Object detection with FasterRCNN

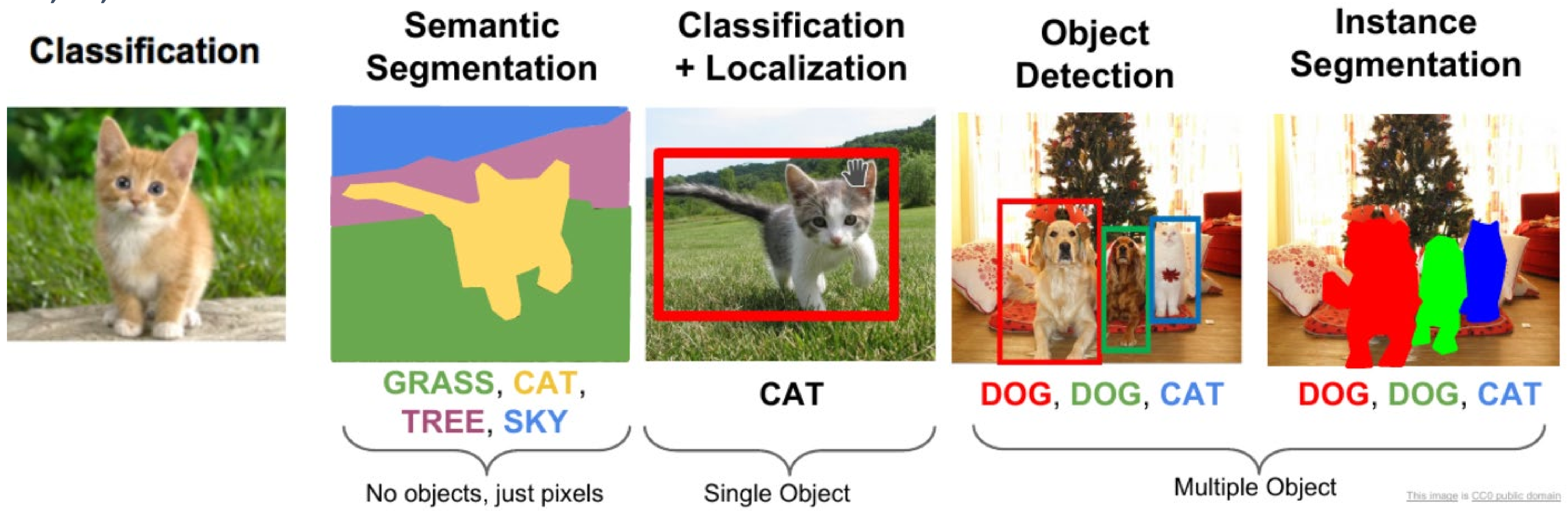
Computer vision tasks

6, 7: CNN
8, 9, 10: Alex Net
HW: VGG16
11, 12, HW5: Res Net

15: U Net

13, 14: Faster RCNN

Mask RCNN

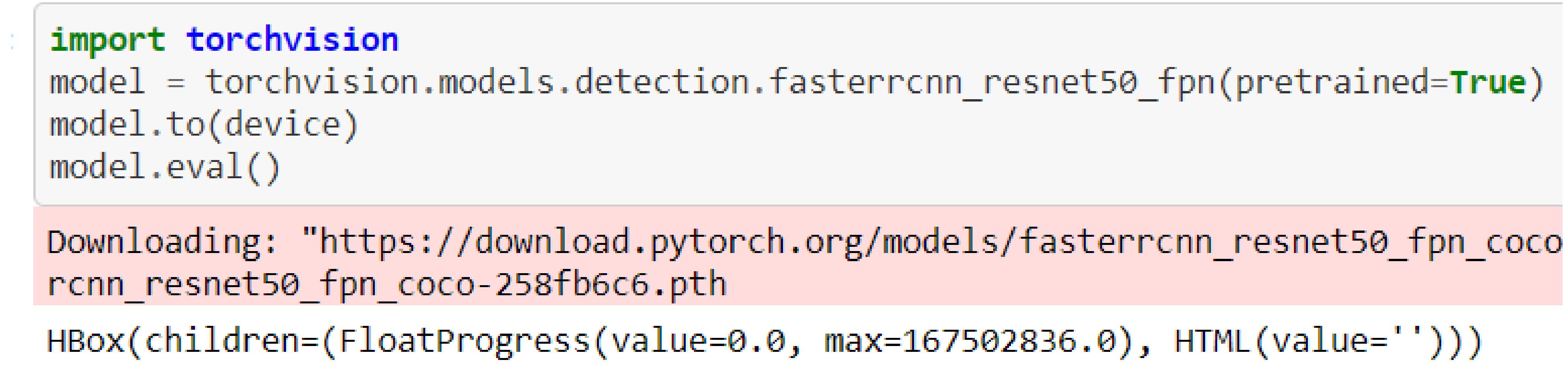


圖片來源: https://kharshit.github.io/blog/2019/08/23/quick-intro-to-instance-segmentation

Practice

- Run “8.1. Faster RCNN.ipynb"

Load pre-trained FasterRCNN



```
import torchvision
model = torchvision.models.detection.fasterrcnn_resnet50_fpn(pretrained=True)
model.to(device)
model.eval()
```

```
Downloading: "https://download.pytorch.org/models/fasterrcnn_resnet50_fpn_coco
rcnn_resnet50_fpn_coco-258fb6c6.pth

HBox(children=(FloatProgress(value=0.0, max=167502836.0), HTML(value='')))
```

How FasterRCNN works?

https://www.youtube.com/watch?v=4yOcsWg-7g8

Practice

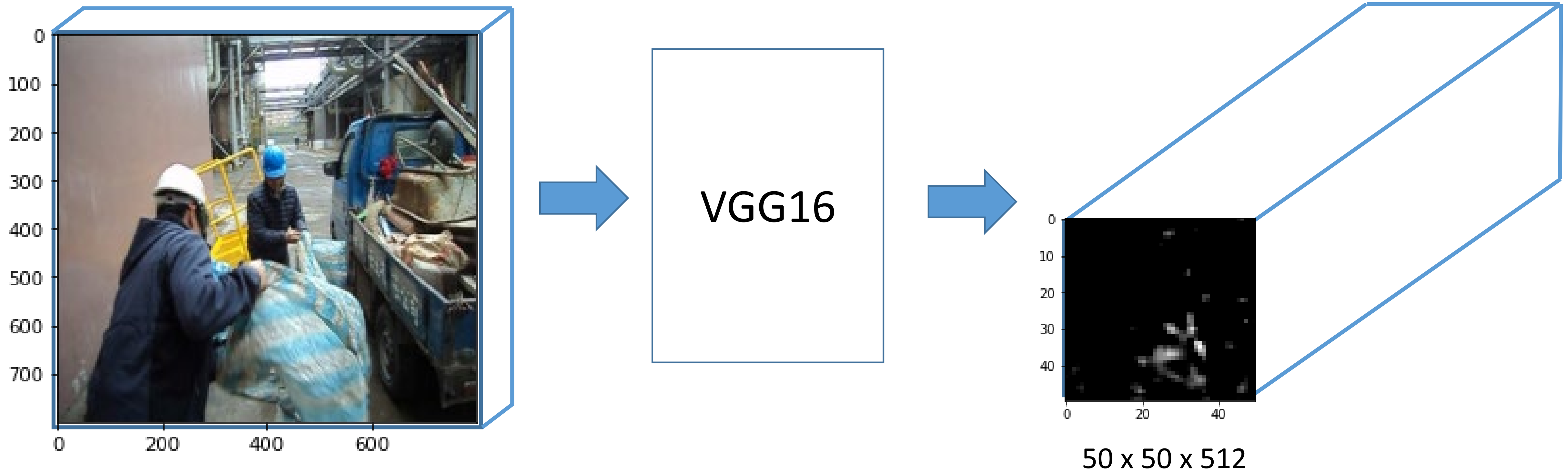
- Run “8.2. FasterRCNN step by step.ipynb"

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

Advances in neural information processing systems 2015 (pp. 91-99).

Feature maps

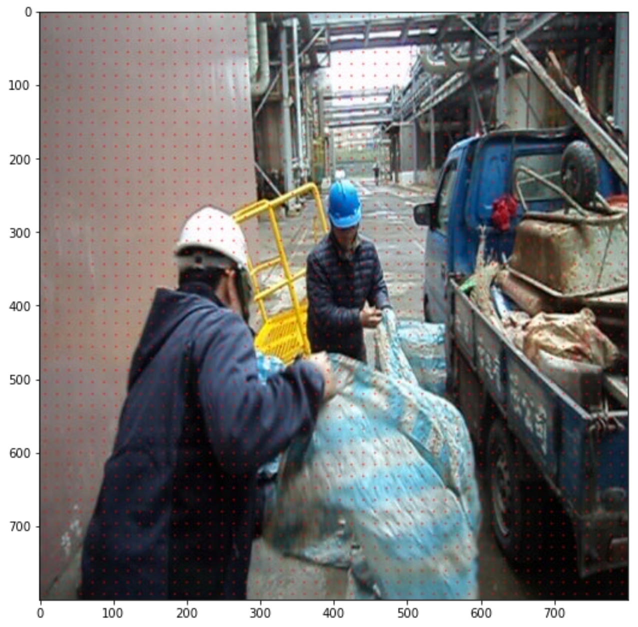


800 x 800 x 3

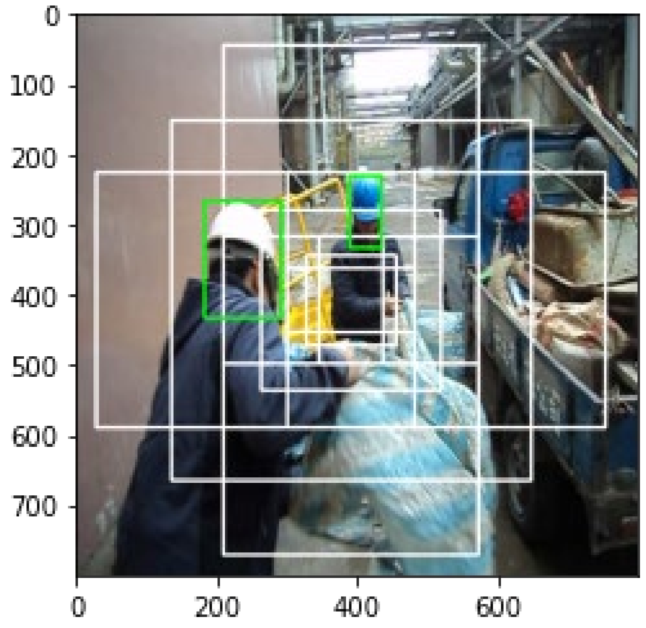
50 x 50 x 512

Anchor boxes

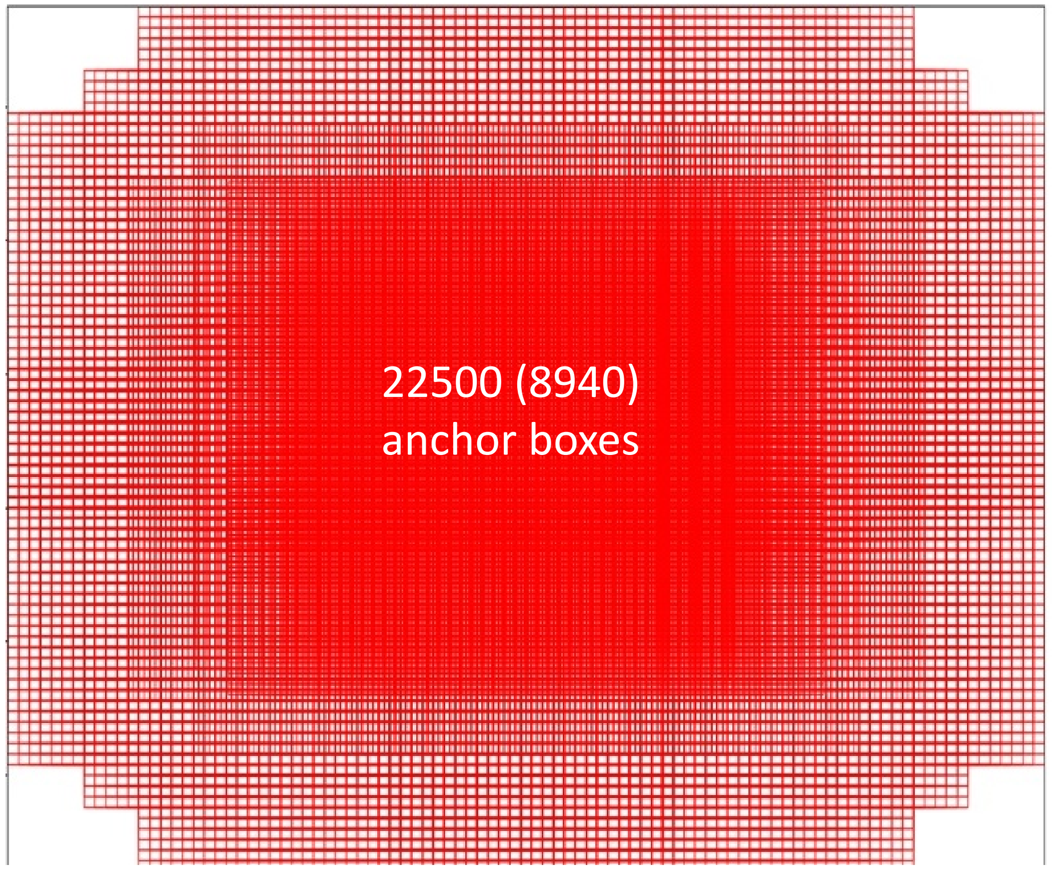
Total number of anchors = 16*16=2500



9 anchor boxes are generated at an anchor point

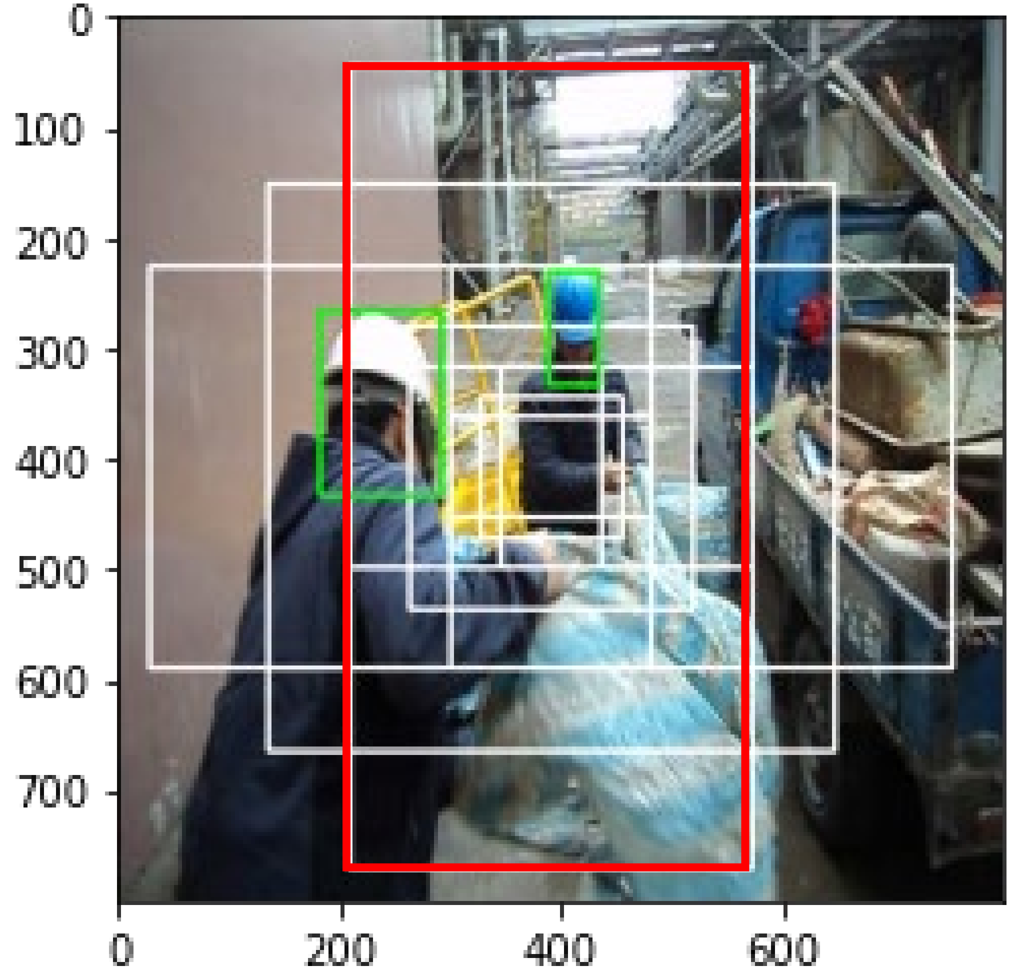


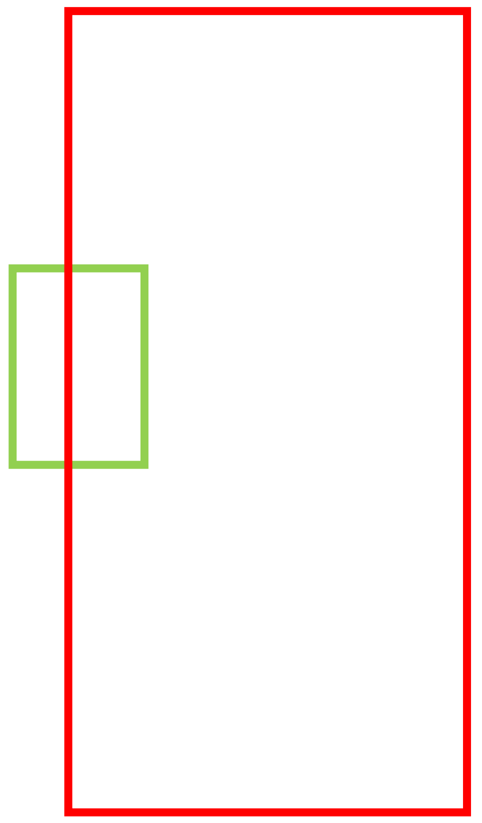
Total number of anchor boxes = 16*16*9



https://zhuanlan.zhihu.com/p/31426458

Intersection over union (IOU)

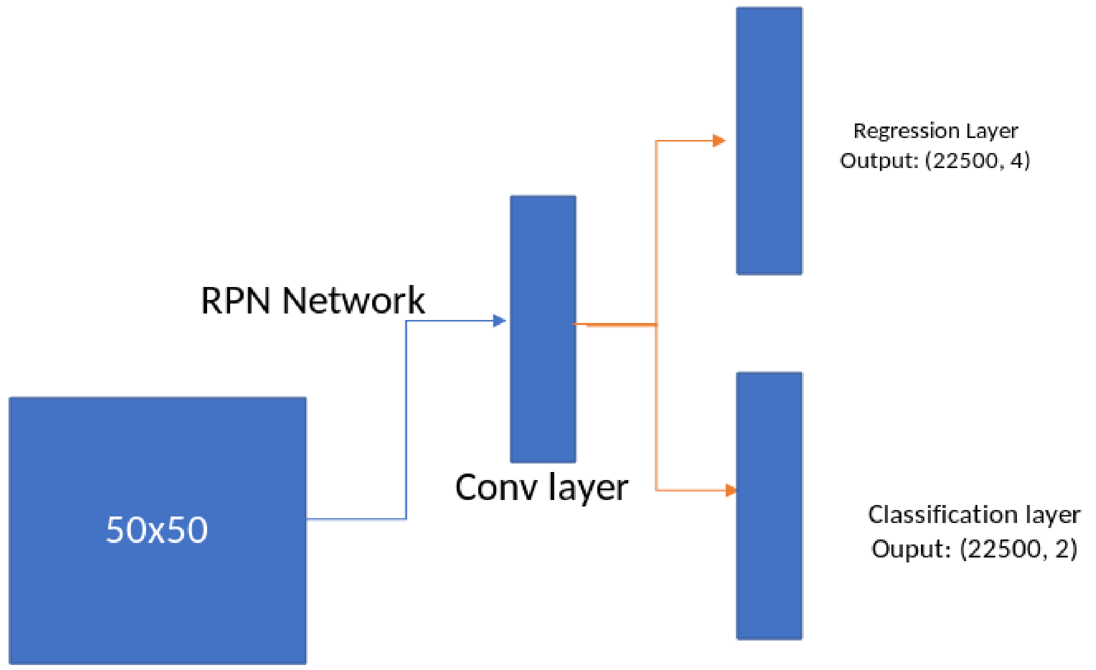




$$\text{iou} = \frac{intersection}{union}$$

RPN

```
(rpn): RegionProposalNetwork(
  (anchor_generator): AnchorGenerator()
  (head): RPNHead(
    (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (cls_logits): Conv2d(256, 3, kernel_size=(1, 1), stride=(1, 1))
    (bbox_pred): Conv2d(256, 12, kernel_size=(1, 1), stride=(1, 1))
  )
```



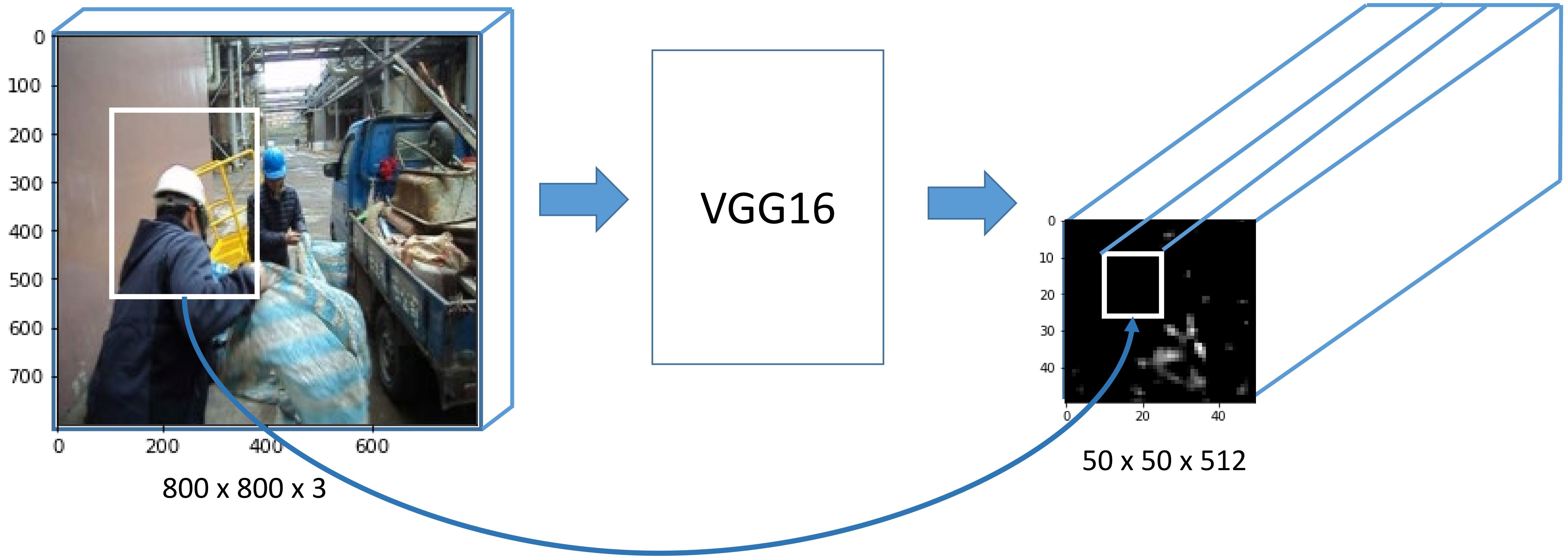
https://medium.com/@fractaldle/guide-to-build-faster-rcnn-in-pytorch-95b10c273439

RPN Loss

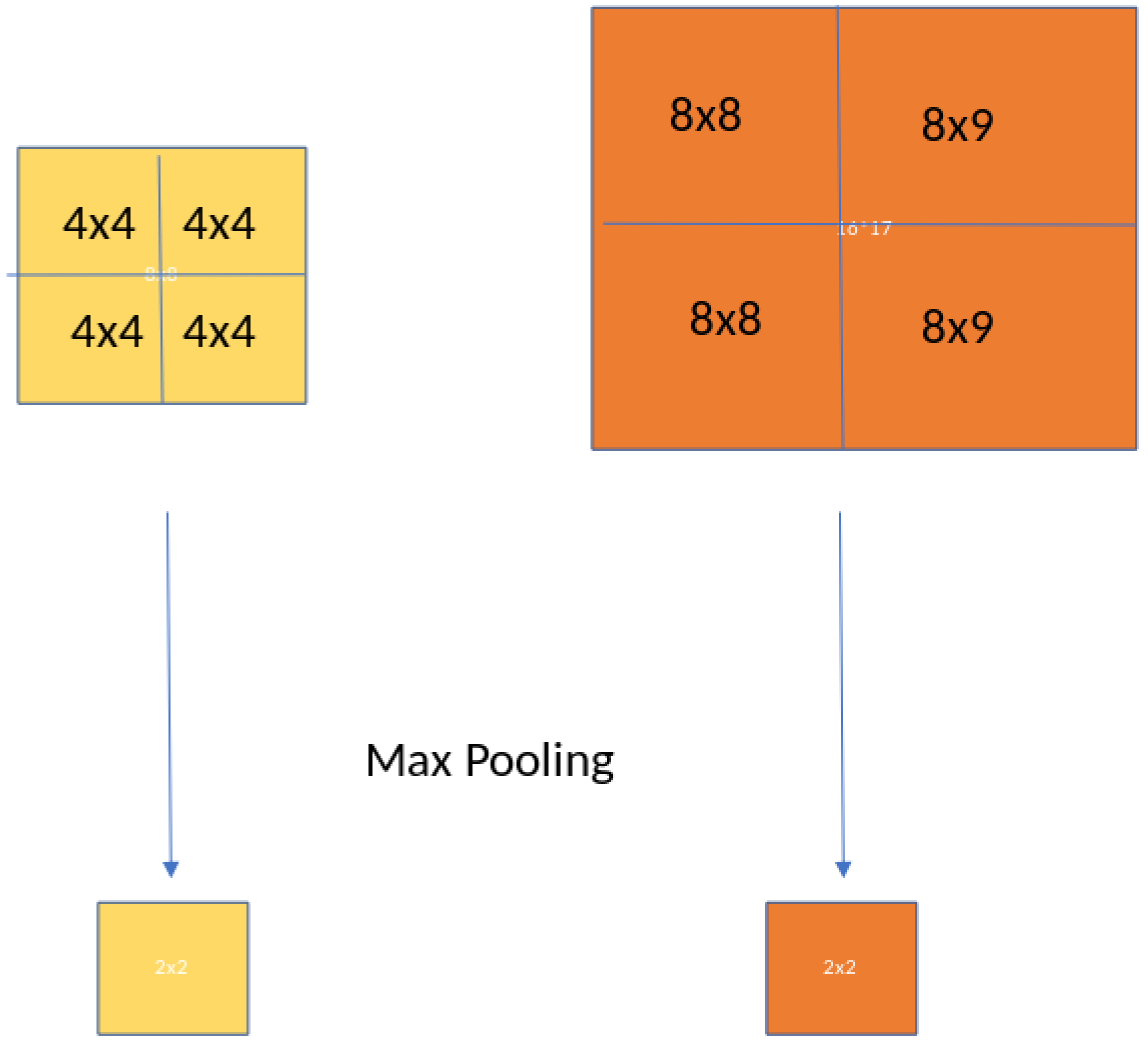
$$L(p_i, t_i) = (1 / N_{cls}) * \sum_i L_{cls}(p_i, p_i^*) + \lambda * (1 / N_{reg}) * \sum_i p_i^* L_{reg}(t_i, t_{i*})$$

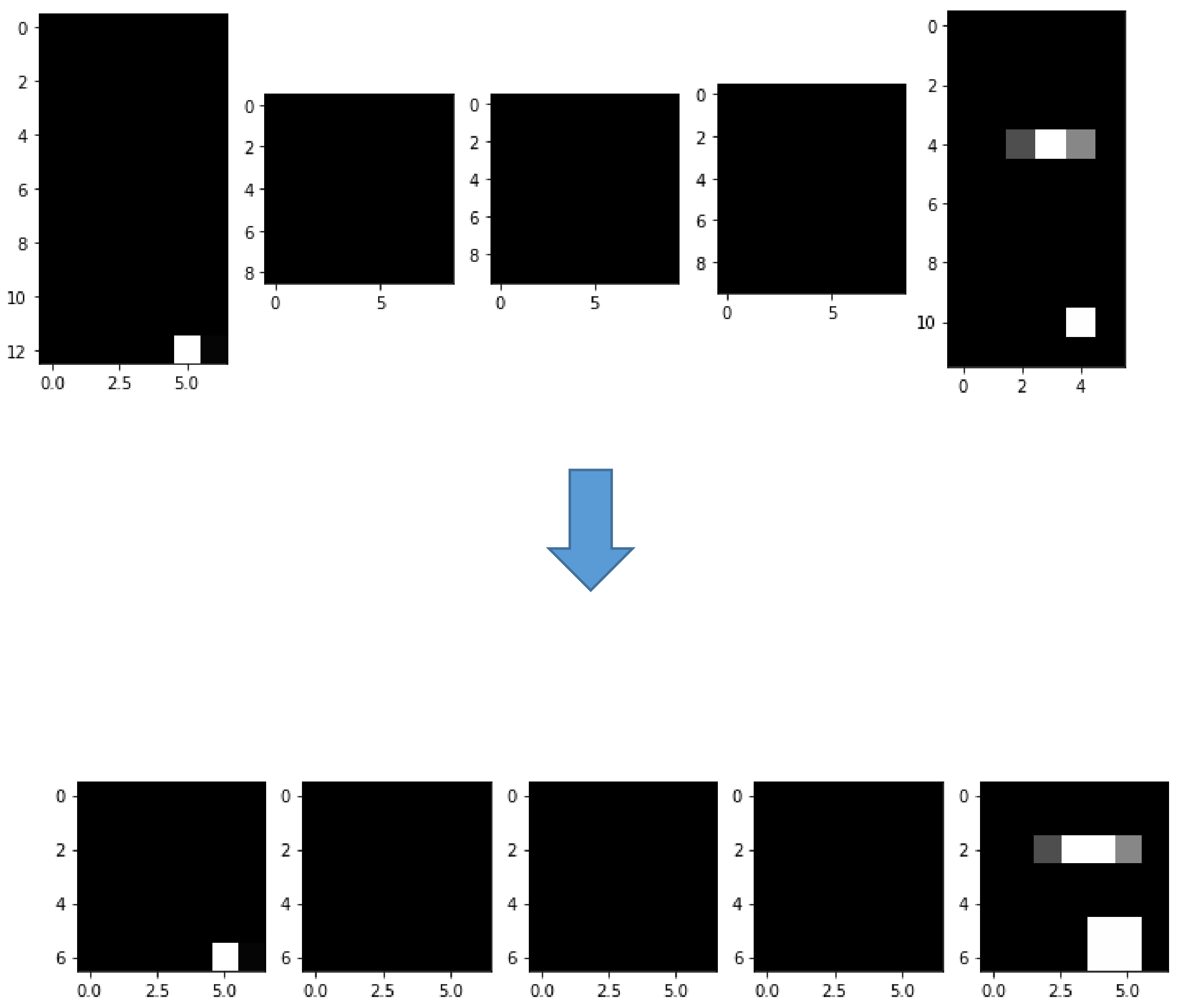
https://medium.com/@fractaldle/guide-to-build-faster-rcnn-in-pytorch-95b10c273439

Feature maps of ROI samples



ROI pooling





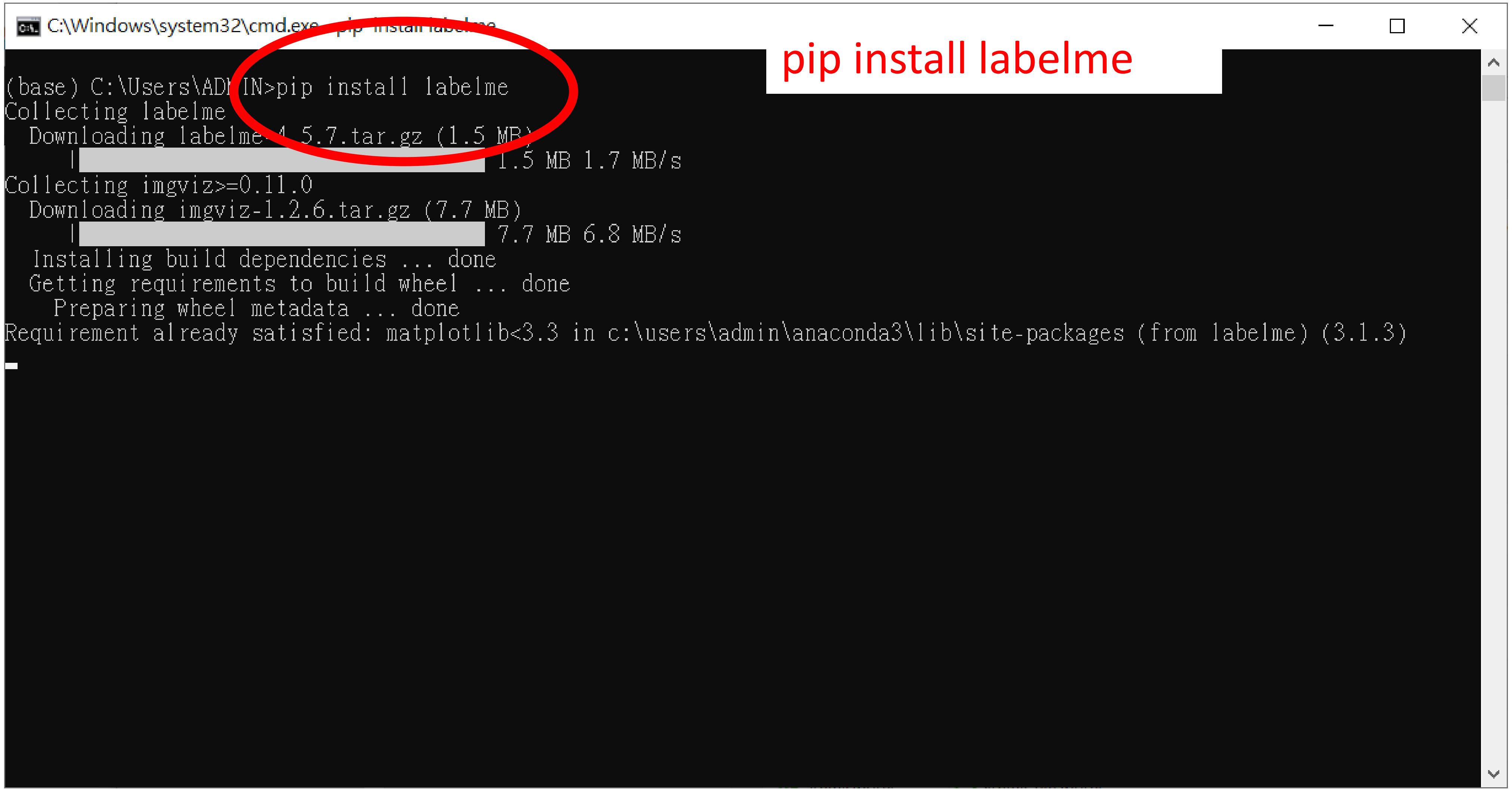
https://medium.com/@fractaldle/guide-to-build-faster-rcnn-in-pytorch-95b10c273439

Detection network

```
(roi_heads): RoIHeads(
  (box_roi_pool): MultiScaleRoIAlign()
  (box_head): TwoMLPHead(
    (fc6): Linear(in_features=12544, out_features=1024, bias=True)
    (fc7): Linear(in_features=1024, out_features=1024, bias=True)
  )
  (box_predictor): FastRCNNPredictor(
    (cls_score): Linear(in_features=1024, out_features=91, bias=True)
    (bbox_pred): Linear(in_features=1024, out_features=364, bias=True)
  )
```

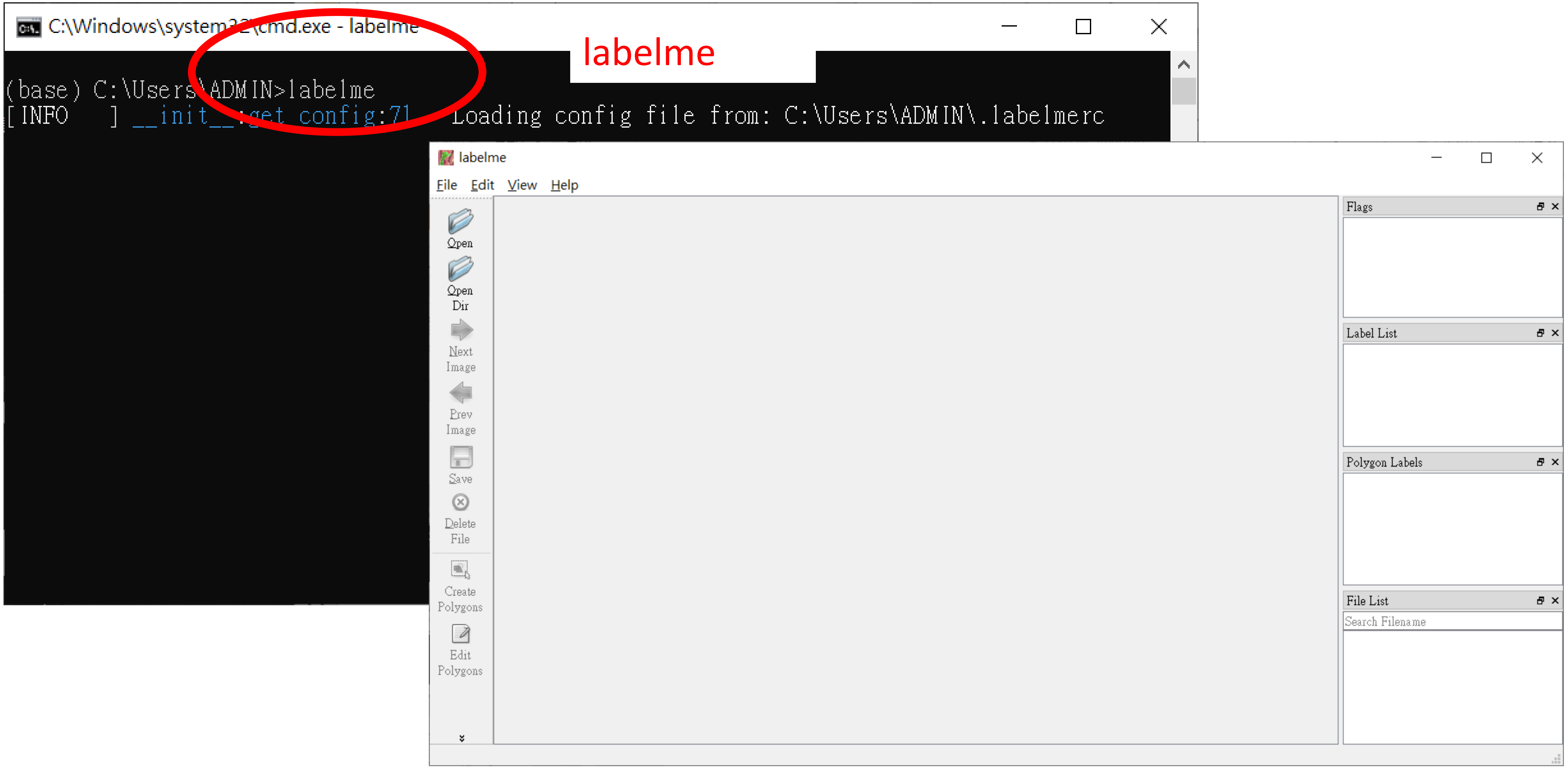
Fine tune to detect our own object

pip install labelme in your Anaconda environment



```
(base) C:\Users\ADMIN>pip install labelme
Collecting labelme
  Downloading labelme-4.5.7.tar.gz (1.5 MB)
     |████████████████████████████████| 1.5 MB 1.7 MB/s
Collecting imgviz>=0.11.0
  Downloading imgviz-1.2.6.tar.gz (7.7 MB)
     |████████████████████████████████| 7.7 MB 6.8 MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
    Preparing wheel metadata ... done
Requirement already satisfied: matplotlib<3.3 in c:\users\admin\anaconda3\lib\site-packages (from labelme) (3.1.3)
```

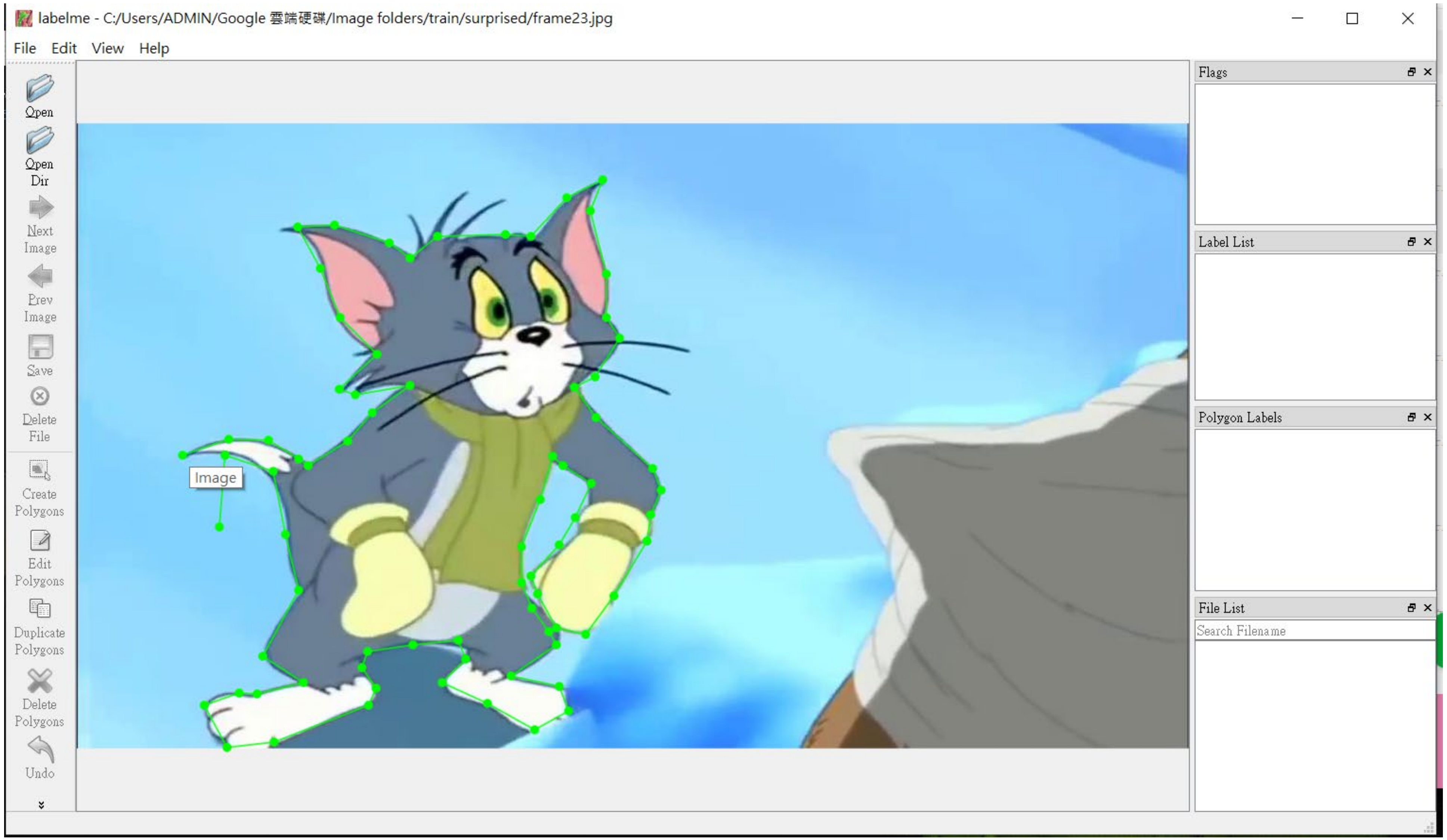
Run labelme



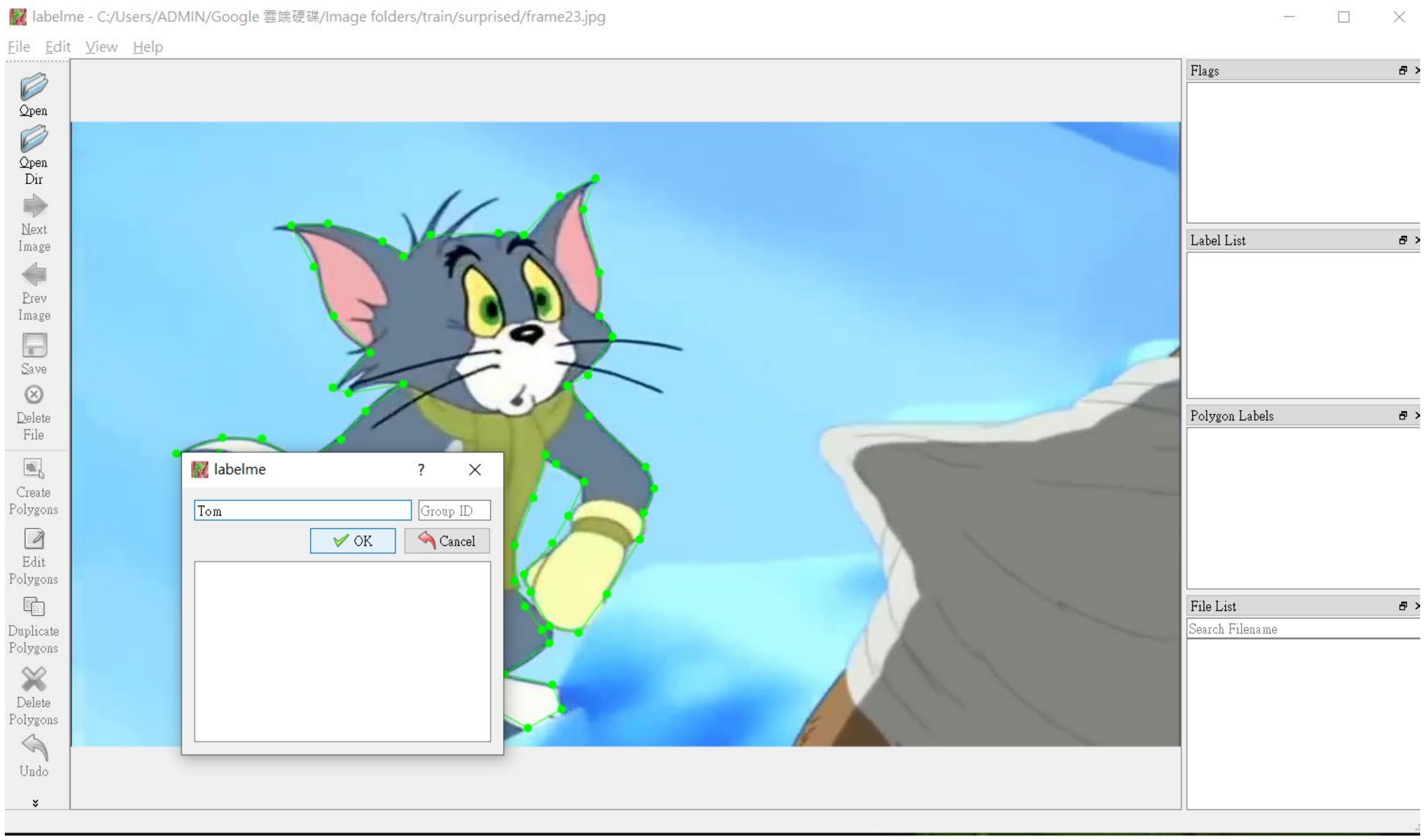
labelme

```
C:\Windows\system32\cmd.exe - labelme
(base) C:\Users\ADMIN>labelme
[INFO   ] __init__:get_config:71 - Loading config file from: C:\Users\ADMIN\.labelmerc
```

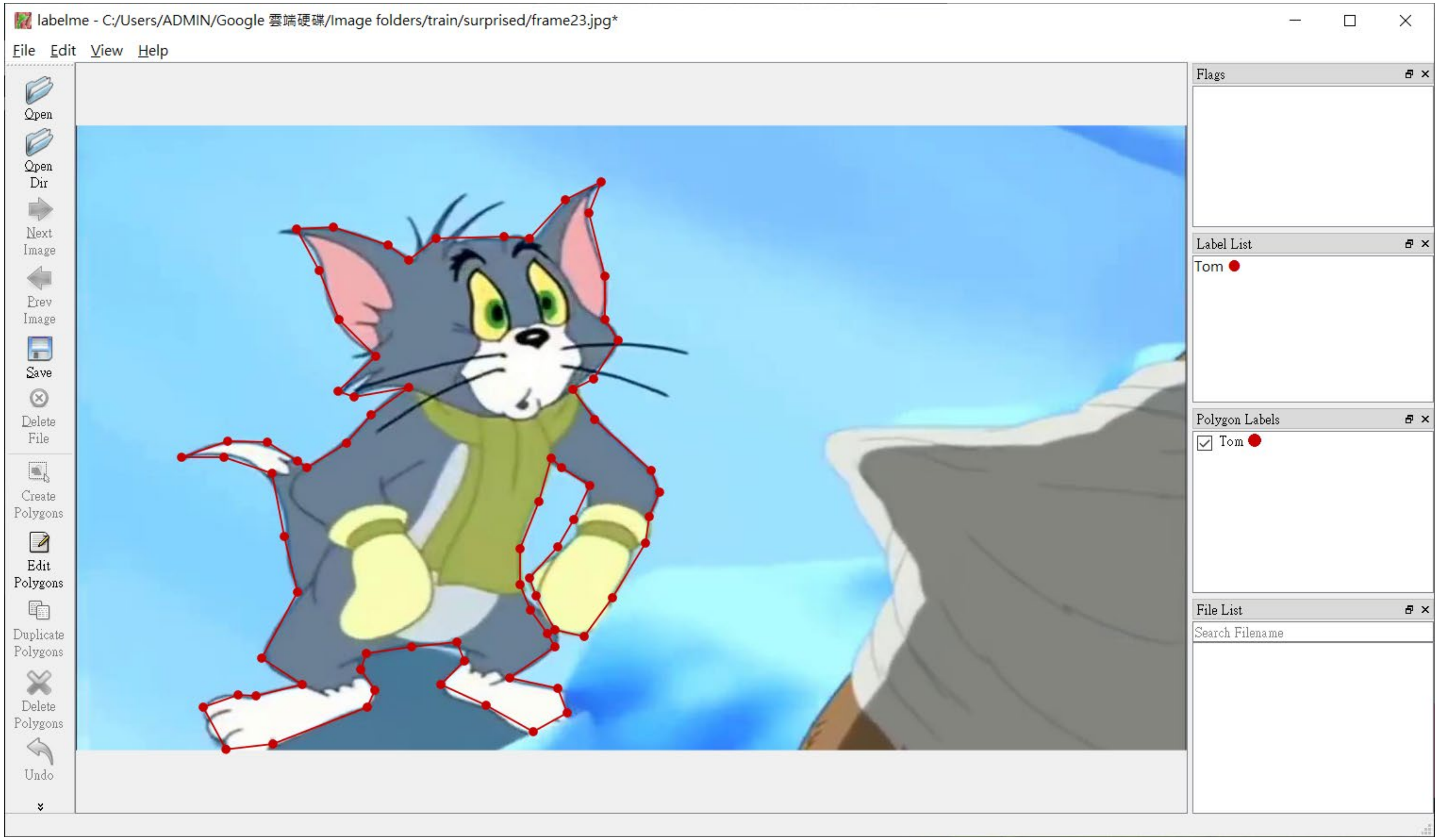
Load an image and draw boundary



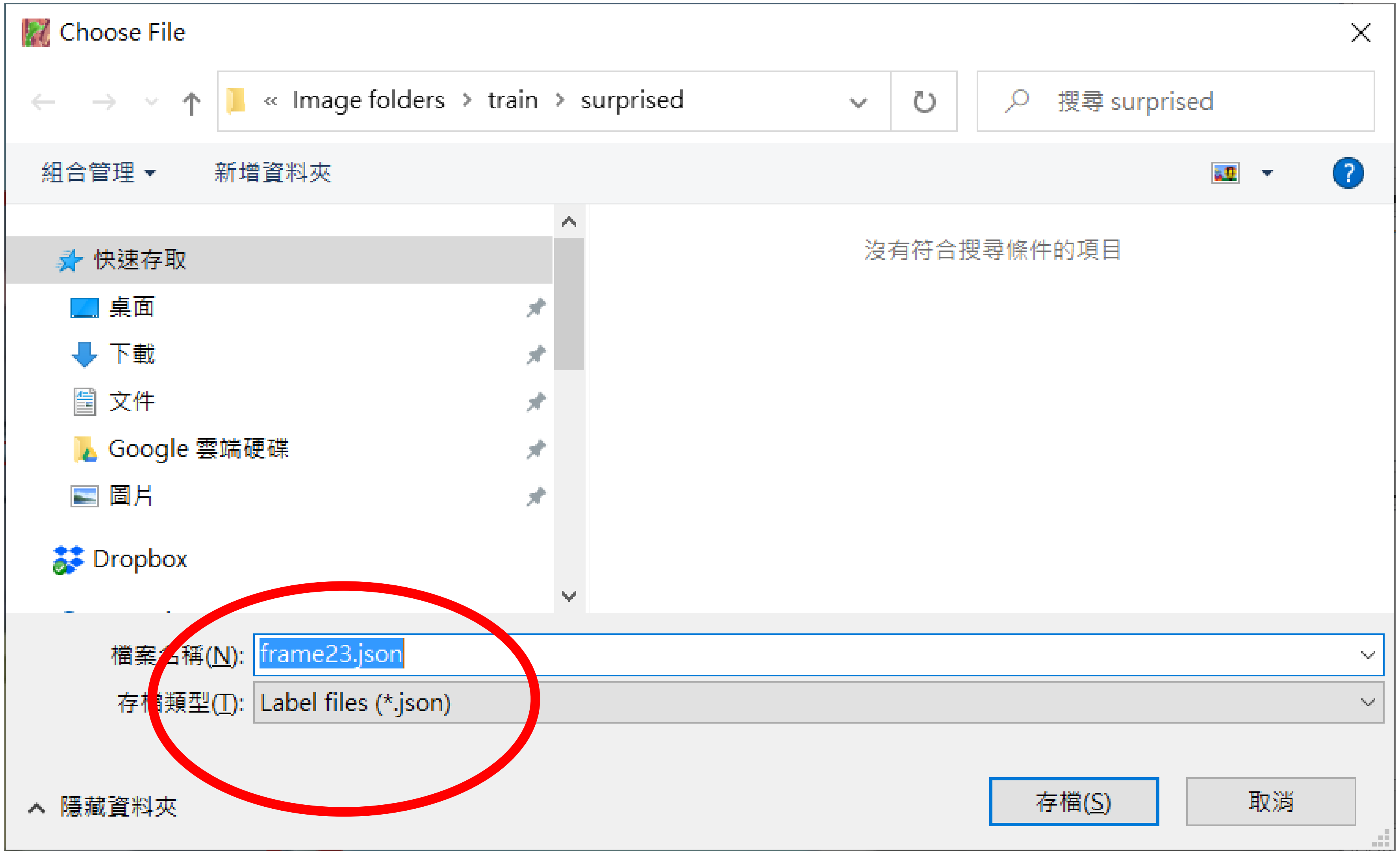
Save label


labelme - C:/Users/ADMIN/Google 雲端硬碟/Image folders/train/surprised/frame23.jpg
File
Edit
View
Help
Open
Open Dir
Next Image
Prev Image
Save
Delete File
Create Polygons
Edit Polygons
Duplicate Polygons
Delete Polygons
Undo
Flags
Label List
Polygon Labels
File List
Search Filename
labelme
Tom
Group ID
OK
Cancel

Saved label


labelme - C:/Users/ADMIN/Google 雲端硬碟/Image folders/train/surprised/frame23.jpg*
File
Edit
View
Help
Open
Open Dir
Next Image
Prev Image
Save
Delete File
Create Polygons
Edit Polygons
Duplicate Polygons
Delete Polygons
Undo
Flags
Label List
Tom
Polygon Labels
Tom
File List
Search Filename

Save boundary to json file


Choose File

« Image folders › train › surprised

搜尋 surprised

組合管理 ▾　新增資料夾

快速存取
桌面
下載
文件
Google 雲端硬碟
圖片
Dropbox

沒有符合搜尋條件的項目

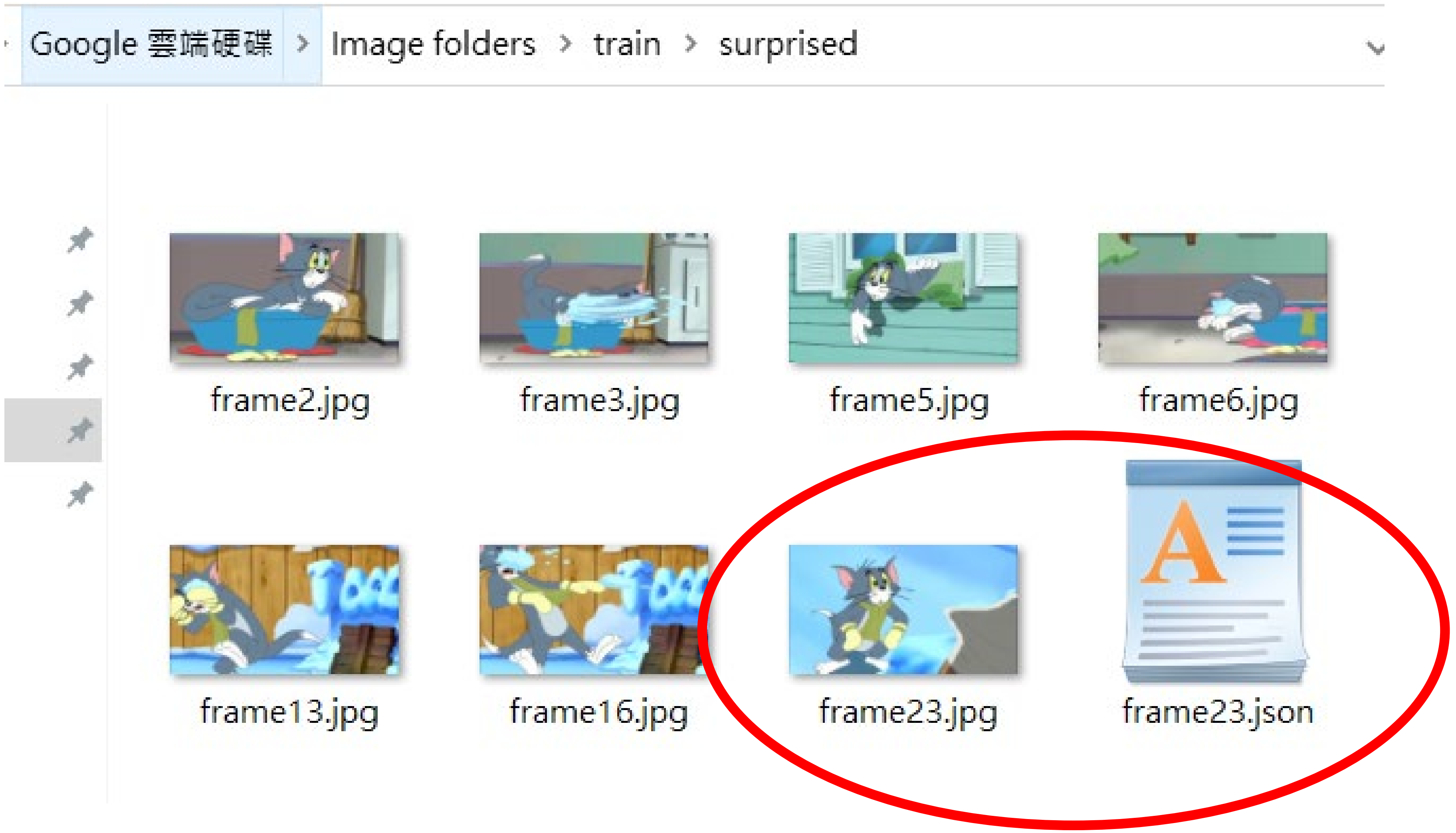
檔案名稱(N): frame23.json

存檔類型(T): Label files (*.json)

隱藏資料夾

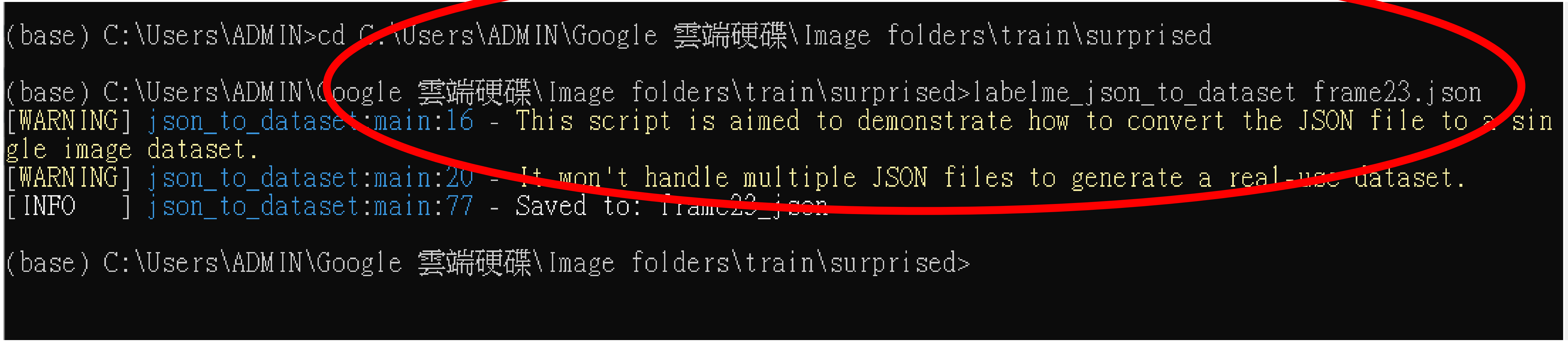
存檔(S)　取消

Saved json file



Convert json file to mask image

cd to the folder where you save the *.json file
Labelme_json_to_dataset *.json

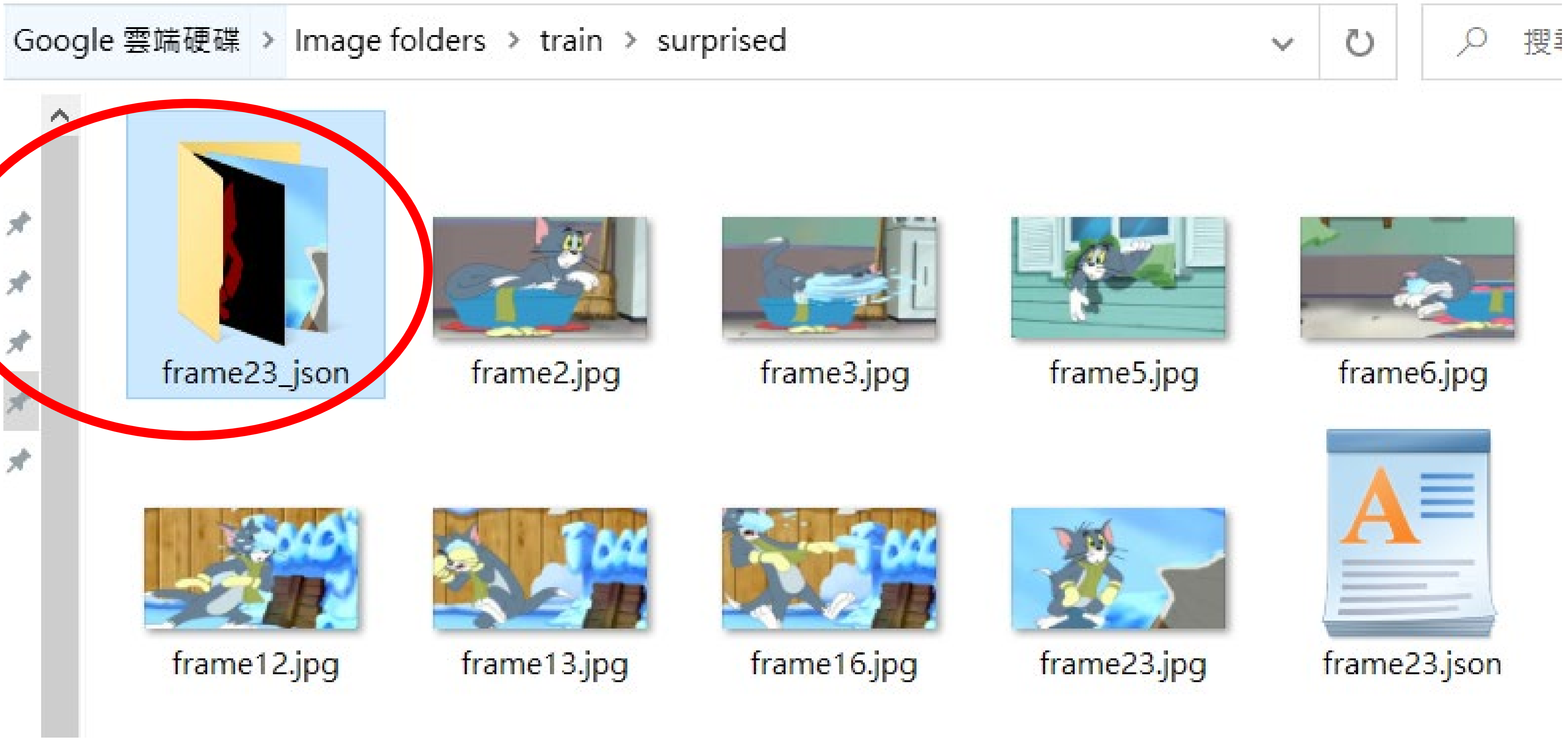


```
(base) C:\Users\ADMIN>cd C:\Users\ADMIN\Google 雲端硬碟\Image folders\train\surprised

(base) C:\Users\ADMIN\Google 雲端硬碟\Image folders\train\surprised>labelme_json_to_dataset frame23.json
[WARNING] json_to_dataset:main:16 - This script is aimed to demonstrate how to convert the JSON file to a sin
gle image dataset.
[WARNING] json_to_dataset:main:20 - It won't handle multiple JSON files to generate a real-use dataset.
[ INFO   ] json_to_dataset:main:77 - Saved to: frame23_json

(base) C:\Users\ADMIN\Google 雲端硬碟\Image folders\train\surprised>
```

Mask images are saved in a folder



Google 雲端硬碟 › Image folders › train › surprised

搜尋

frame23_json

frame2.jpg

frame3.jpg

frame5.jpg

frame6.jpg

frame12.jpg

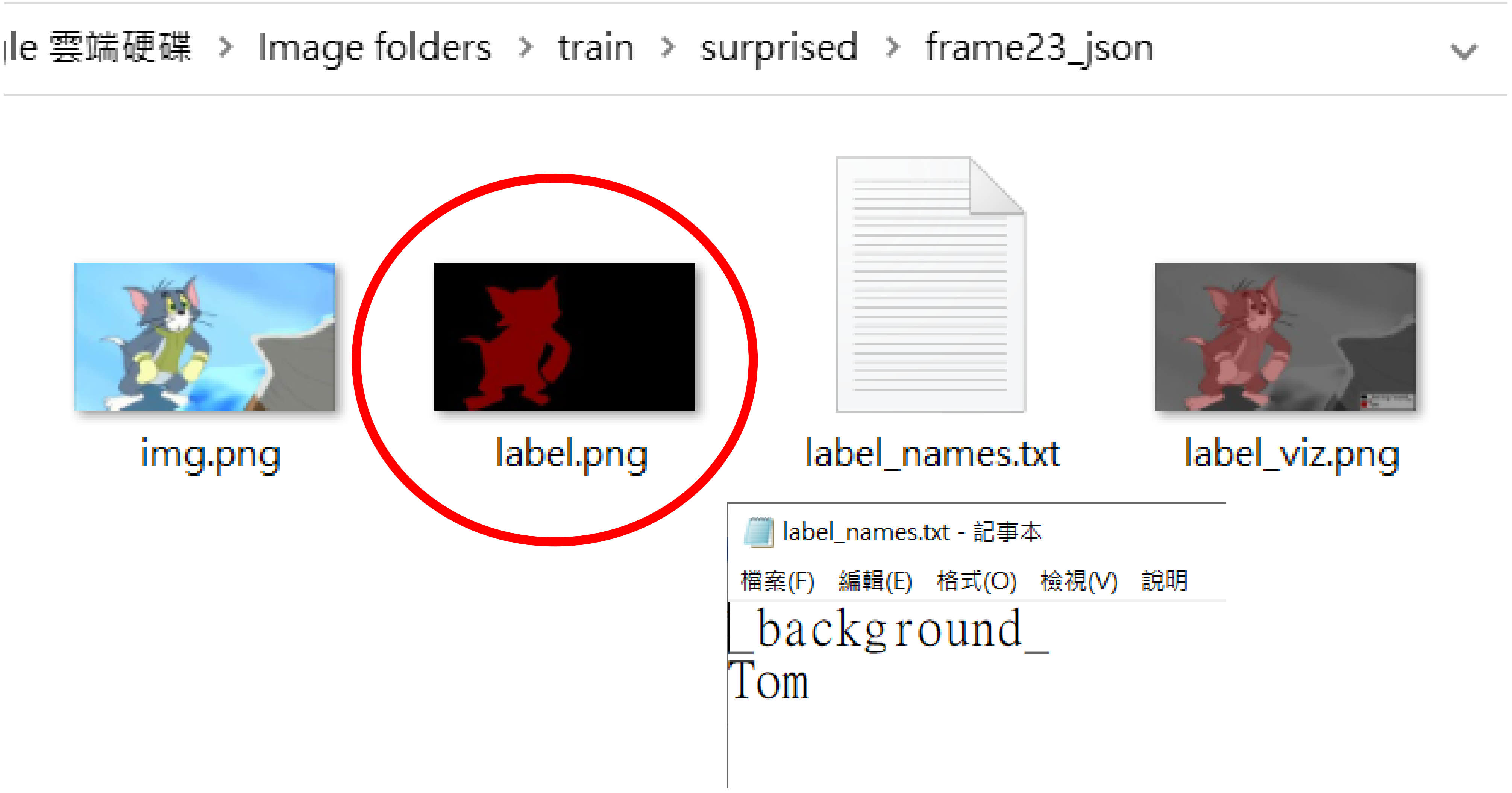
frame13.jpg

frame16.jpg

frame23.jpg

frame23.json

Mask image



ıle 雲端硬碟 › Image folders › train › surprised › frame23_json

img.png

label.png

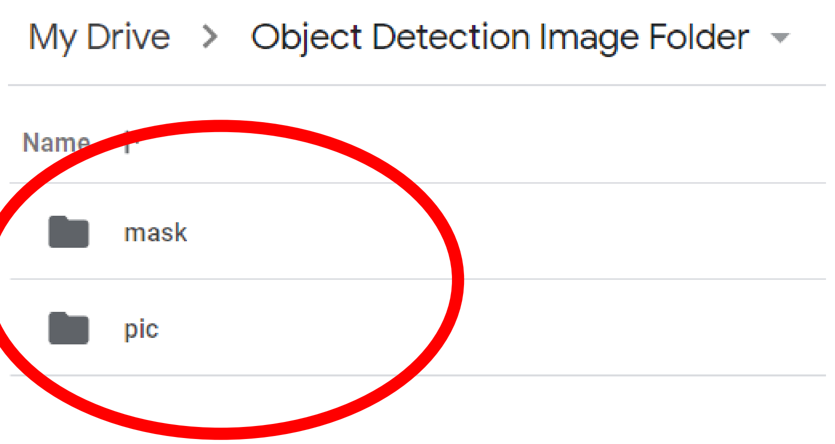
label_names.txt

label_viz.png

label_names.txt - 記事本

```
_background_
Tom
```

Save RGB and mask images on your Google drive

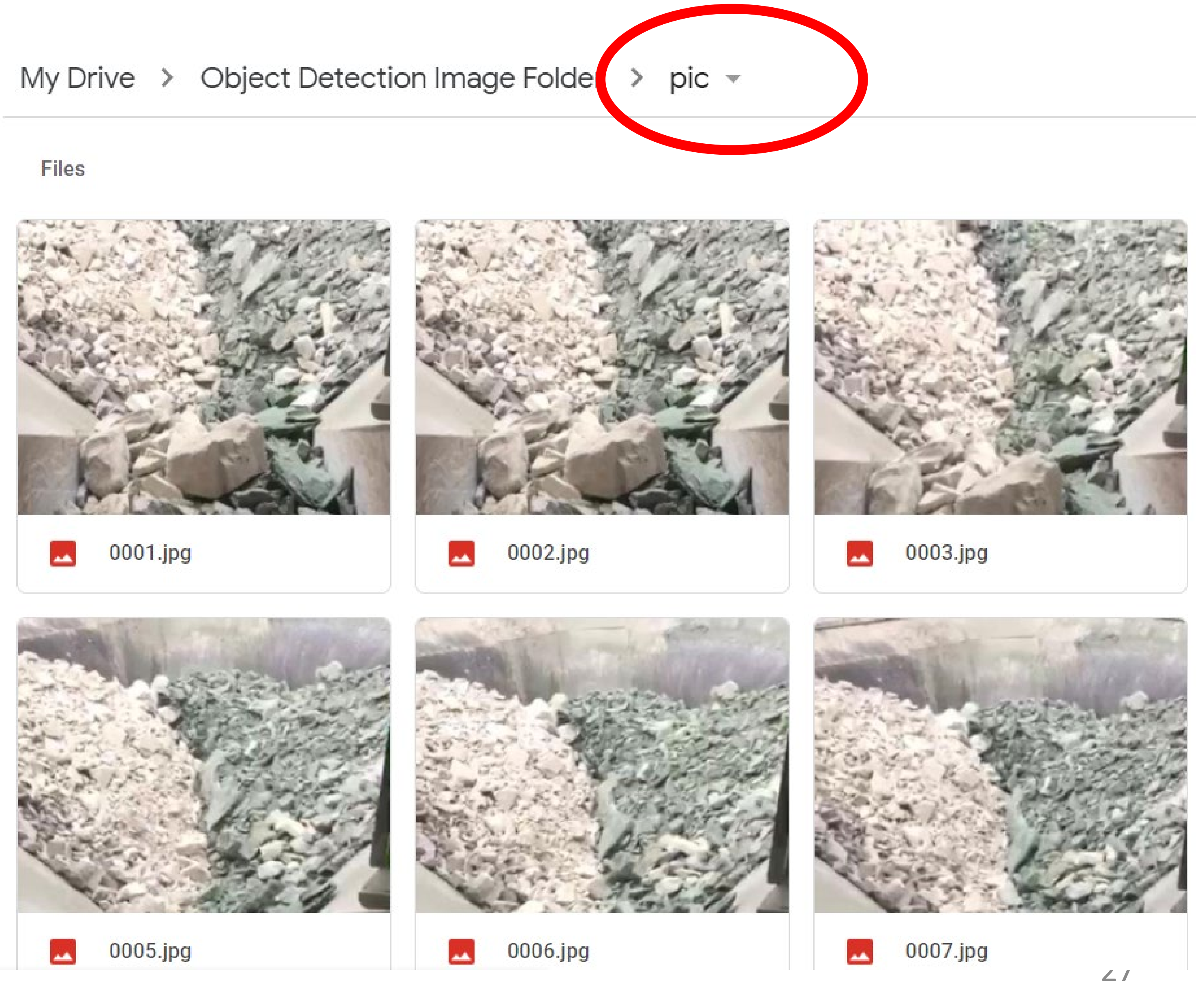


My Drive > Object Detection Image Folder

Name

mask

pic



My Drive > Object Detection Image Folder > pic

Files

0001.jpg

0002.jpg

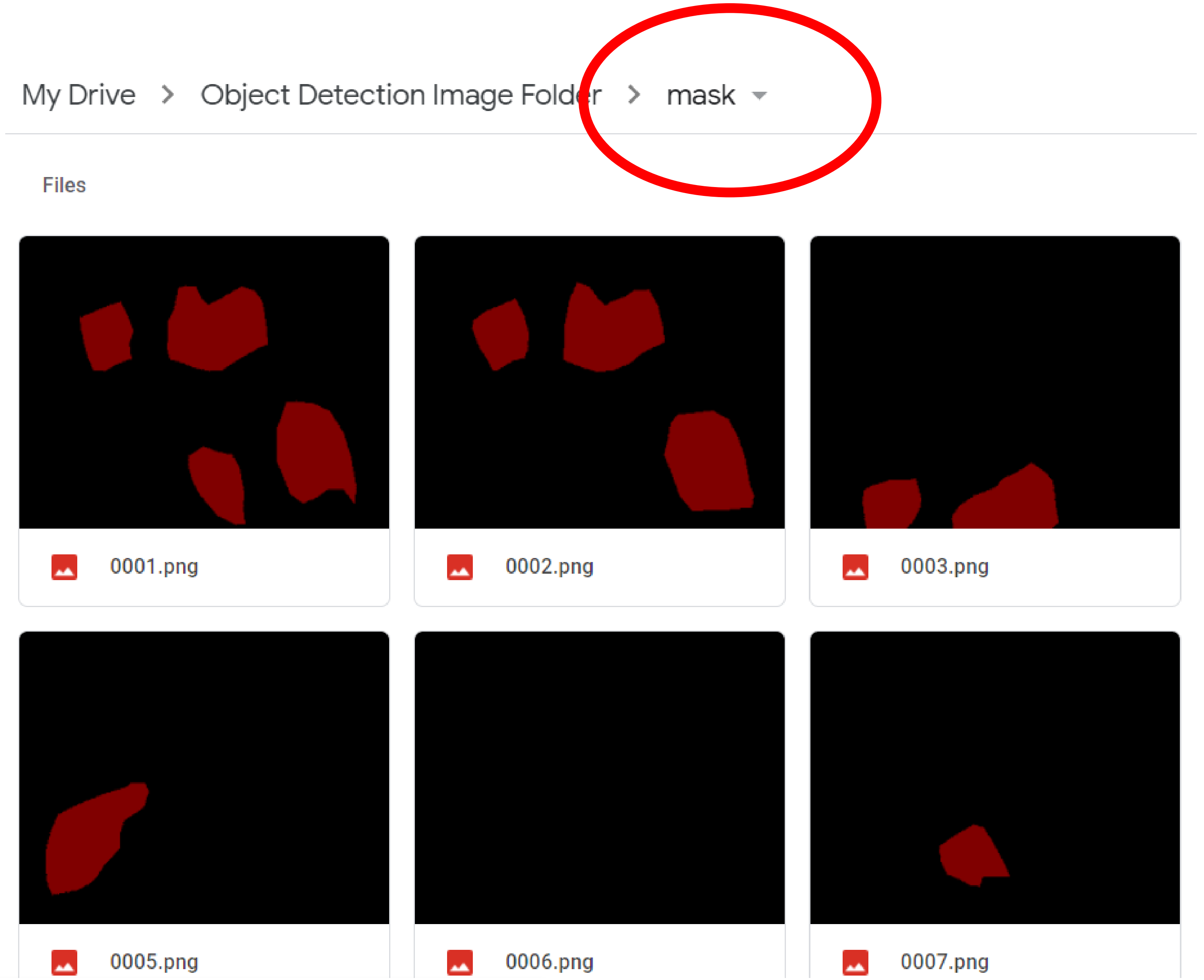
0003.jpg

0005.jpg

0006.jpg

0007.jpg

Save RGB and mask images on your Google drive



My Drive > Object Detection Image Folder > mask

Files

0001.png

0002.png

0003.png

0005.png

0006.png

0007.png

Practice

- Run “8.3 FasterRCNN fine tune.ipynb"

HW7

- Fine tune FasterRCNN to detect eyes or noses.