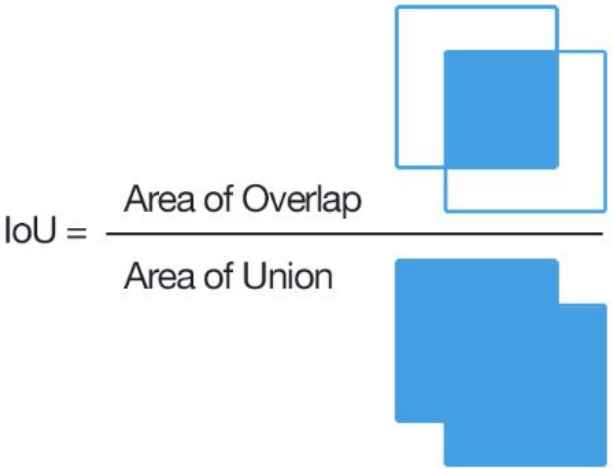
https://pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/

Object detection: benchmark



https://paperswithcode.com/sota/object-detection-on-coco-minival

Object detection: การวัดผล

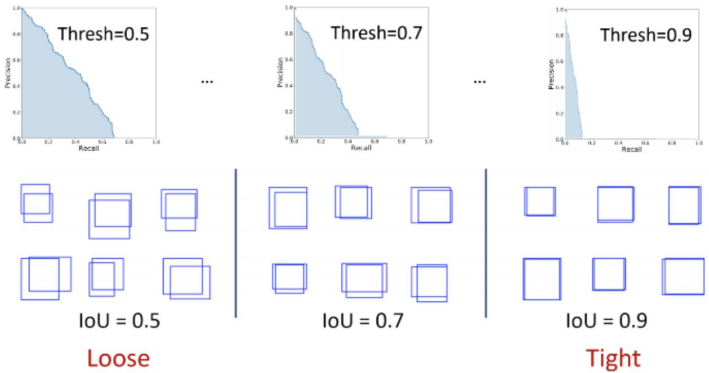


Intersection over union (IoU)

Two minute additions: Usually, the averages are taken in a different order (the final result is same), and in COCO, mAP is also referred to as AP i.e.

- *Step 1:* For each class, calculate AP at different IoU thresholds and take their average to get the AP of that class.

$$\text{AP}[\text{class}] = \frac{1}{\#\text{thresolds}} \sum_{\text{iou} \in \text{thresholds}} AP[class, iou]$$



- *Step 2:* Calculate the final AP by averaging the AP over different classes.

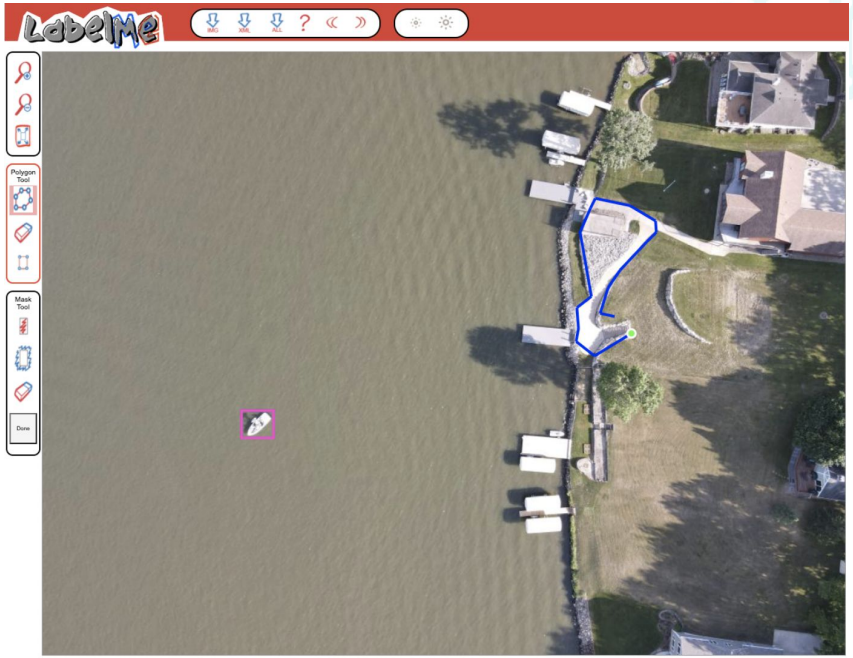
$$\text{AP} = \frac{1}{\#\text{classes}} \sum_{\text{class} \in \text{classes}} AP[class]$$

AP is in fact an average, average, average precision.

Reference: https://kharshit.github.io/blog/2019/09/20/evaluation-metrics-for-object-detection-and-segmentation

Object detection dataset

- การเก็บ data ของ Object detection มีหลากหลายวิธีมาก เช่น COCO JSON, Tensorflow TFRecord, Pascal VOC XML, Amazon Sagemaker GroundTruth Manifest, ...
- COCO format เป็นฟอร์แมตแบบ JSON ที่ใช้สำหรับเก็บชุดข้อมูลแบบหนึ่ง
- ไลบรารี่บางอย่างสร้างให้สามารถรับไฟล์แบบ CSV ได้เช่นกัน
- ส่วนมากการเก็บข้อมูลต้องใช้ Annotation tool
- Annotation tool เช่น LabelMe, LabelImg สามารถ export ข้อมูลออกมาอยู่ในรูปแบบของ COCO format ได้



LabelMe Annotation Interface

ตัวอย่างจาก LabelMe

Object detection: COCO format

- **COCO format** เก็บข้อมูลโดยใช้ JSON format ซึ่งประกอบด้วย keys ต่างๆ ได้แก่ info, licenses, images, annotations, categories
- Info บอกถึง metadata ของชุดข้อมูลของเรา
- Images บอกว่ามีภาพอะไรบ้างในชุดข้อมูล
- Annotations เก็บข้อมูลระหว่าง Image ID, Segmentation, bbox ที่บอกว่าวัตถุอยู่ที่ใดในภาพ
- Categories บอกว่า object ชื่ออะไรบ้าง

The following is an example COCO manifest file. For more information, see COCO format.

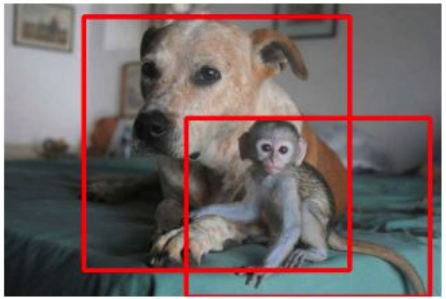
```
{
    "info": {
        "description": "COCO 2017 Dataset","url": "http://cocodataset.org","version": "1.0","ye
    },
    "licenses": [
        {"url": "http://creativecommons.org/licenses/by/2.0/","id": 4,"name": "Attribution Lice
    ],
    "images": [
        {"id": 242287, "license": 4, "coco_url": "http://images.cocodataset.org/val2017/xxxxxx
        {"id": 245915, "license": 4, "coco_url": "http://images.cocodataset.org/val2017/nnnnnnr
    ],
    "annotations": [
        {"id": 125686, "category_id": 0, "iscrowd": 0, "segmentation": [[164.81, 417.51,.....1
        {"id": 1409619, "category_id": 0, "iscrowd": 0, "segmentation": [[376.81, 238.8,.....
        {"id": 1410165, "category_id": 1, "iscrowd": 0, "segmentation": [[486.34, 239.01,.....
    ],
    "categories": [
        {"supercategory": "speaker","id": 0,"name": "echo"},
        {"supercategory": "speaker","id": 1,"name": "echo dot"}
    ]
}
```

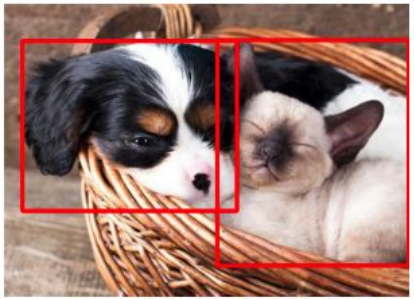
ตัวอย่างของชุดข้อมูลในรูปแบบ COCO Format

https://docs.aws.amazon.com/rekognition/latest/customlabels-dg/md-transform-coco.html

Object detection (example)

- ใช้ Transfer learning เพื่อสร้างโมเดล Object detection สำหรับ detect ภาพหมา/แมว/ลิง จากเว็บไซต์ Kaggle
- เทรนโมเดล Object detection ด้วยไลบรารี่ FastAI, Icevision ด้วยโมเดล Object detection พื้นฐาน





Ref: https://www.kaggle.com/tarunbisht11/yolo-animal-detection-small

AI BUILDERS

Deep learning for object detection and semantic segmentation

Titipat Achakulvisut

Department of Biomedical Engineering,Mahidol University

Kukkik Oparad

425Degree

NLP ไม่ได้มีแค่ตัดคำ

Object detection ไม่ได้มีแค่ YOLO

Semantic Segmentation

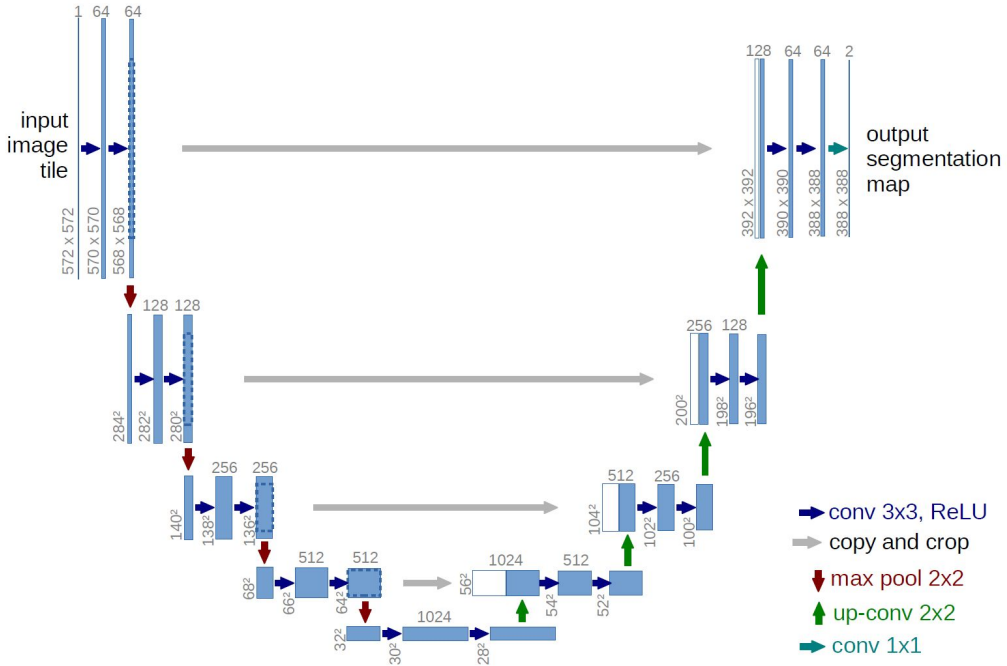
AI BUILDERS

Transfer learning for semantic segmentation

Semantic Segmentation

- Semantic Segmentation เป็นหนึ่งในการทำงานที่มีความสำคัญหนึ่งของการประมวลผลภาพ
- Deep learning เช่น Convolutional Neural Network, CNN มีความสามารถแยกแยะประเภทของพิกเซลในภาพได้ดีมากโมเดลหนึ่ง
- ตัวอย่างของการทำงานเหล่านี้สามารถนำไปใช้ในหลายระบบ เช่น หุ่นยนต์ทำความสะอาดสำหรับ Home navigation หรือรถยนต์ขับเคลื่อนอัจฉริยะสำหรับการแยกแยะถนนหรือรถยนต์ออกจากภาพที่ได้รับมา

Semantic Segmentation: U-Net



Generative model:
Generative Adversarial Network

Titipat Achakulvisut

Department of Biomedical Engineering,Mahidol University

Kukkik Oparad

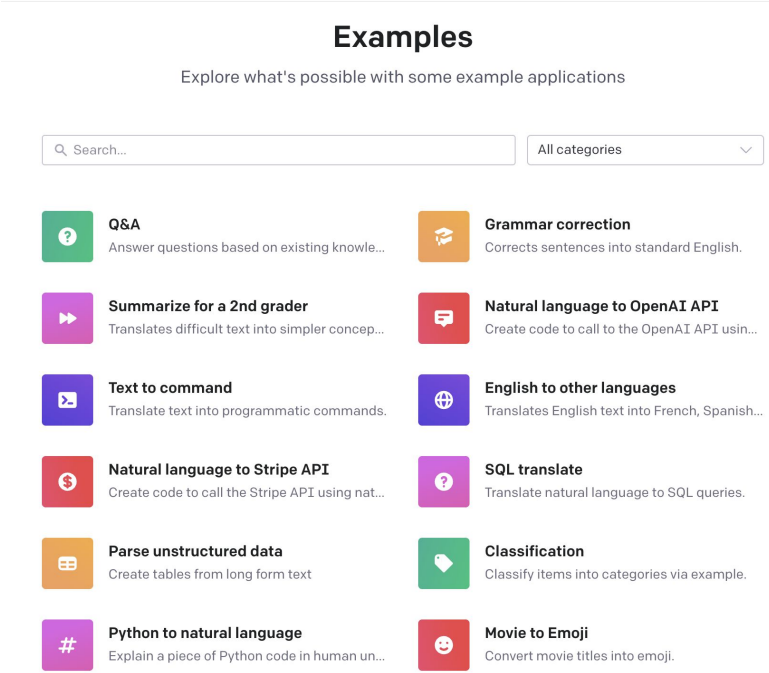
425Degree

ตัวอย่างการใช้งานของ Generative model ในปัจจุบัน

DALL-E 2 created this image in response to the text "teddy bears mixing sparkling chemicals as mad scientists in a steampunk style"

Example of image generated from Dall-E2 model

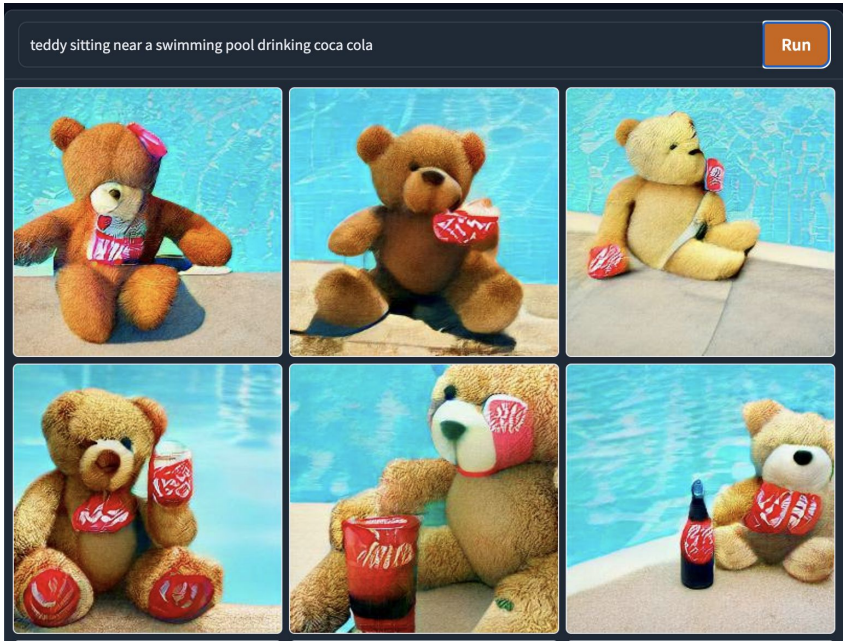
Reference: https://www.dezeen.com/2022/04/21/openai-dall-e-2-unseen-images-basic-text-technology/



Generative model สำหรับภาษา

(https://beta.openai.com/examples)

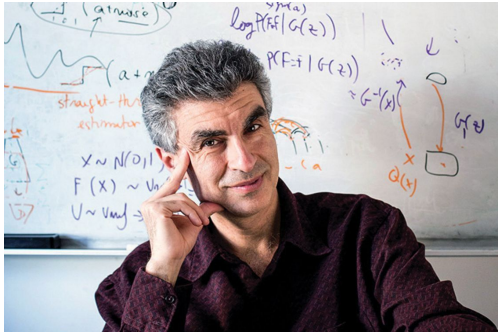
ตัวอย่างการใช้งานของ Generative model ในปัจจุบัน

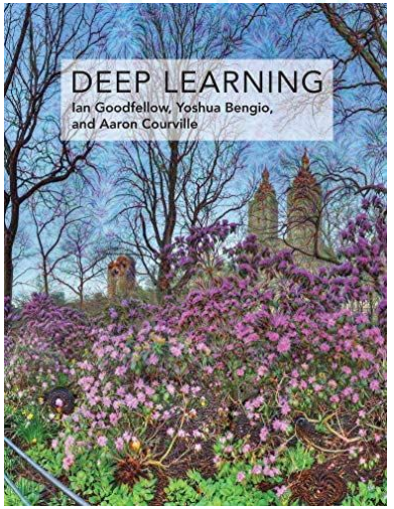


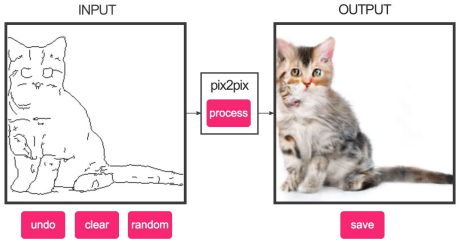
Reference:

- https://medium.com/nerd-for-tech/face-generation-using-generative-adversarial-networks-gan-6d279c2d5759
- https://huggingface.co/spaces/dalle-mini/dalle-mini

Generative Adversarial Network (GAN)







- ในปี 2014, เอียน กู๊ดเฟลโล่ว และทีม (หนึ่งในนั้นคือโยชัว เบนจิโอ) ได้คิดค้นโมเดล Generative adversarial network (GAN) ขึ้นมา
- ทีมค้นพบว่า GAN ที่ใช้ Deep learning มีความสามารถในการสร้างข้อมูลได้ดีมากๆ
- GAN ถูกพัฒนาและคิดค้นมาอย่างต่อเนื่อง ในปัจจุบันมีโมเดลให้เลือกใช้หลากหลายทั้ง GAN, DCGAN, CycleGAN, ... รวมถึงมีการใช้ใน Text generation ด้วยบ้าง
- มีการใช้งานในแอพพลิเคชั่นอื่นๆที่ใกล้เคียงเช่น ร่างภาพแล้วให้โมเดล generate ภาพต่อ (pix2pix)

Ref: https://www.youtube.com/watch?v=pcVQ3UzENAM

GAN อธิบายเบื้องต้น

- GAN ประกอบด้วยโมเดล 2 ส่วนหลักๆคือ Generator ที่ทำหน้าที่สร้างข้อมูลขึ้นมา และ Discriminator ที่บอกว่าข้อมูลนี้เป็นข้อมูลจริงหรือไม่
- ลองนึกภาพถ้าพี่กุกกิกเป็นคนทำธนบัตรปลอม และพี่มายเป็นตำรวจจับธนบัตรปลอม ⟶ หน้าที่ของพี่กุกกิกคือการสร้างแบงค์ที่ดูใกล้เคียงหน้าตาธนบัตรจริงให้มากที่สุด จนกระทั่งพี่มายแยกไม่ออกว่าภาพไหนเป็นธนบัตรจริงหรือปลอม

GAN: Loss function

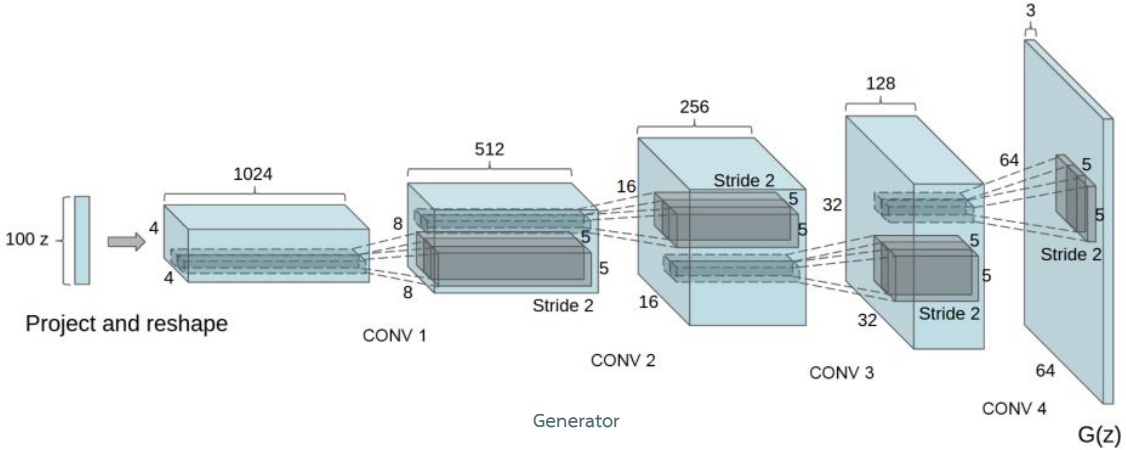
- เนื่องจากมี 2 โมเดลที่เราต้องทำการ optimize นักวิจัยจึงนำเสนอวิธีการที่ชื่อว่า Minimax loss ขึ้นมาซึ่งหน้าตาเป็นดังนี้

$$E_x[log(D(x))] + E_z[log(1 - D(G(z)))]$$

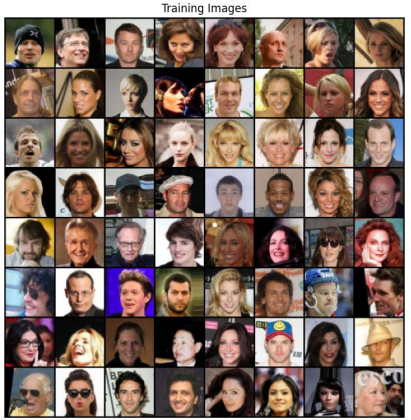
สำหรับ Discriminator ต้องบอกได้ว่าภาพจริงควรมีความน่าจะเป็นที่เป็นภาพจริงสูง เพื่อที่จะ minimize loss

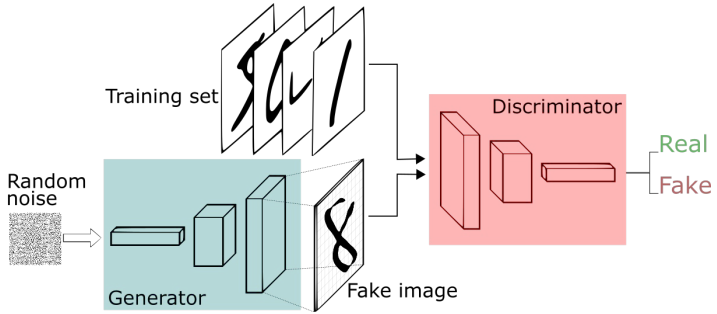
สำหรับ Discriminator ที่ใช้บอกภาพหลอก ส่วนของ loss ต้องใช้ 1 - D(G(z)) แทน เพื่อที่จะ minimize loss

หน้าตาของ GAN Generator



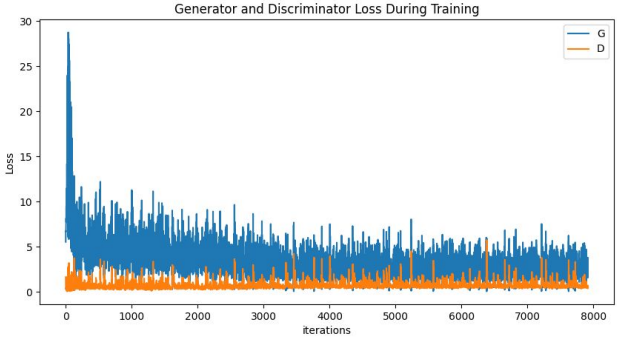
Generator





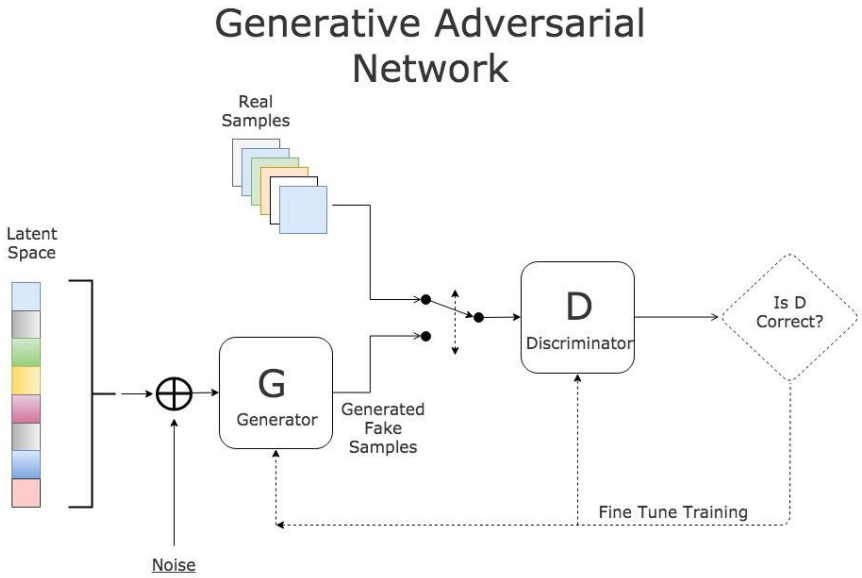
Generator and Discriminator

Generator and discriminator loss



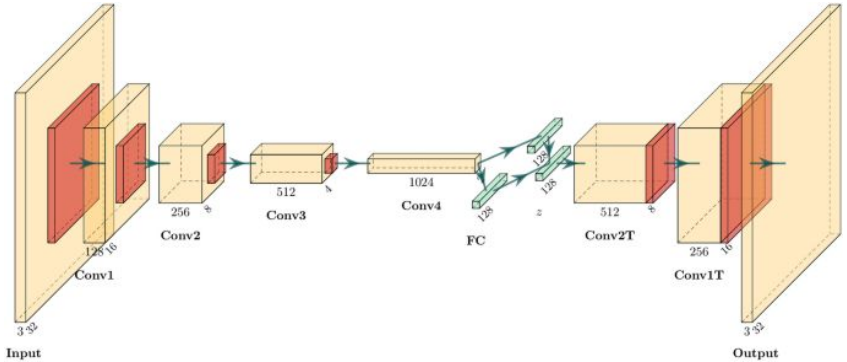
Reference: https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

GAN vs. Variational Autoencoder



Generative Adversarial Network (GAN)

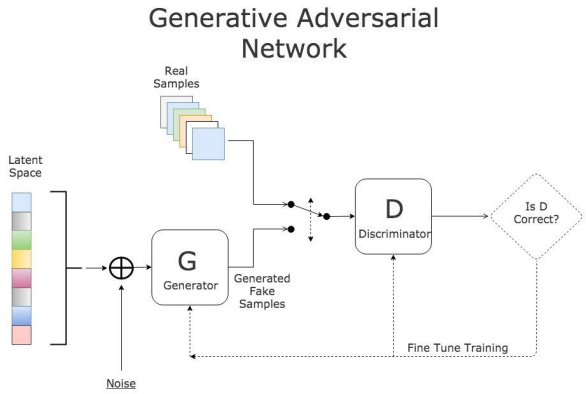
มีโมเดลที่ใช้สำหรับสร้างภาพ และโมเดลที่ใช้แบ่งภาพ



Variational Autoencoder (VAE)

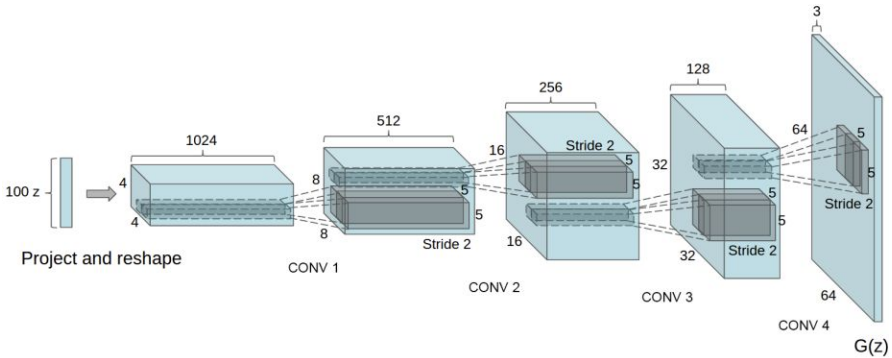
ทำการบีบอัดข้อมูลไปอยู่ใน dimension ขนาดเล็กลงและขยายออกเพื่อให้ได้ข้อมูลเดิม

ข้อสังเกตของการเทรน Deep convolutional GAN (DCGAN)



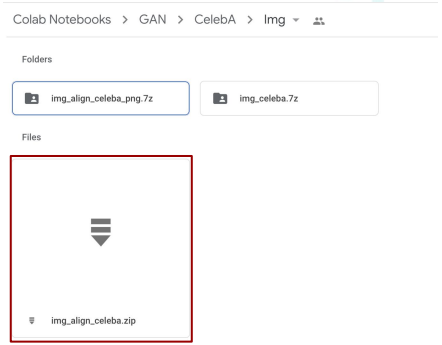
Deep Convolutional Generative Adversarial Network (GAN)

ใช้ Convolutional layer เป็นส่วนประกอบของ Generator



- **DCGAN = Deep Convolutional GAN** เป็นการขยายร่างของ GAN ที่ใช้ Convolutional และ Convolutional transpose layers เป็นองค์ประกอบของโมเดล
- จากเปเปอร์ของ DCGAN มีการใช้ ConvTranspose2d สำหรับการขยายมิติและใช้ LeakyReLU เป็น Non-linear activation function สำหรับ Discriminator
- เปเปอร์กล่าวว่า Conv2d, LeakyReLU, BatchNorm มีส่วนสำคัญในการคำนวณ gradient ในกรณีการสร้าง Discriminator ของ DCGAN
- ในเปเปอร์เลือกใช้ strided convolution แทนที่จะใช้ max pooling เนื่องจากเป็นการทำให้ Discriminator เรียนรู้ pooling function ด้วยตัวเอง
- สำหรับน้องๆที่ต้องใช้ GAN ในโปรเจกต์ แนะนำให้อ่าน Architecture จากเปเปอร์ร่วมกับ Mentor ก่อนเนื่องจากมีความจำเพาะมากๆ (และมีการสืบหา)

สร้างภาพจากจากชุดข้อมูล CelebA



โหลดภาพจาก folder: img_align_celeba.zip,
Unzip และวางไว้ใน folder ที่สามารถ access ได้

โหลดข้อมูลได้จาก http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html