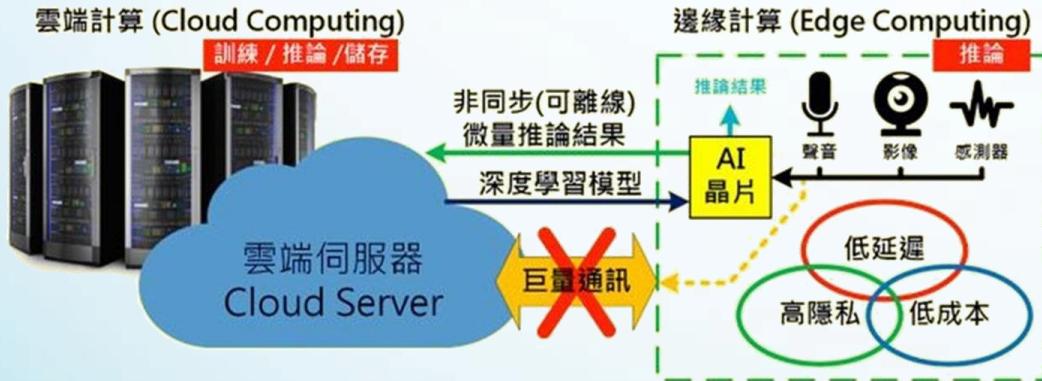


OmniXRI's Edge AI & TinyML 小學堂



【第10講】

實作案例 — 影像分割



歐尼克斯實境互動工作室 (OmniXRI Studio)
許哲豪 (Jack Hsu)

簡報大綱



- 10.1. 影像分割簡介
- 10.2. 影像分割模型
- 10.3. 影像分割評量
- 10.4. 影像分割實作

本課程完全免費，請勿移作商業用途！
歡迎留言、訂閱、點讚、轉發，讓更多需要的朋友也能一起學習。

完整課程大綱：<https://omnixri.blogspot.com/2024/02/omnixris-edge-ai-tinyml-0.html>
課程直播清單：<https://www.youtube.com/@omnixri1784streams>

影像分割（像素級分類）類型

影像分割為像素等級分類

➤ 語義分割 (Semantic)

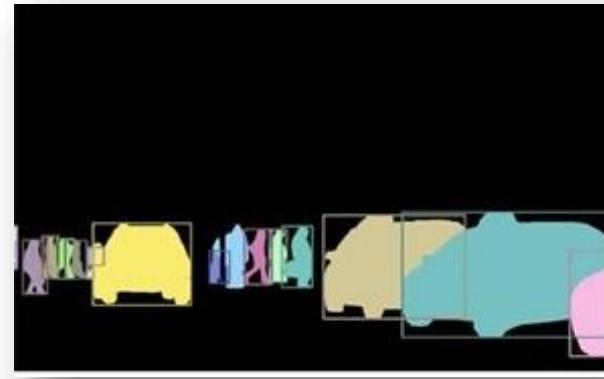
每個像素分到指定類別，無法分類的全部算成背景。



原始影像

➤ 實例分割 (Instance)

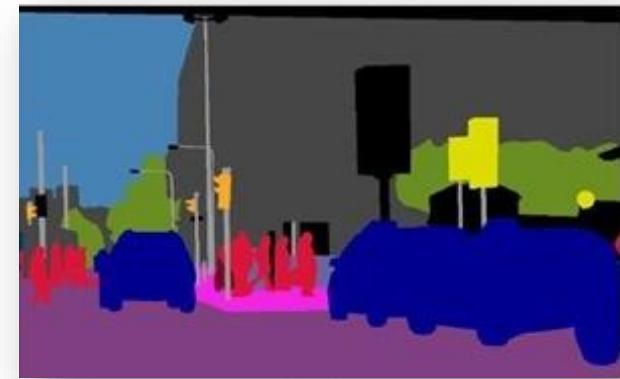
分到相同類別的像素還要更進一步區分是否為同一物件



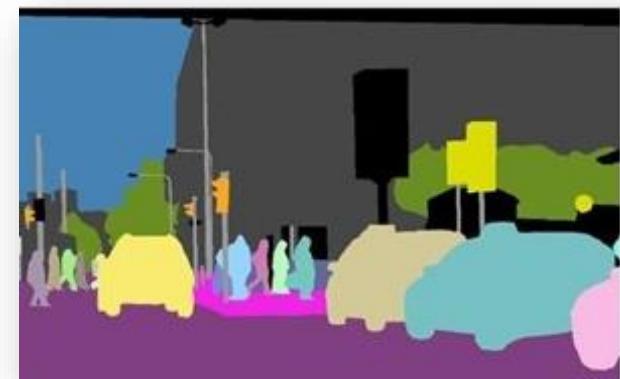
實例分割

➤ 全景分割 (Panoramic)

結合語義和實例分割，多用於自駕車上，包括街景、路面、號誌、天空等都要分割。



語義分割



全景分割

資料來源：<https://omnixri.blogspot.com/p/ntust-edge-ai-ch3-3.html>

傳統影像分割技術

尋找邊界

- Sobel, Laplacian, Canny...

特定封閉區域

- 橢圓形、旋轉矩形、多角形...

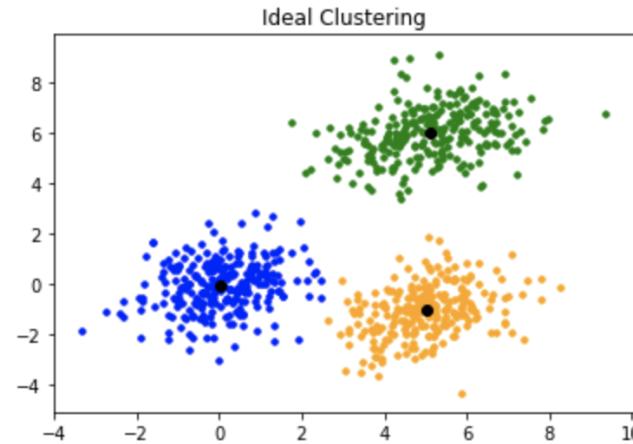
特徵區域提取

- 色彩、亮度、紋理...
- 型態學分割、合併...

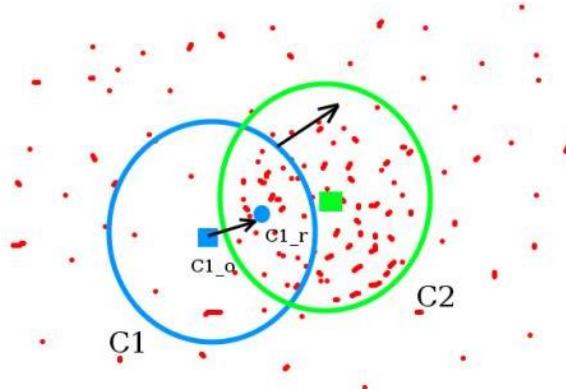
像素聚類

- Seeds Growing, K-Mean, Mean-shift, Watershed, Grabcut...

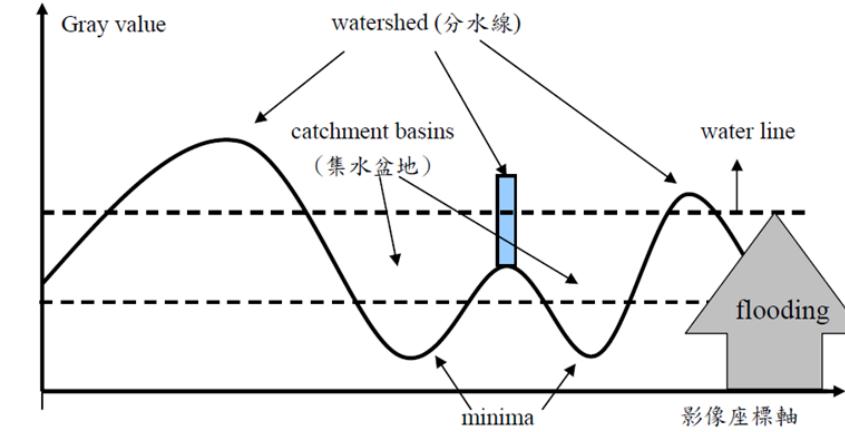
OpenCV常用影像分割算法(1/2)



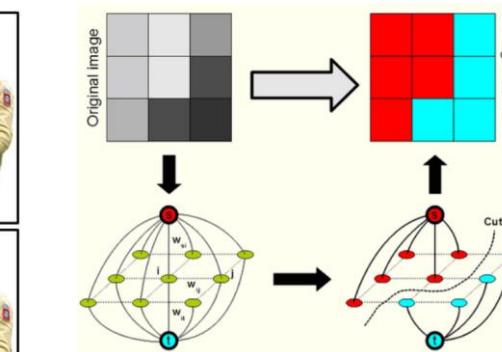
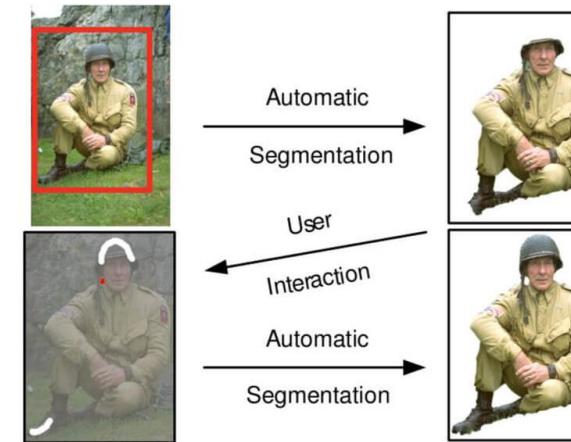
中心聚類 (K-Mean)



中心移位 (Mean-Shift)



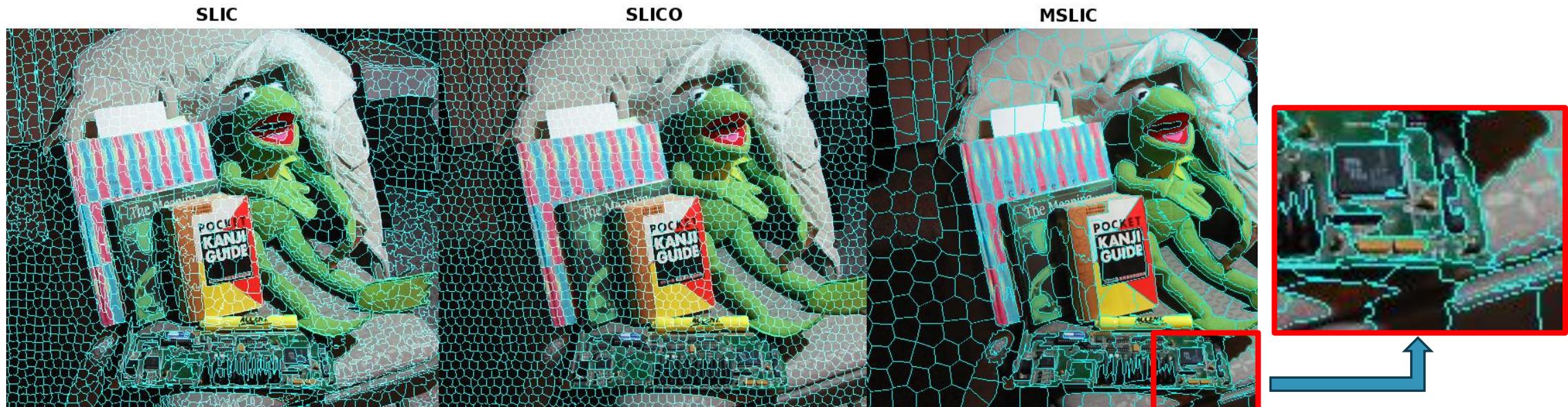
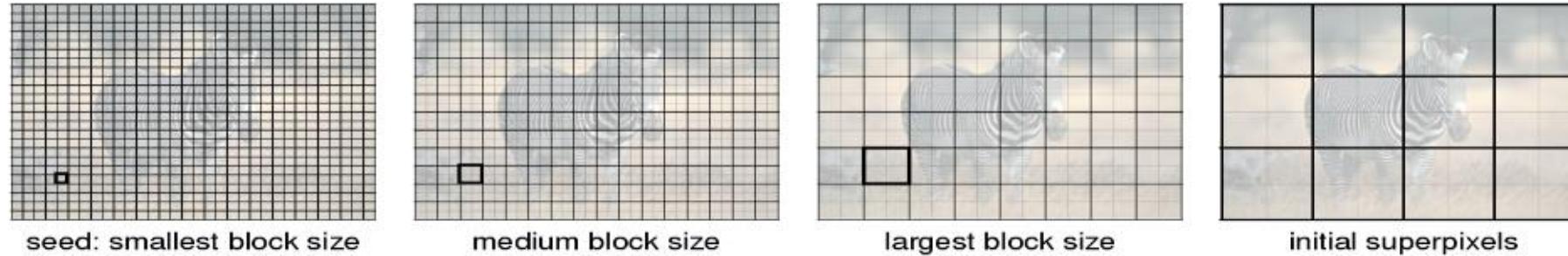
分水嶺 (Watershed)



互動式分割 (Grabcut)

OpenCV常用影像分割算法(2/2)

超像素 (Superpixel)



資料來源：https://docs.opencv.org/3.4/df/d6c/group_ximgproc_superpixel.html

常見影像分割標註方式

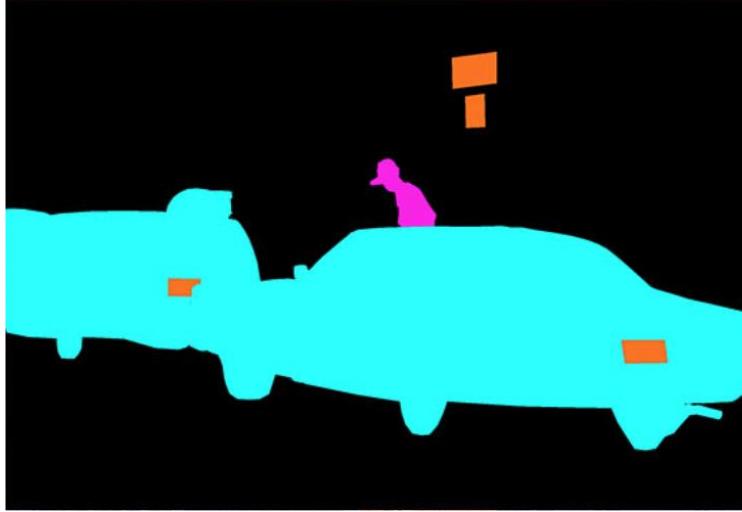
貝茲曲線
(Bezier)

不受影像
尺寸影響



封閉多邊形
(Polygon)

不精準但
標註速度快



自由筆刷
(Brush)

很不精準



超像素
(Superpixel)

OpenCV

- LSC
- SEEDS
- SLIC

無自動合併



影像分割標註工具

PixelAnnotationTool

<https://github.com/abreheret/PixelAnnotationTool>

基於OpenCV分水嶺方式實現語義分割，分割結果以圖檔輸出。

superpixels-segmentation

<https://github.com/Labelbox/superpixels-segmentation>

基於超像素SLIC方式實現影像分割，分割結果以圖檔輸出。

semantic-segmentation-editor

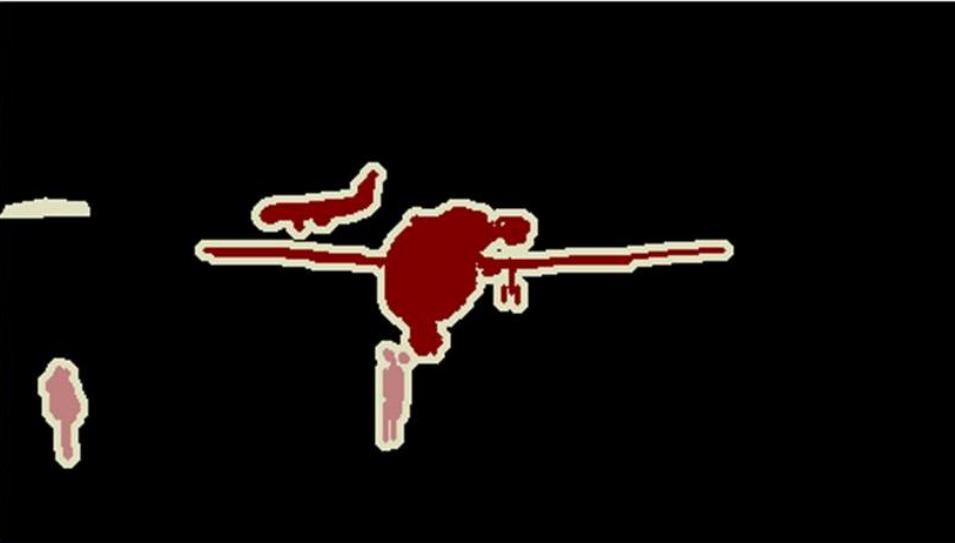
<https://github.com/Hitachi-Automotive-And-Industry-Lab/semantic-segmentation-editor>

基於WEB的標註工具，主要用於自駕車領域，支援2D(jpg, png)、3D(pcd)影像檔，具多邊形繪圖工具、魔術棒等工具，亦可對多邊形進行切割等工作。

VOC語義分割資料格式



原始影像 (*.jpg)



語義分割後影像 (*.png)

黑色為背景

白色為邊界
(非必要)

PNG標註檔
探索引式
(Index)格式
類似GIF格式

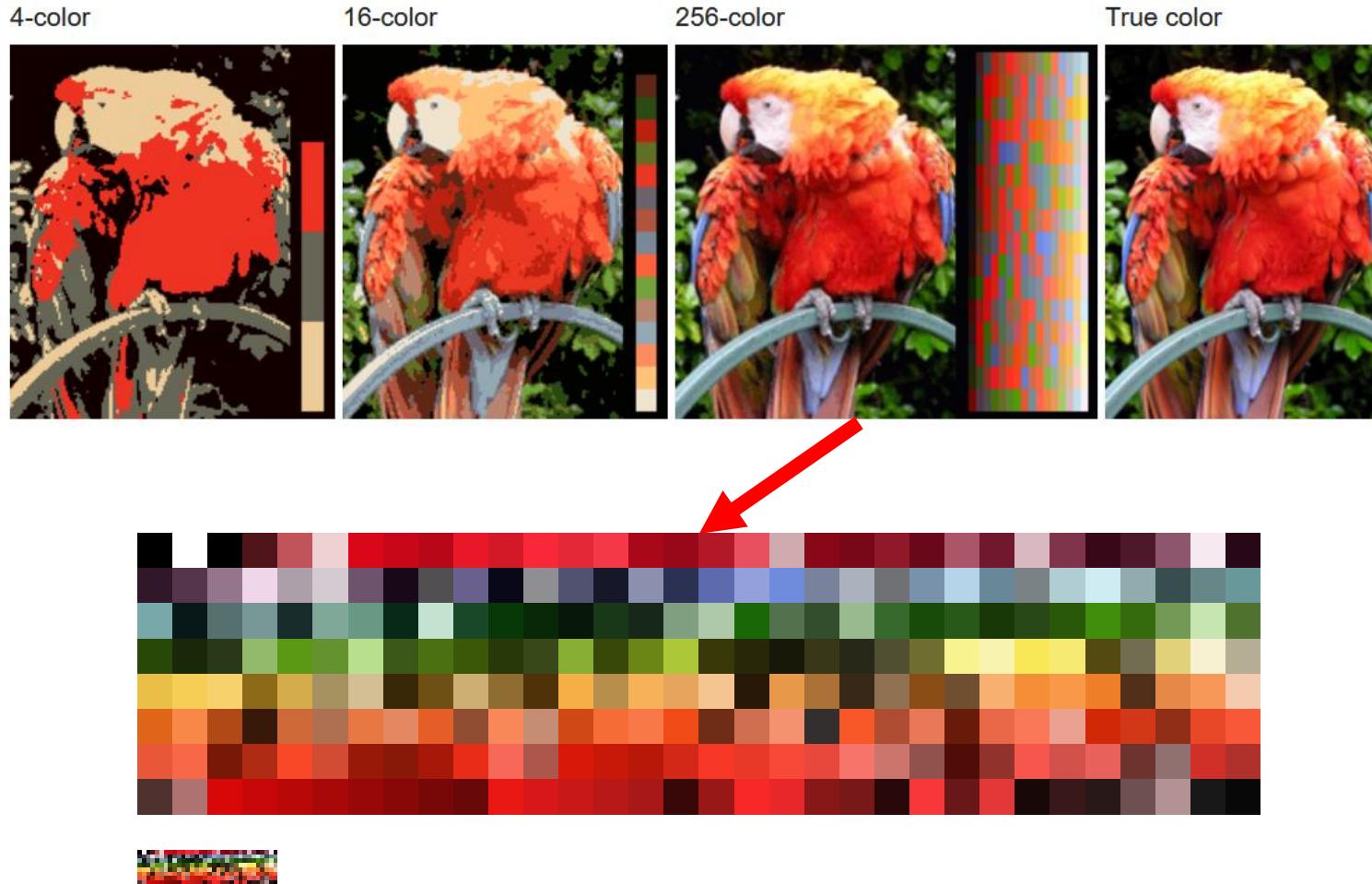
	Aeroplane		Diningtable		Bottle
	Bicycle		Cat		Bus
	Bird		Horse		Car
	Boat		Motorbike		Dog

	Person		Chair		Train
	Pottedplant		Cow		Tvmonitor
	Sheep		Sofa		

VOC 20+1類別

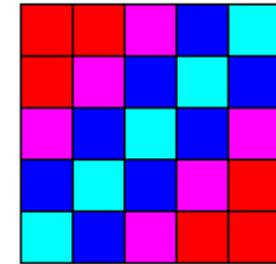
OmniXRI Oct. 2020 整理繪製

PNG8 (Indexed, Palette) 格式



0	0	1	2	3
0	1	2	3	2
1	2	3	2	1
2	3	2	1	0
3	2	1	0	0

0 = 
 1 = 
 2 = 
 3 = 
indexed



PNG8 使用 8bit ($2^8 = 256$ 色) 來表示索引色，須搭配調色盤使用。

資料來源：<https://zh.wikipedia.org/zh-tw/PNG> , https://en.wikipedia.org/wiki/Indexed_color

COCO語義分割資料格式

COCO 資料格式

```
{
  "info": info,
  "images": [image],
  "annotations": [annotation],
  "licenses": [license],
}

info{
  "year": int,
  "version": str,
  "description": str,
  "contributor": str,
  "url": str,
  "date_created": datetime,
}

image{
  "id": int,
  "width": int,
  "height": int,
  "file_name": str,
  "license": int,
  "flickr_url": str,
  "coco_url": str,
  "date_captured": datetime,
}

license{
  "id": int,
  "name": str,
  "url": str,
}
}
```

影像分類/物件偵測/語義分割標註資料格式

```
annotation{
  "id": int,
  "image_id": int,
  "category_id": int,
  "segmentation": RLE or [polygon],
  "area": float,
  "iscrowd": 0 or 1,
  "bbox": [x,y,width,height],
}

categories:[
  {"id": int,
  "name": str,
  "supercategory": str,
}]
```

Segmentation : [polygon]

```
"segmentation": [[510.66,423.01,...,510.45,423.01]],
"area": 702.1057499999998,
"iscrowd": 0,
```

Segmentation : [RLE]

```
"segmentation": {"counts": [20736,2,453,5,452,9,447,13,444,...,5,34552],
"size": [457,640]},
"area": 3074,
"iscrowd": 1,
```

OmniXRI Oct. 2020 整理繪製

Segmentation : [polygon] 封閉多邊形

"segmentation": X1, Y1, X2, Y2, ... , Xn, Yn
[[510.66,423.01,...,510.45,423.01]],
"area": 702.1057499999998,
"iscrowd": 0,

Segmentation : [RLE] 二進制影像壓縮

"segmentation": {"counts": [20736,2,453,5,452,9,447,13,444,...,5,34552],
"size": [457,640]},
"area": 3074,
"iscrowd": 1,

黑像素個數, 白像素個數, 黑, 白, ...

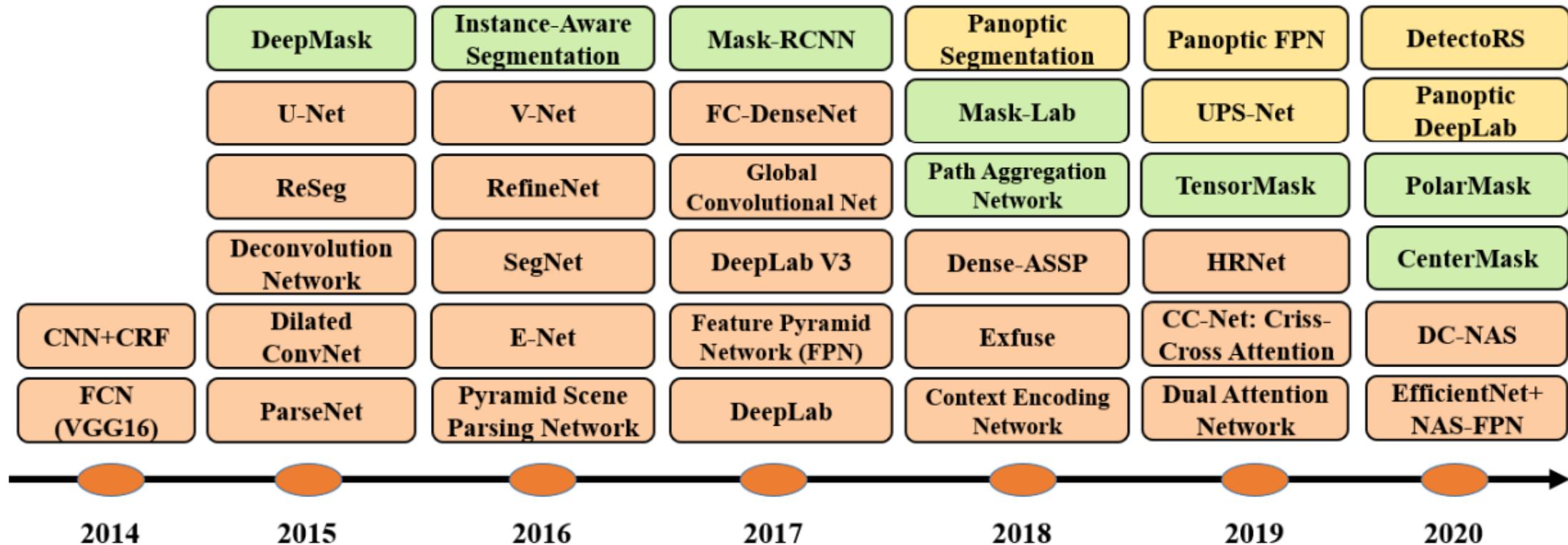


深度學習影像分割技術演進

橙色：語義分割

綠色：實例分割

黃色：全景分割



參考資料：<https://arxiv.org/abs/2001.05566>

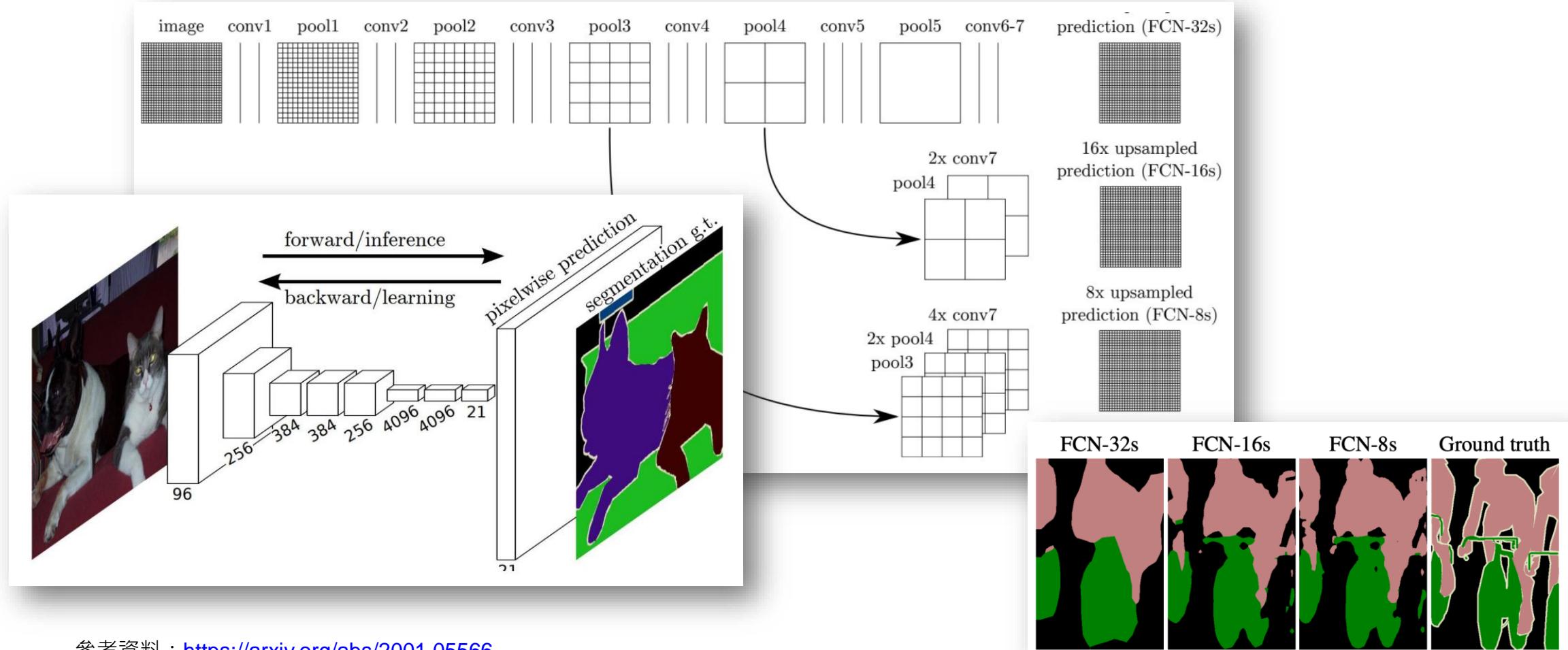
常見語義分割技術

1. Fully convolutional networks
2. Convolutional models with graphical models
3. Encoder-decoder based models
4. Multi-scale and pyramid network based models
5. R-CNN based models (for instance segmentation)
6. Dilated convolutional models and DeepLab family
7. Recurrent neural network based models
8. Attention-based models
9. Generative models and adversarial training
10. Convolutional models with active contour models
11. Other models

參考資料：<https://arxiv.org/abs/2001.05566>

常見語義分割模型 FCN

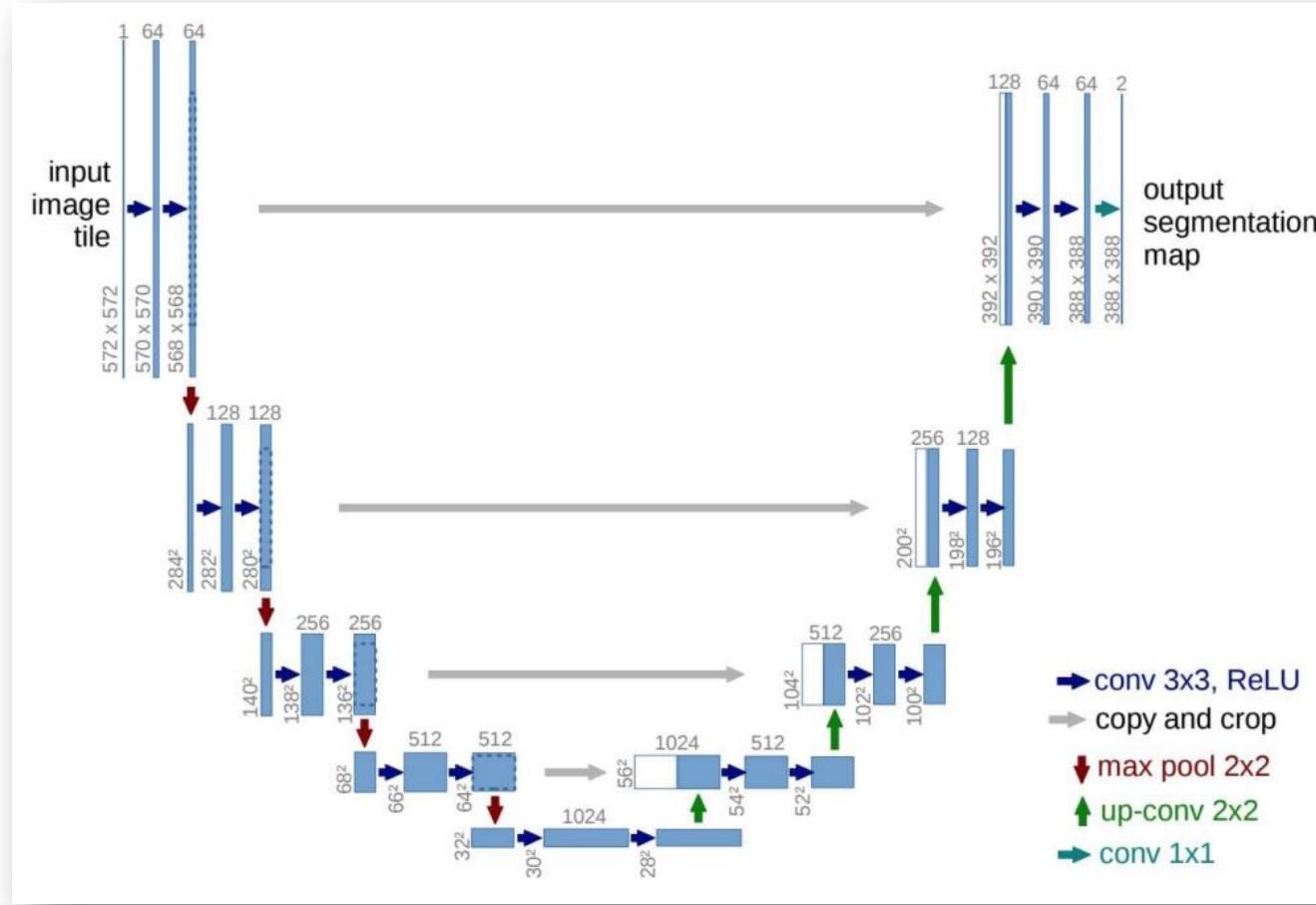
Fully Convolutional Networks Based Model



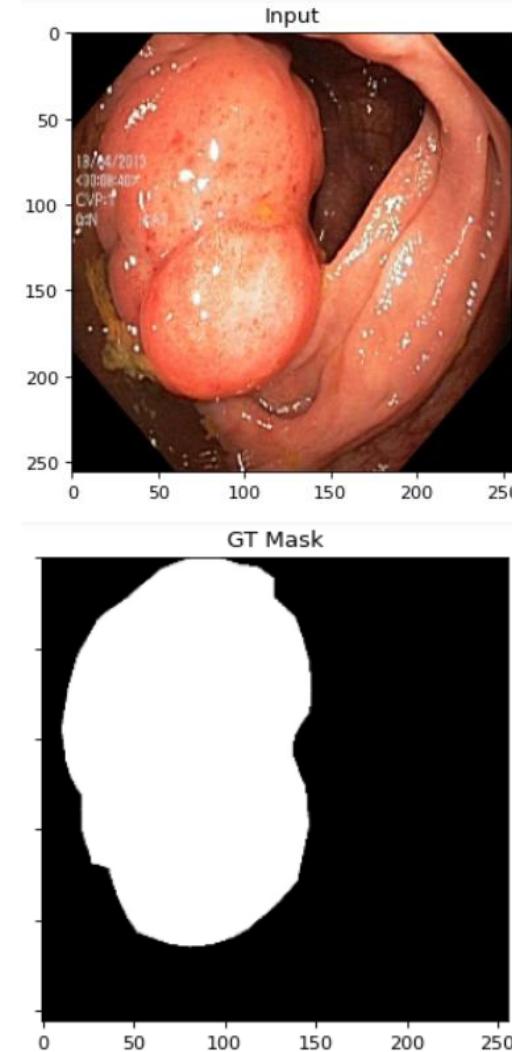
參考資料：<https://arxiv.org/abs/2001.05566>

常見語義分割模型 U-Net

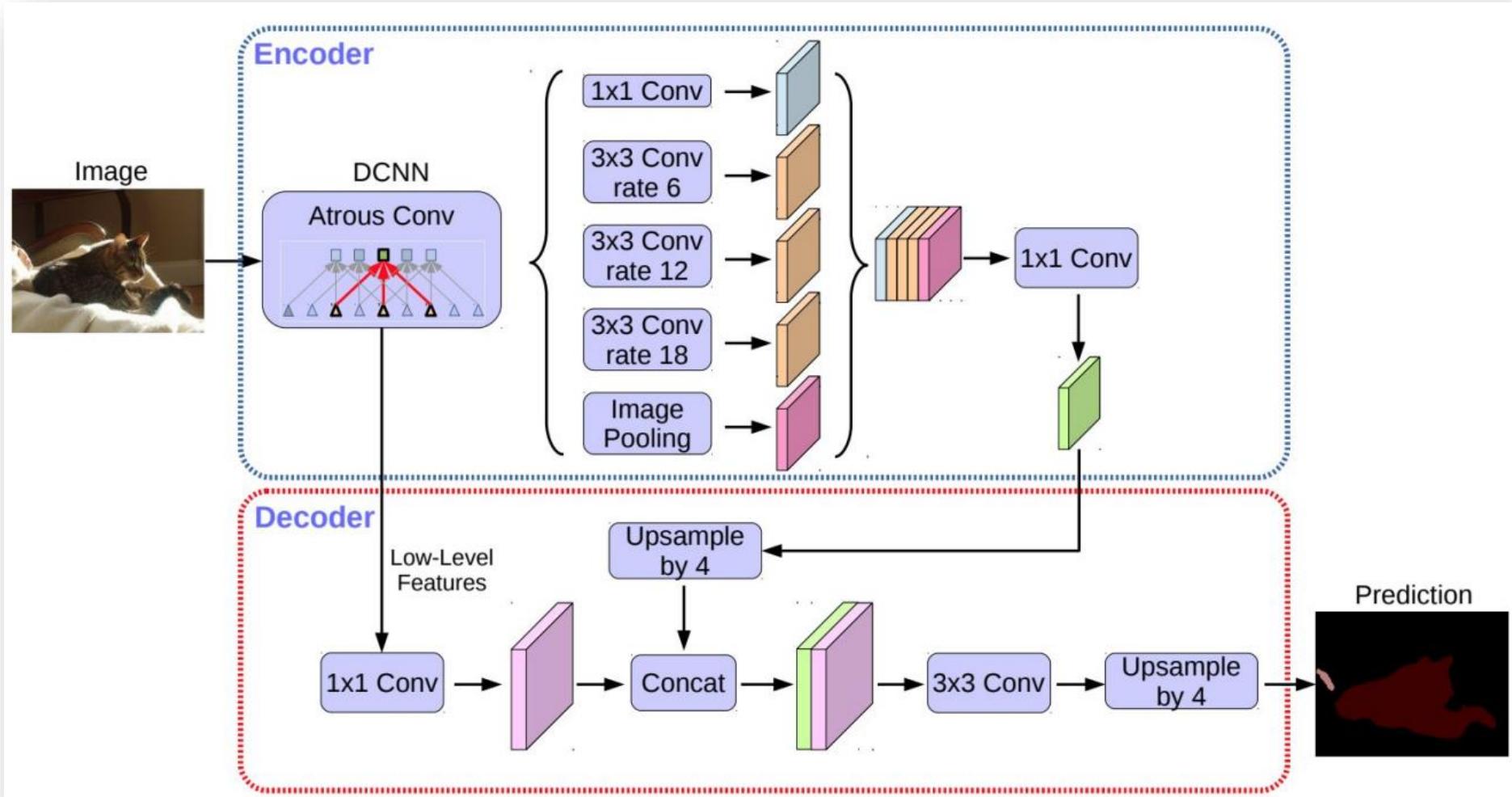
Encoder-Decoder Based Models



參考資料：<https://arxiv.org/abs/2001.05566>



常見語義分割模型 DeepLabV3+

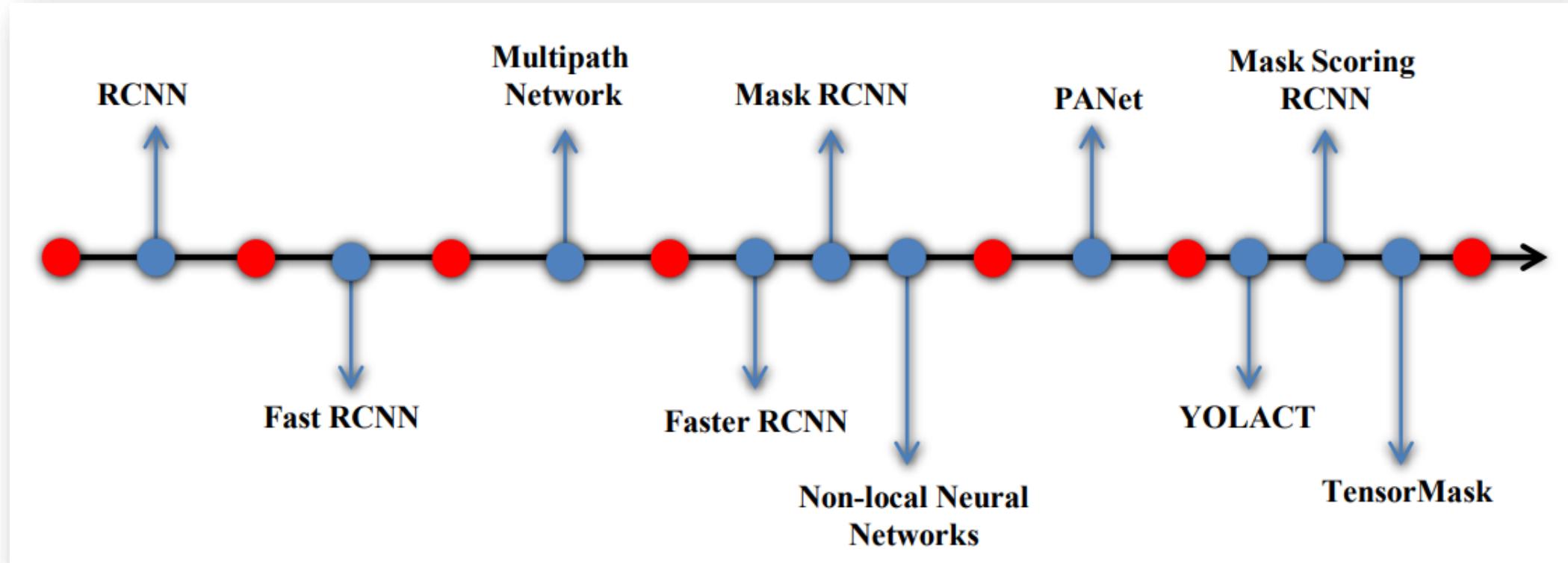


參考資料：<https://arxiv.org/abs/2001.05566>

常見實例分割技術

1. Classification of mask proposals
2. Detection followed by segmentation
3. Labelling pixels followed by clustering
4. Dense sliding window methods

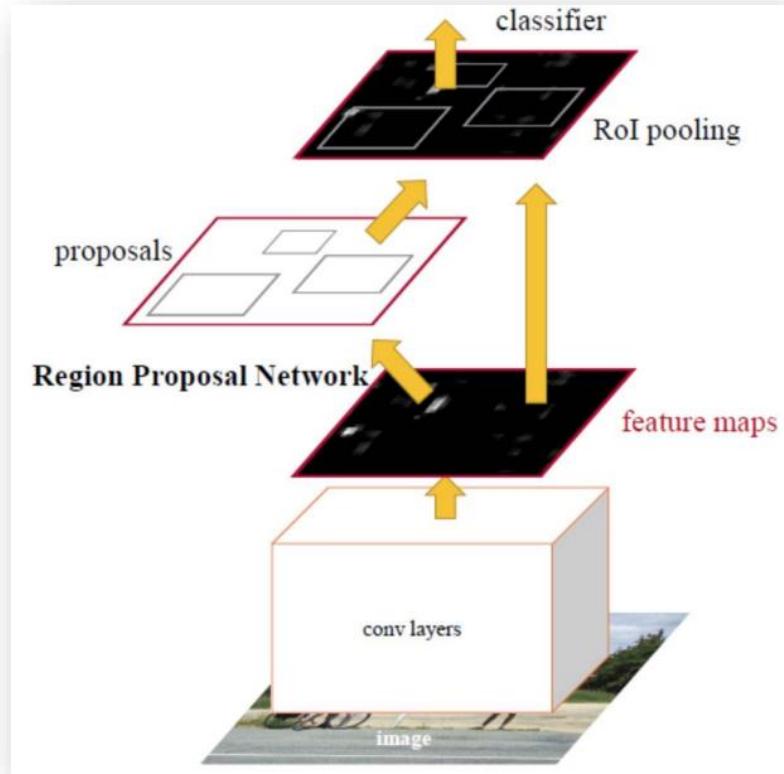
實例分割技術演進



資料來源：<https://arxiv.org/abs/2007.00047>

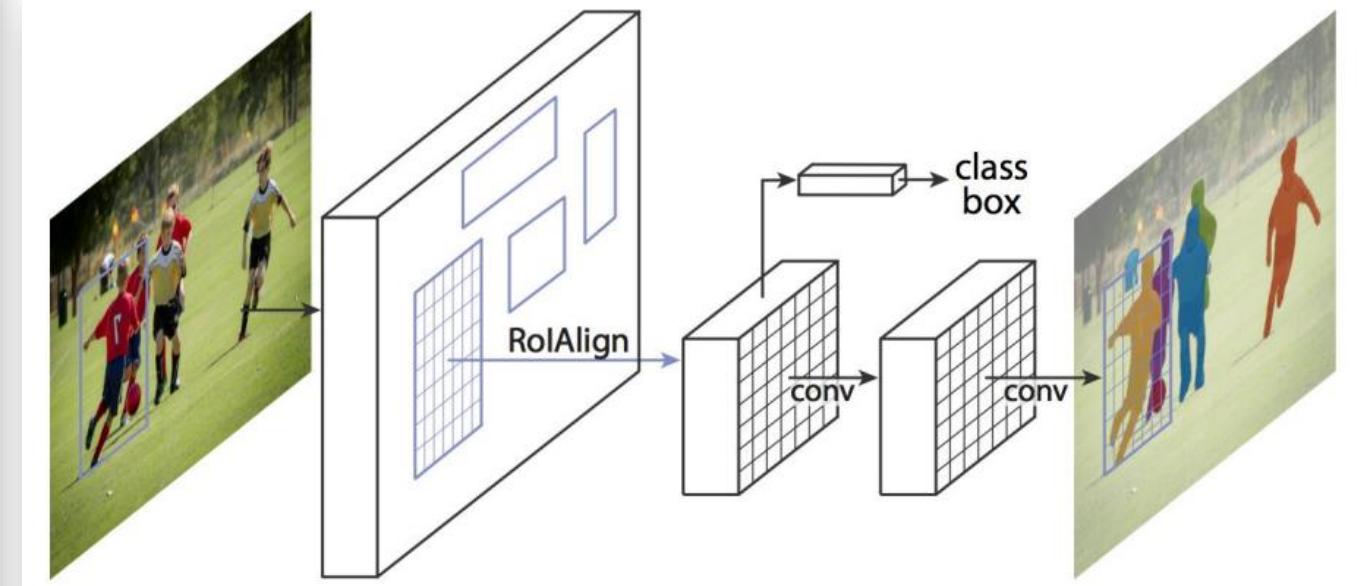
常見實例分割模型 R-CNN

Region Convolution Neural Network Based Model



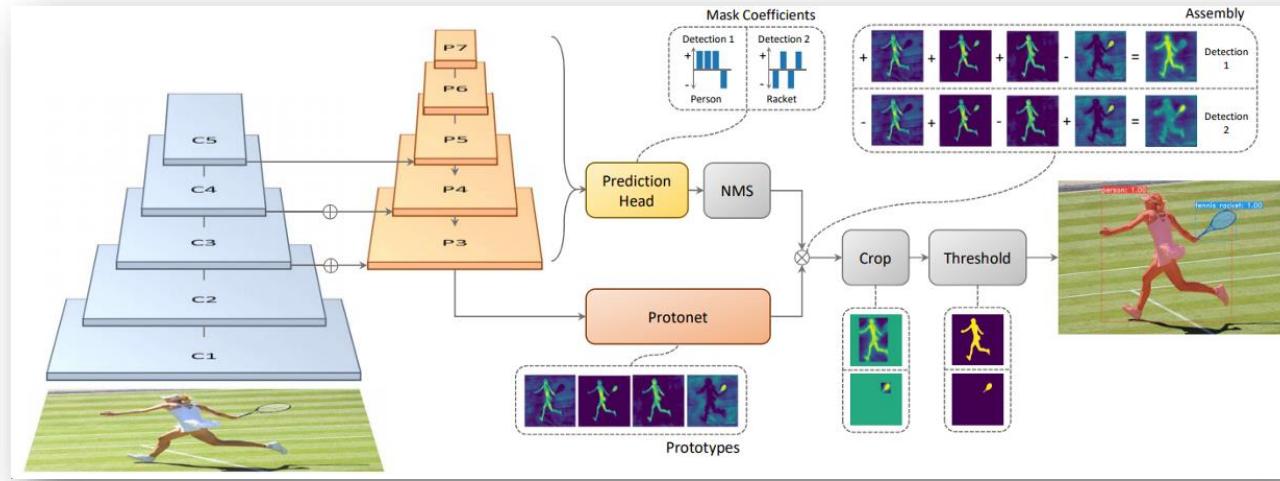
Faster R-CNN

參考資料：<https://arxiv.org/abs/2001.05566>

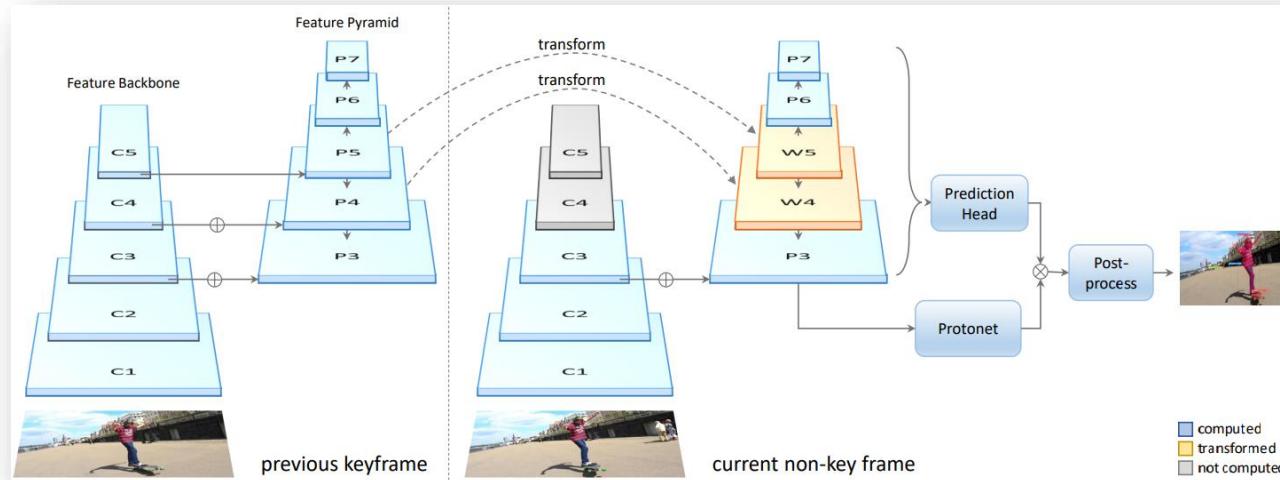


Mask R-CNN

常見實例分割模型 Yolact & Edge



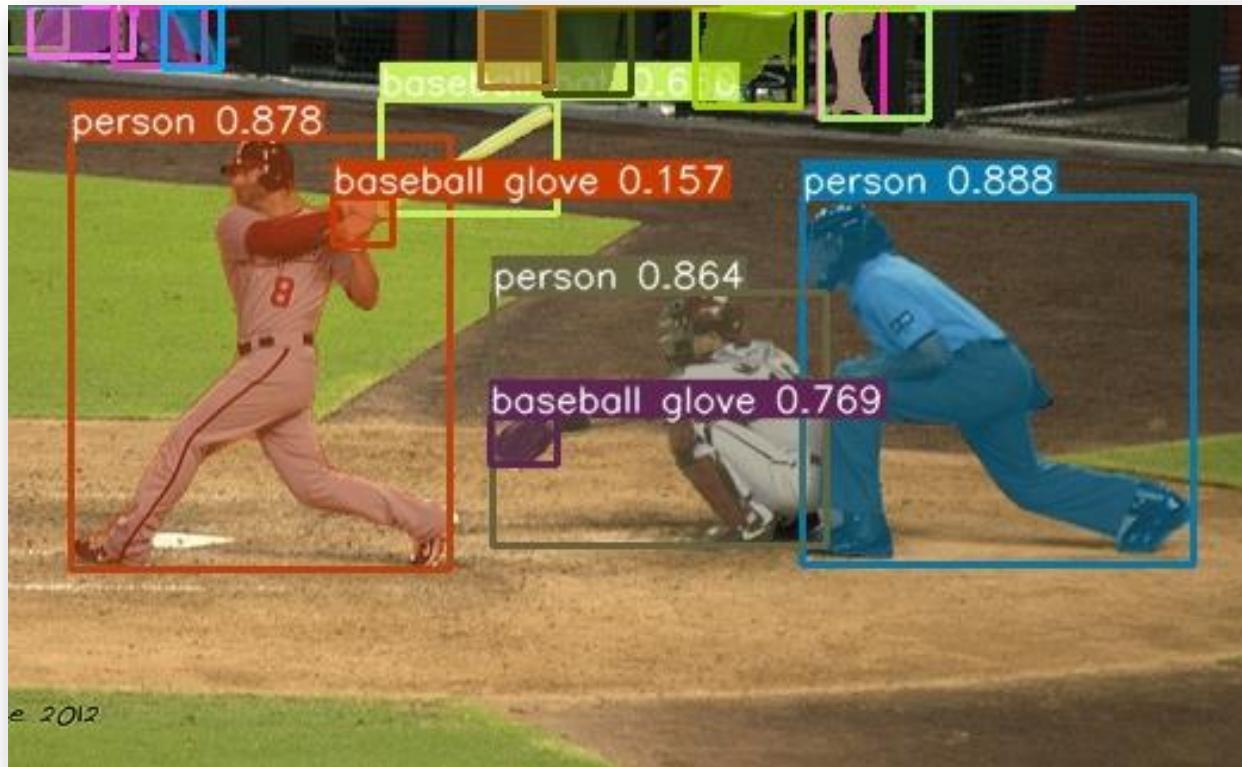
You Only
Look At
CoefficienTs



YOLACT
Edge

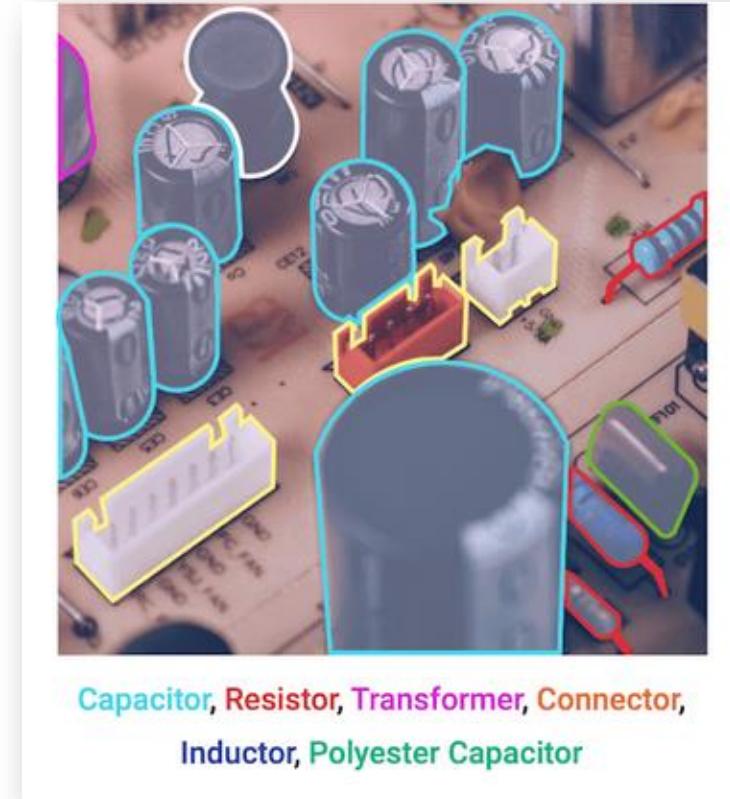
常見實例分割 YOLO 系列

Academia Sinica YOLOv7



<https://github.com/WongKinYiu/yolov7>

Ultralytics YOLOv8



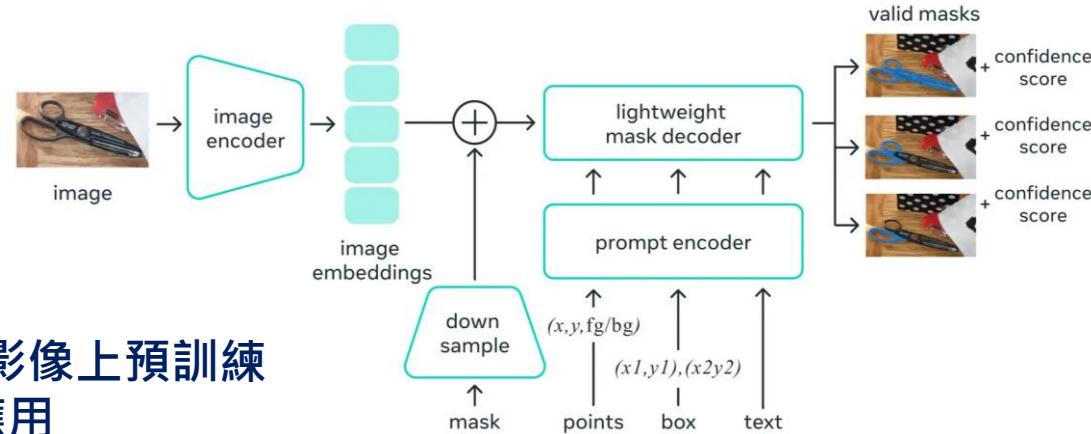
<https://docs.ultralytics.com/tasks/segment/>

提示式分割 Meat SAM

Meta - Segment Anything Model

- ViT影像編碼器
- 提示編碼器
- 輕量遮罩解碼器

- 1.1億個遮罩在1100萬個影像上預訓練
- Data-Centric AI的新式應用



點擊互動式分割



自動分割物件

<https://segment-anything.com/>



不明確提示生成遮罩

評量指標 — 像素級



像素準確率
(Pixel Accuracy, PA)

$$PA = \frac{\sum_{i=0}^k p_{ii}}{\sum_{i=0}^k \sum_{j=0}^k p_{ij}}$$

平均像素準確率
(Mean Pixel Accuracy, MPA)

$$MPA = \frac{1}{k+1} \sum_{i=0}^k \frac{p_{ii}}{\sum_{j=0}^k p_{ij}}$$

平均交集聯集比
(Mean Intersection over Union, MIoU)

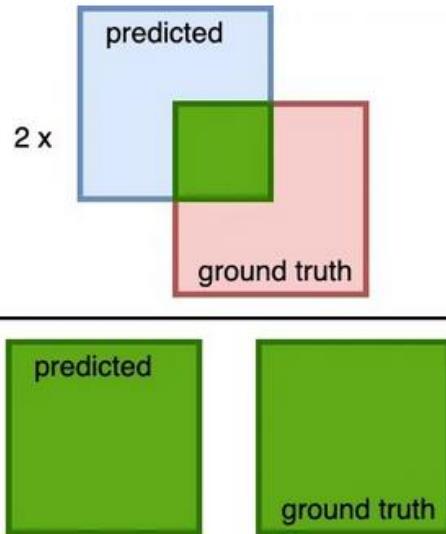
$$MIoU = \frac{1}{k+1} \sum_{i=0}^k \frac{p_{ii}}{\sum_{j=0}^k p_{ij} + \sum_{j=0}^k p_{ji} - p_{ii}}$$

圖片來源：<http://omnixri.blogspot.com/2019/07/aigoai.html>

評量指標 – DICE Coefficient

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

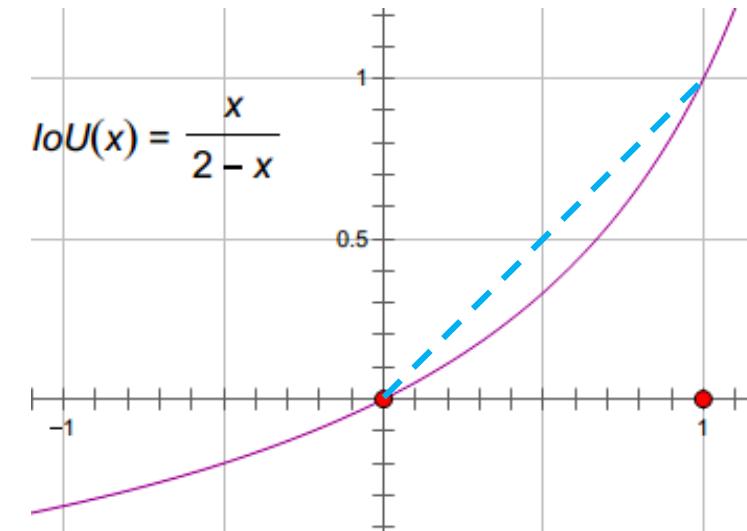
Dice coefficient = $\frac{\text{area of overlapped (green)}}{\text{total area (green)}} = \frac{2x}{\text{predicted} + \text{ground truth}}$



$$Dice = \frac{2 \times TP}{(TP + FP) + (TP + FN)}$$

predicted **Ground truth**

真陽 : True Positive 真陰 : True Negative
 偽陽 : False Positive 偽陰 : False Negative



OpenVINO Open Model Zoo (Public)

Semantic Segmentation Models

Model Name	Implementation	OMZ Model Name	Accuracy	GFlops	mParams
Model Name	Implementation	OMZ Model Name	Accuracy	GFlops	mParams
DeepLab V3	TensorFlow*	deeplabv3	68.41%	11.469	23.819
DRN-D-38	PyTorch*	drn-d-38	71.31%	1768.3276	25.9939
Erfnet	PyTorch*	erfnet	76.47%	11.13	7.87
HRNet V2 C1 Segmentation	PyTorch*	hrnet-v2-c1-segmentation	77.69%	81.993	66.4768
Fastseg MobileV3Large LR-ASPP, F=128	PyTorch*	fastseg-large	72.67%	140.9611	3.2
Fastseg MobileV3Small LR-ASPP, F=128	PyTorch*	fastseg-small	67.15%	69.2204	1.1
PSPNet R-50-D8	PyTorch*	pspnet-pytorch	70.6%	357.1719	46.5827

3D Semantic Segmentation Models

Model Name	Implementation	OMZ Model Name	Accuracy	GFlops	mParams
Model Name	Implementation	OMZ Model Name	Accuracy	GFlops	mParams
Brain Tumor Segmentation 2	PyTorch*	brain-tumor-segmentation-0002	91.4826%	300.801	4.51

Instance Segmentation Models

Instance segmentation is an extension of object detection and semantic segmentation problems. Instead of predicting a bounding box around each object instance instance segmentation model outputs pixel-wise masks for all instances.

Model Name	Implementation	OMZ Model Name	Accuracy	GFlops	mPara
Model Name	Implementation	OMZ Model Name	Accuracy	GFlops	mPara
Mask R-CNN Inception ResNet V2	TensorFlow*	mask_rcnn_inception_resnet_v2_atrous_coco	39.86%/35.36%	675.314	92.368
Mask R-CNN ResNet 50	TensorFlow*	mask_rcnn_resnet50_atrous_coco	29.75%/27.46%	294.738	50.222
YOLACT ResNet 50 FPN	PyTorch*	yolact-resnet50-fpn-pytorch	28.0%/30.69%	118.575	36.829

https://docs.openvino.ai/2024/omz_models_group_public.html#segmentation-models

OpenVINO Notebooks Image Segmentation

Filter AI Tasks by name

Multimodal

- [Text-to-Image](#)
- [Image-to-Text](#)
- [Text-to-Video](#)
- [Video-to-Text](#)
- [Text-to-Audio](#)
- [Audio-to-Text](#)
- [Visual Question Answering](#)
- [Image Captioning](#)
- [Feature Extraction](#)
- [Text-to-Image Retrieval](#)
- [Image-to-Text Retrieval](#)
- [Image-to-3D](#)
- [Image-to-Video](#)

Computer Vision

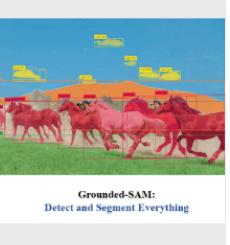
- [Image Classification](#)
- [Image Segmentation](#)
- [Image Inpainting](#)



Background removal with RMBG v1.4 and OpenVINO

Model Demos • Image Segmentation

[View on GitHub](#) [Open in Colab](#) [Launch in Binder](#)



Object detection and masking from prompts with GroundedSAM (GroundingDINO + SAM) and OpenVINO

Convert • Optimize • Model Demos • Live Demos • Object Detection • Image Segmentation

[View on GitHub](#) [Open in Colab](#)



Object segmentations with EfficientSAM and OpenVINO

Model Demos • Image Segmentation

[View on GitHub](#)



Object segmentations with FastSAM and OpenVINO

Model Demos • Image Segmentation

https://openvinotoolkit.github.io/openvino_notebooks/

實作範例1 — 路況實例分割 (colab)

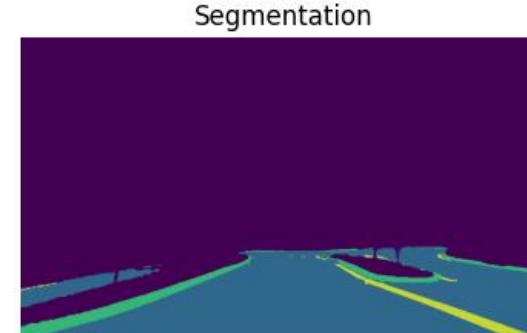
hello-segmentation

(原003-hello-segmentation)

使用 [road-segmentation-adas-0001](#) 模型，主要將影像像素分為四類，背景、道路、路緣及路標。

程式導讀

- Imports
- Download model weights
- Select inference device
- Load the Model
- Load an Image
- Do Inference
- Prepare Data for Visualization
- Visualize data



https://colab.research.google.com/github/openvinotoolkit/openvino_notebooks/blob/latest/notebooks/hello-segmentation/hello-segmentation.ipynb

實作範例2 – YOLOv8 實例分割 (Colab)

Yolov8-optimization/yolov8-instance-segmentation (原230-yolov8-instance-segmentation) 程式導讀



程式導讀

- Prepare the PyTorch model.
- Download and prepare a dataset.
- Validate the original model.
- Convert the PyTorch model to OpenVINO IR.
- Validate the converted model.
- Prepare and run optimization pipeline.
- Compare performance of the FP32 and quantized models.
- Compare accuracy of the FP32 and quantized models.
- Live demo

https://colab.research.google.com/github/openvinotoolkit/openvino_notebooks/blob/latest/notebooks/yolov8-optimization/yolov8-instance-segmentation.ipynb

實作範例3—物件分割 FastSAM (Colab)

Object segmentations with FastSAM and OpenVINO (fast-segment-anything)



Point-prompt



Box-prompt

程式導讀：

Prerequisites

Install requirements

Imports

FastSAM in Ultralytics

Convert the model to OpenVINO Intermediate representation (IR) format

Embedding the converted models into the original pipeline

Select inference device

Adapt OpenVINO models to the original pipeline

Optimize the model using NNCF Post-training Quantization API

Compare the performance of the Original and Quantized Models

Try out the converted pipeline



https://colab.research.google.com/github/openvinotoolkit/openvino_notebooks/blob/latest/notebooks/fast-segment-anything/fast-segment-anything.ipynb

參考文獻

- 許哲豪，臺灣科技大學資訊工程系「人工智慧與邊緣運算實務」
(2021~2023)

<https://omnixri.blogspot.com/p/ntust-edge-ai.html>

- 許哲豪，【AI HUB專欄】如何建立精準標註的電腦視覺資料集

https://omnixri.blogspot.com/2020/10/ai-hub_16.html

- Shervin Minaee etc., Image Segmentation Using Deep Learning: A Survey

<https://arxiv.org/abs/2001.05566>

- Abdul Mueed Hafiz etc., A Survey on Instance Segmentation: State of the art

<https://arxiv.org/abs/2007.00047>

延伸閱讀

➤ 武卓，AI分割一切：用OpenVINO加速Meta SAM大模型

<https://makerpro.cc/2023/05/openvino-accelerated-meta-segment-anything-model/>

➤ Sunidhi Ashtekar, Dynamic Object Detection and Segmentation with YOLOv9+SAM

<https://medium.com/@sunidhi.ashtekar/dynamic-object-detection-and-segmentation-with-yolov9-sam-de258238546f>

沒有最邊



只有更邊



歡迎加入
邊緣人俱樂部



YOUTUBE 直播 : <https://www.youtube.com/@omnixri1784streams>



歐尼克斯實境互動

歐尼克斯實境互動工作室
(OmniXRI Studio)

許哲豪 (Jack Hsu)

[Facebook : Jack Omnidri](#)

[FB社團 : Edge AI Taiwan 邊緣智能交流區](#)

[電子郵件 : omnixri@gmail.com](#)

[部落格 : https://omnidri.blogspot.tw](#)

[開 源 : https://github.com/OmniXRI](#)