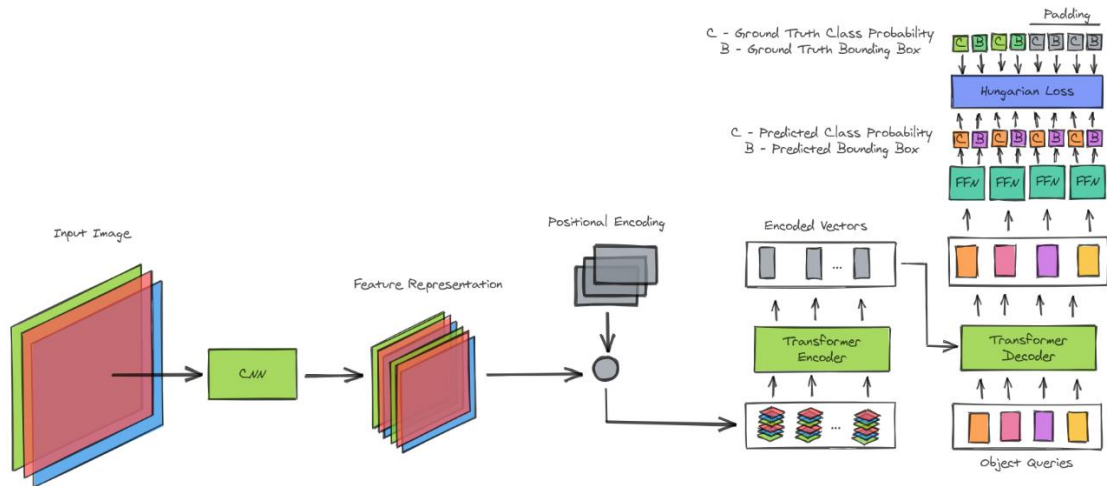
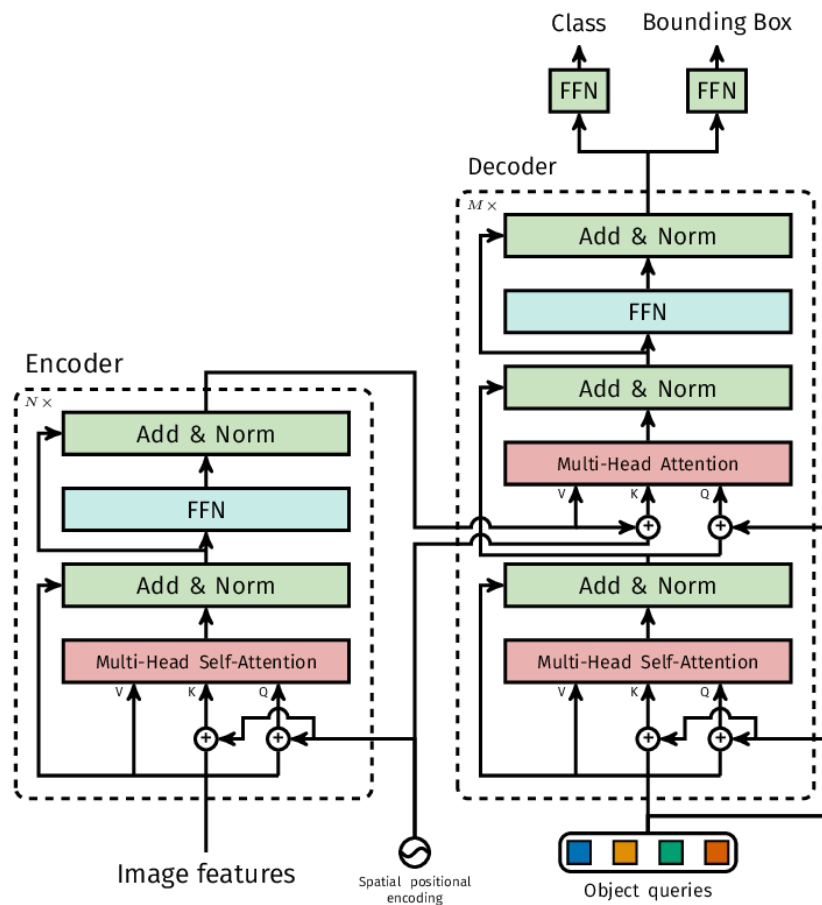


# 1. Draw the architecture (DETR)

Input Image  $\rightarrow$  Backbone (ResNet-50)  $\rightarrow$  Image Features  $\rightarrow$  Transformer Encoder  $\rightarrow$  Transformer Decoder  $\rightarrow$  Detection Heads (Bounding Boxes & Labels)



Refer: <https://github.com/anasch07/DETR-Object-Detection?tab=readme-ov-file>



⤴ Transformer (DETR)

## 2. Implement details

e.g.: augmentation, loss function, parameter settings

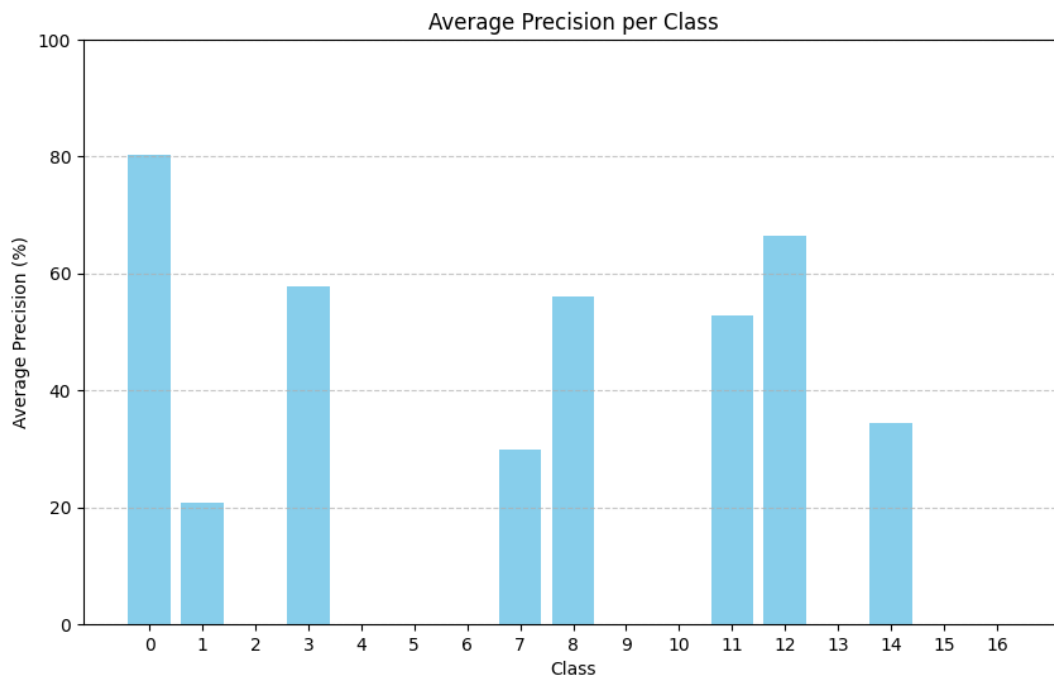
- Data Augmentation(預處理階段):
  - Resize: 將圖像調整為 1024x1024 的固定大小，確保模型接受一致的輸入尺寸。
  - Horizontal Flip: 以 100% 的概率隨機進行水平翻轉，增加模型的 robust。
  - Random Brightness and Contrast: 隨機調整圖像的亮度和對比度，以增強模型對不同光線條件的適應能力。
- Loss Function
  - Classification Loss: 使用交叉熵損失來計算物體類別的預測誤差。
  - Bounding Box Loss: 使用 L1 損失來衡量預測邊界框與真實邊界框的差異。
- Parameter Settings (for training)
  - Learning Rate: 設置為  $1e-5$ 。
  - Batch Size: 設置為 8，在 GPU 訓練中保持穩定的性能。
  - Number of Epochs: 總訓練輪數為 100，確保模型有足夠的訓練時間以學習特徵。
  - Weight Decay: 設置為  $1e-3$ ，用於防止 overfit。
  - Gradient Accumulation: 設置為 2，允許在無法增加 batch 大小的情況下累積梯度。
  - metric\_for\_best\_model, greater\_is\_better: 根據 loss 越小做為較好的模型
- Callbacks
  - Early Stopping: 設 patience 為 5，當驗證集的 loss 不再改善時提前終止訓練。
  - Model Checkpointing: 保存最佳兩模型的權重，以便後續評估。
- Specific Architecture Details
  - Backbone: 使用 ResNet-50 作為特徵提取器，提取圖像的底層特徵。
  - Transformer: 包含多層 Encoder 和 Decoder，使用 multi-head self-attention 捕捉全局上下文資訊。
- Parameter Settings (for inference)
  - Threshold(後處理): 0.9, 模型卻認為框的分數要達到 0.9 才會被列入可用的框,減少錯誤框的機會。
  - iou\_threshold: 0.7, 要求預測框與實際框之間的重疊度達到 70% 及以上,才算有效檢測。
- Experimental Setup
  - NVIDIA RTX 4070 GPU 進行訓練

### 3. Table of your performance for validation set (mAP, AP50, AP75)

mAP	0.4653
AP50	0.6898
AP75	0.4935

### 4. Visualization and discussion

Demonstrate the detection results, discussion for the long tail effect, etc.



部分類別表現優異：有幾個類別的平均精準度接近或超過 80%，如 class 0(person)顯示模型對這些類別的識別能力很強。

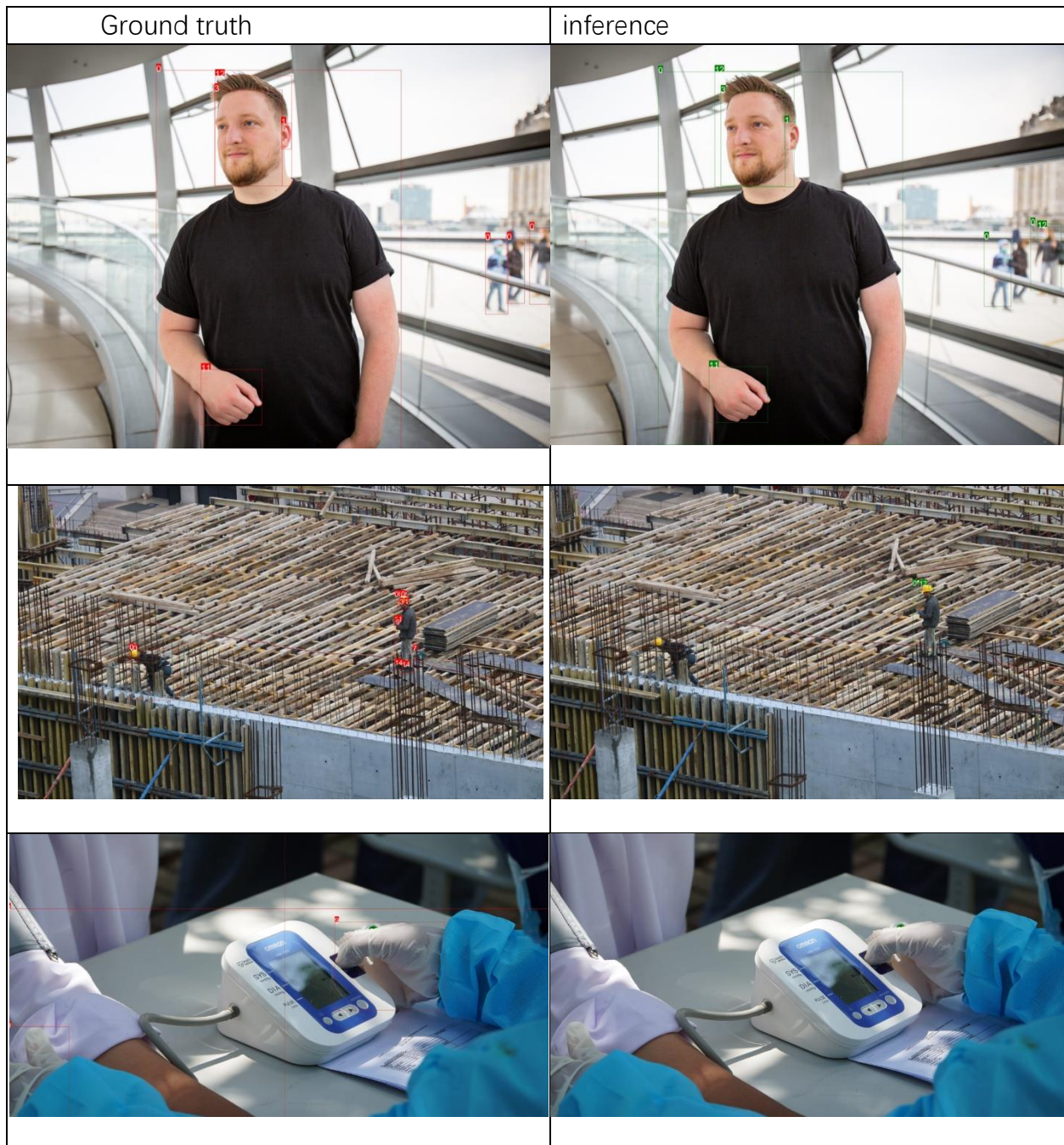
部分類別表現較差：有幾個類別的平均精準度較低，甚至接近或等於 0%，如 class2(Earmuffs) class4(Face-guard) class5 (Face-mask-medical) class6(Foot)等表示模型對這些類別的識別能力較弱。

此模型存在明顯的長尾現象：少數類別的樣本數量較多，模型對這些類別的學習效果較好；而大多數類別的樣本數量較少，模型對這些類別的學習效果較差。

可以改善的方法：

- 數據增強：對尾部類別的數據進行過採樣或數據增強，增加模型對這些類別的訓練樣本數量。

- 類別平衡：在訓練過程中，對不同類別的樣本賦予不同的權重，以平衡不同類別對模型訓練的影響。
- One-Shot Learning 或 Few-Shot Learning：對於極端不平衡的數據集，可以考慮使用 One-Shot Learning 或 Few-Shot Learning 技術，從少量樣本中學習。



可以得知從模型推測出來的框數量較少,這是因為在後處理時的 threshold 設較高的關係,從避免掉分數不高的框(判斷錯誤類別的框)以增加 mAP

Threshold in post_process_object_detection	Threshold in IoU	mAP
0.4	0.5	0.4012
0.5	0.5	0.4154
0.6	0.6	0.4348
0.6	0.7	0.4434
0.7	0.6	0.4474
0.7	0.7	0.4519
0.9	0.7	0.4653