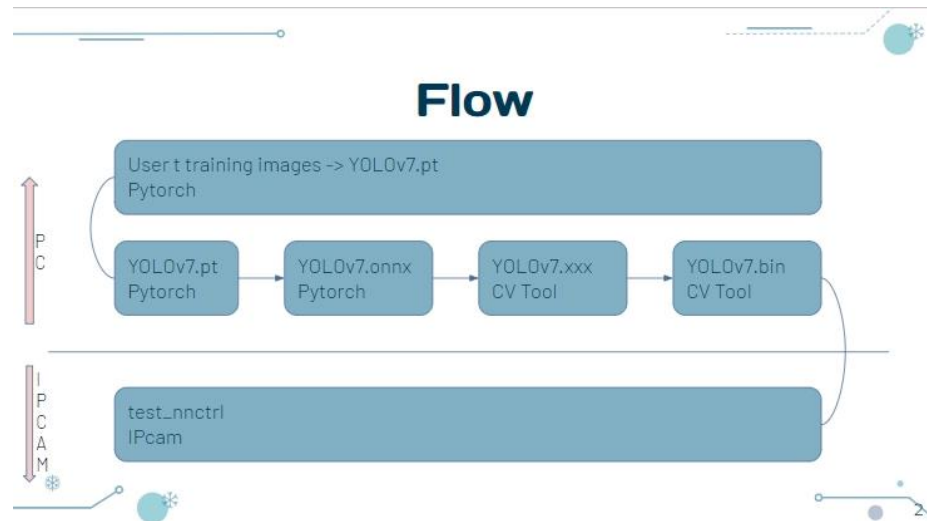


## 轉換流程圖



- ✓ 建立 VM ONNX (amba-cv-tools-basic.2.5.6.0.1245.ubuntu-20.04 SDK)
- ✓ 建立 VM SDKKK (cv2x\_sdk30\_patch9 SDK)
- ✓ 整體流程參考文件：
  - VM SDK 文件：(In VM ONNX) /home/lilin/onnx\_code.txt
  - lilin 文件：AmbaToOnnx\_環境及使用方式文件
  - amba 文件: Ambarella\_CVflow\_CNNGen\_Basic\_Introduction\_v0p2\_20230213

## Step 1: YOLO PT weight -> ONNX

- ✓ [yolov7 官網](#)提供 export.py 可以直接轉譯，但有錯誤，修改後的編輯檔如 yolov7tpt\_2\_onnx.ipynb，可以直接上傳到 colab 執行
- ✓ yolov7tpt\_2\_onnx.ipynb 主要功能：
  - 驗證 yolov7\_tiny.pt 推論
  - 產生 yolov7\_tiny.onnx
  - 驗證 yolov7\_tiny.onnx 推論

## Step 2: ONNX -> Modified ONNX

### 檢測 ONNX

- ✓ 參考 amba 文件：Ambarella-CV2x\_Onnx\_Utils\_Onnx\_Print\_Graph\_Summary
- ✓ 執行所有功能前都要先 VM Build enviroment:

```
# Enter VM ONNX
```

```
$ source /usr/local/amba-cv-tools-basic.2.5.6.0.1245.ubuntu-20.04/env/cv22.env
```

- ✓ 將 yolov7\_tiny.onnx 透過 amba 檢測程式 onnx\_print\_graph\_summary.py 查看有無不支援的內容：

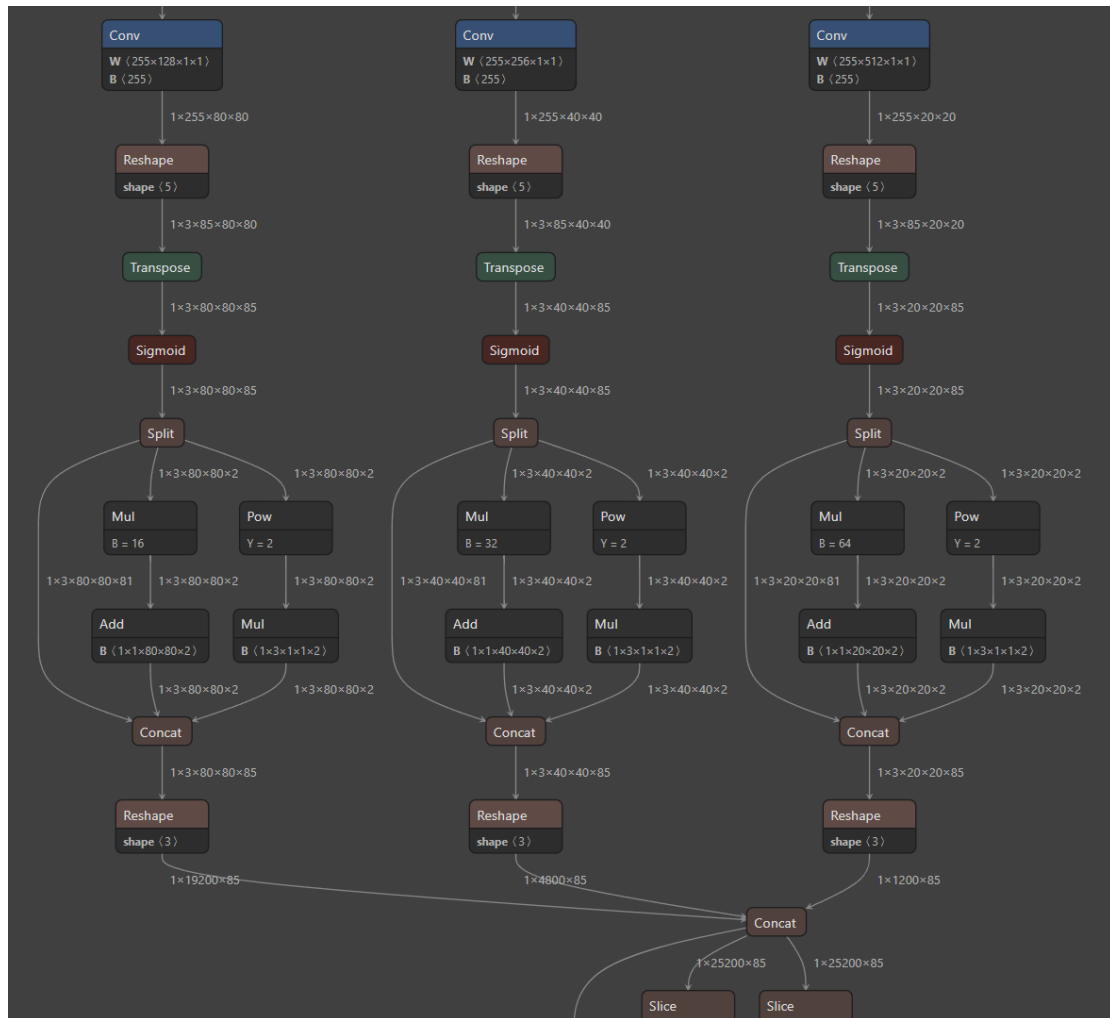
```
$ cd /usr/local/amba-cv-tools-basic.2.5.6.0.1245.ubuntu-
```

```
20.04/cv2/tv2/release/CommonCnnUtils/cv2.basic.2.5.6.0.1245.ubu2004
```

```
$ onnx_print_graph_summary.py -p yolov7_tiny.onnx
```

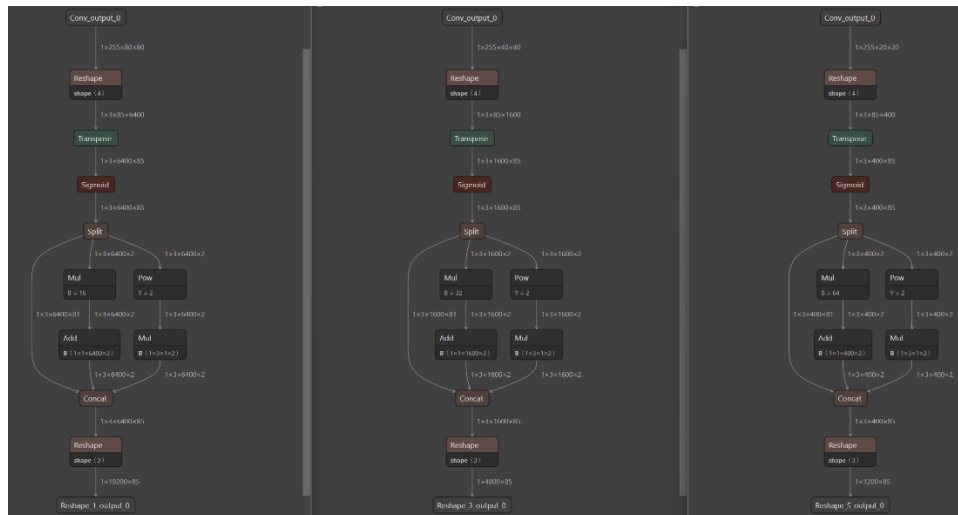
```
lilin@lilin-VirtualBox: /usr/local/amba-cv-tools-basic.2.5.6.0.1245.ubuntu-20.04/cv2/tv2/release/CommonCnnUtils/cv2.basic.2.5.6.0.1245.ubu2004$ onnx_print_graph_summary.py -p yolov7_tiny.onnx
INFO: 10/25/2024 10:37:18.897137 logger.py:372 [PrintGraphSummary] Application logs will be written to the log directory: ./logs
INFO: 10/25/2024 10:37:19.469580 onnx_print_graph_summary.py:199 [PrintGraphSummary] ***** Metadata *****
INFO: 10/25/2024 10:37:19.469762 onnx_print_graph_summary.py:248 [PrintGraphSummary] ONNX Version: 1.8.1
INFO: 10/25/2024 10:37:19.469873 onnx_print_graph_summary.py:249 [PrintGraphSummary] ONNX Path: /home/lilin/.local/lib/python3.8/site-packages/onnx
INFO: 10/25/2024 10:37:19.469963 onnx_print_graph_summary.py:256 [PrintGraphSummary] IR Version: 7
INFO: 10/25/2024 10:37:19.470051 onnx_print_graph_summary.py:251 [PrintGraphSummary] Producer: pytorch-2.4.1
INFO: 10/25/2024 10:37:19.470137 onnx_print_graph_summary.py:252 [PrintGraphSummary] Operator Sets: ai.onnx v12
INFO: 10/25/2024 10:37:19.470224 onnx_print_graph_summary.py:199 [PrintGraphSummary] ***** Graph summary *****
INFO: 10/25/2024 10:37:19.470317 onnx_print_graph_summary.py:353 [PrintGraphSummary] Total operators: 192
INFO: 10/25/2024 10:37:19.470446 onnx_print_graph_summary.py:354 [PrintGraphSummary] Total (non-constant) operators: 192
INFO: 10/25/2024 10:37:19.470609 onnx_print_graph_summary.py:358 [PrintGraphSummary] Primary inputs (1):
INFO: 10/25/2024 10:37:19.470756 onnx_print_graph_summary.py:360 [PrintGraphSummary] -> Tensor(images, shape=[1 x 3 x 640 x 640], dtype=float32)
INFO: 10/25/2024 10:37:19.470894 onnx_print_graph_summary.py:362 [PrintGraphSummary] Primary outputs (1):
INFO: 10/25/2024 10:37:19.471013 onnx_print_graph_summary.py:364 [PrintGraphSummary] -> Tensor(output, shape=[output_dim x 7], dtype=float32)
INFO: 10/25/2024 10:37:19.471188 onnx_print_graph_summary.py:379 [PrintGraphSummary] Unsupported tensors with rank > 4 (33):
INFO: 10/25/2024 10:37:19.471288 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Reshape_output_0' [1 x 3 x 85 x 80 x 80]
INFO: 10/25/2024 10:37:19.471288 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/transpose_output_0' [1 x 3 x 80 x 80 x 85]
INFO: 10/25/2024 10:37:19.471375 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Sigmoid_output_0' [1 x 3 x 80 x 80 x 85]
INFO: 10/25/2024 10:37:19.471462 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Split_output_0' [1 x 3 x 80 x 80 x 2]
INFO: 10/25/2024 10:37:19.472370 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Split_output_1' [1 x 3 x 80 x 80 x 2]
INFO: 10/25/2024 10:37:19.472519 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Split_output_2' [1 x 3 x 80 x 80 x 81]
INFO: 10/25/2024 10:37:19.472633 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Mul_output_0' [1 x 3 x 80 x 80 x 2]
INFO: 10/25/2024 10:37:19.472723 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Add_output_0' [1 x 3 x 80 x 80 x 2]
INFO: 10/25/2024 10:37:19.472811 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Power_output_0' [1 x 3 x 80 x 80 x 2]
INFO: 10/25/2024 10:37:19.472897 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Mul_1_output_0' [1 x 3 x 80 x 80 x 2]
INFO: 10/25/2024 10:37:19.472982 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Concat_output_0' [1 x 3 x 80 x 80 x 85]
INFO: 10/25/2024 10:37:19.473068 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Reshape_2_output_0' [1 x 3 x 85 x 40 x 40]
INFO: 10/25/2024 10:37:19.473154 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/transpose_1_output_0' [1 x 3 x 40 x 40 x 85]
INFO: 10/25/2024 10:37:19.473240 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Sigmoid_1_output_0' [1 x 3 x 40 x 40 x 85]
INFO: 10/25/2024 10:37:19.473326 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Split_1_output_0' [1 x 3 x 40 x 40 x 2]
INFO: 10/25/2024 10:37:19.473413 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Split_1_output_1' [1 x 3 x 40 x 40 x 2]
INFO: 10/25/2024 10:37:19.473508 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Split_1_output_2' [1 x 3 x 40 x 40 x 81]
INFO: 10/25/2024 10:37:19.473596 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Mul_2_output_0' [1 x 3 x 40 x 40 x 2]
INFO: 10/25/2024 10:37:19.473683 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Add_1_output_0' [1 x 3 x 40 x 40 x 2]
INFO: 10/25/2024 10:37:19.473771 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Power_1_output_0' [1 x 3 x 40 x 40 x 2]
INFO: 10/25/2024 10:37:19.473859 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Mul_3_output_0' [1 x 3 x 40 x 40 x 2]
INFO: 10/25/2024 10:37:19.473946 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Concat_1_output_0' [1 x 3 x 40 x 40 x 85]
INFO: 10/25/2024 10:37:19.474033 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Reshape_4_output_0' [1 x 3 x 85 x 20 x 20]
INFO: 10/25/2024 10:37:19.474120 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/transpose_2_output_0' [1 x 3 x 20 x 20 x 85]
INFO: 10/25/2024 10:37:19.474208 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Sigmoid_2_output_0' [1 x 3 x 20 x 20 x 85]
INFO: 10/25/2024 10:37:19.474297 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Split_2_output_0' [1 x 3 x 20 x 20 x 2]
INFO: 10/25/2024 10:37:19.474384 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Split_2_output_1' [1 x 3 x 20 x 20 x 2]
INFO: 10/25/2024 10:37:19.474471 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Split_2_output_2' [1 x 3 x 20 x 20 x 81]
INFO: 10/25/2024 10:37:19.474560 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Mul_4_output_0' [1 x 3 x 20 x 20 x 2]
INFO: 10/25/2024 10:37:19.474647 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Add_2_output_0' [1 x 3 x 20 x 20 x 2]
INFO: 10/25/2024 10:37:19.474735 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Power_2_output_0' [1 x 3 x 20 x 20 x 2]
INFO: 10/25/2024 10:37:19.474821 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Mul_5_output_0' [1 x 3 x 20 x 20 x 2]
INFO: 10/25/2024 10:37:19.474907 onnx_print_graph_summary.py:381 [PrintGraphSummary] -> '/model/model.77/Concat_2_output_0' [1 x 3 x 20 x 20 x 85]
INFO: 10/25/2024 10:37:19.475018 onnx_print_graph_summary.py:384 [PrintGraphSummary] Unsupported operators (1):
INFO: 10/25/2024 10:37:19.475127 onnx_print_graph_summary.py:389 [PrintGraphSummary] -> NonMaxSuppression x 1
INFO: 10/25/2024 10:37:19.475215 onnx_print_graph_summary.py:403 [PrintGraphSummary] Pass-through operators (1):
INFO: 10/25/2024 10:37:19.475301 onnx_print_graph_summary.py:405 [PrintGraphSummary] -> Cast x 3
INFO: 10/25/2024 10:37:19.475387 onnx_print_graph_summary.py:199 [PrintGraphSummary] *****
```

- ✓ 問題主要包含 3 個：
  - 部分 reshape layer 將 dimension 增加到 5，但 amba 不支援 dimension > 4 的 tensor
  - 不支援 NonMaxSuppression，其作用為過濾掉與先前選取的框具有較高交集 (IOU) 重疊的框，刪除分數小於 score\_threshold 的邊界框
  - 不支援 cast，其作用為將輸入 tensor 轉換成指定的資料類型，並以轉換後的類型傳回相同大小的輸出 tensor
  - 可以使用[可視化工具](#)查看架構的節點關係、名稱或 tensor 大小，方便找出有問題的地方



## 修改維度

- ✓ 參考 amba 文件：Ambarella-CV2x\_Onnx\_Utils\_Onnx\_Graph\_Surgery，章節 4.2.3  
Generating Replacement Subgraphs 有範例程式
- ✓ 目的：建立新的 subgraph，取代掉原先不支援的局部架構
- ✓ 這邊把所有 5 維 tensor 的 80x80, 40x40, 20x20 特徵層合併成 6400, 1600, 400
  - 可再評估不同的合併方式對模型效果的影響
- ✓ 重拉過需替換的部分，並注意替換的開頭和結尾節點命名和大小需與原架構相同，amba 的程式才能成功替換
- ✓ 下圖從左到右的 subgraph 建立程式依序為：replace\_1.py, replace\_2.py, replace\_3.py，運行完會產生 3 個 onnx 檔：replace\_subgraph\_1.onnx, replace\_subgraph\_2.onnx, replace\_subgraph\_3.onnx



- ✓ 建立 config.json，將要替換的 3 個 onnx 檔資訊寫入，確保 node 命名符合原架構和新架構

- ✓ 透過以下指令進行替換：

```
$ cd /usr/local/amba-cv-tools-basic.2.5.6.0.1245.ubuntu-
```

```
20.04/cv2/tv2/release/CommonCnnUtils/cv2.basic.2.5.6.0.1245.ubu2004
```

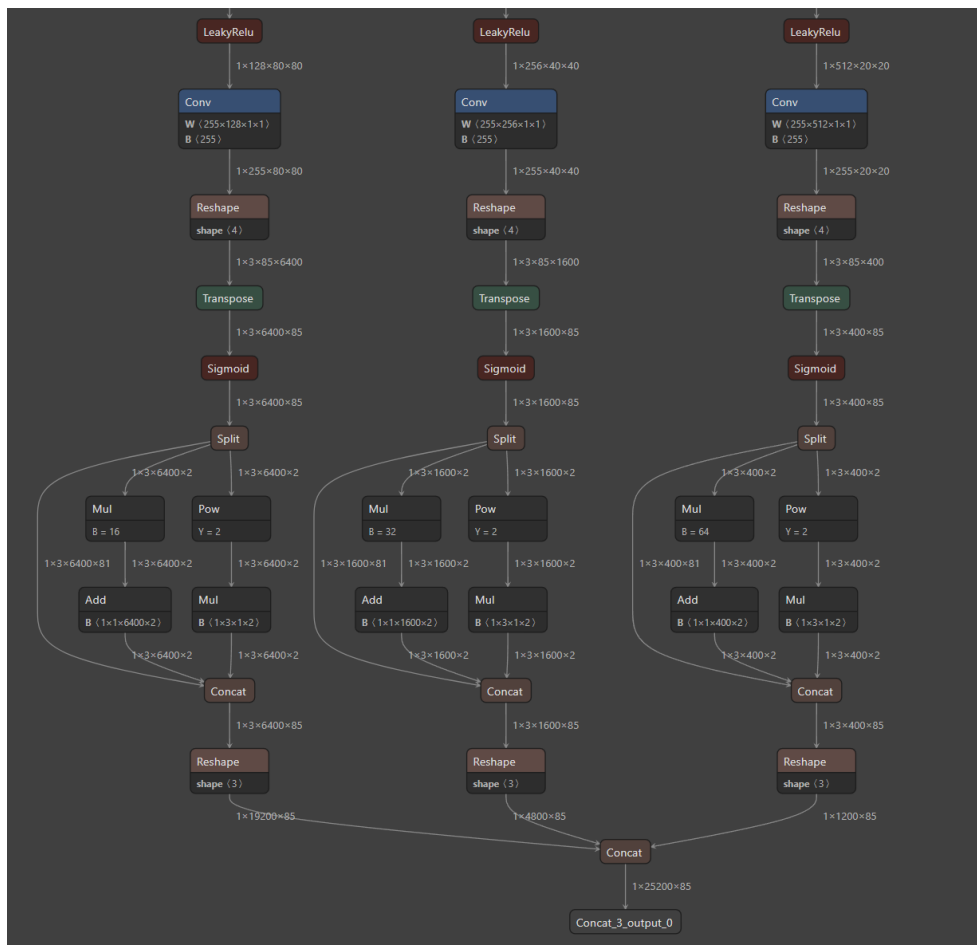
```
$ graph_surgery.py -fw onnx -m version1.onnx -t ReplaceSubgraph=path/to/config.json -o version2.onnx
```

```
# 如果出現 graph summary recommendations 就依照他上面建議用的透過 -t 加入
```

```
$ graph_surgery.py -fw onnx -m version2.onnx -t CVFlow
```

## 修改 NonMaxSuppression, Cast

- ✓ [參考連結](#)中解釋了 Concat 的輸出 [1, 25200, 85] 含意：
  - 1: Batch size
  - 25200 =  $3 \times (20 \times 20 + 40 \times 40 + 80 \times 80)$ : 總 Anchor box 數量，每個特徵點會對應 3 個 Anchor box。20, 40, 80 分別為三個不同大小的特徵層
  - 85 = (4+1+80):
    - 前 4 個參數用於判斷每一個特徵點的迴歸參數，迴歸參數調整後可獲得 Bounding Box 的位置
    - 第 5 個參數為 Bounding Box 的信心分數，用來判斷每一個特徵點是否包含預測類別
    - 最後 80 個參數為特徵點分別對 80 個 COCO 類別的預測分數。
- ✓ 目的：拿掉包含到 NonMaxSuppression, cast 的架構，將模型最後的 output 修改成 concat 的輸出
- ✓ output\_modified.py 程式最終調整完畢的模型架構為下圖 (接近模型輸出的地方)：

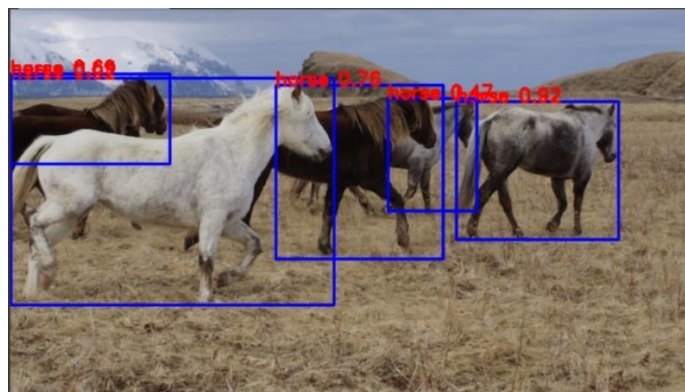


✓ 再次執行檢測可看到輸入輸出正常，且已無問題

```

lililili-VirtualBox: /usr/local/ana-cv-tools-basic.2.9.6.8.1245.ubuntu-20.04/cv2/tv2/release/CommonCVUtils/cv2.basic.2.9.6.8.1245.ubuntu2004$ onnx_print_graph_summary.py -p version3.onnx
INFO: 10/25/2024 11:06:01.889574 logger.py:372 [PrintGraphSummary] Application logs will be written to the log directory: ./logs
INFO: 10/25/2024 11:06:02.544019 onnx_print_graph_summary.py:199 [PrintGraphSummary] ***** Metadata *****
INFO: 10/25/2024 11:06:02.544254 onnx_print_graph_summary.py:248 [PrintGraphSummary] ONNX Version: 1.8.1
INFO: 10/25/2024 11:06:02.544339 onnx_print_graph_summary.py:249 [PrintGraphSummary] ONNX Path: /home/lililili/.local/lib/python3.8/site-packages/onnx
INFO: 10/25/2024 11:06:02.544412 onnx_print_graph_summary.py:250 [PrintGraphSummary] IR Version: 7
INFO: 10/25/2024 11:06:02.544483 onnx_print_graph_summary.py:251 [PrintGraphSummary] Producer: pytorch-2.4.1
INFO: 10/25/2024 11:06:02.544551 onnx_print_graph_summary.py:252 [PrintGraphSummary] Operator Sets: ai.onnx.v12
INFO: 10/25/2024 11:06:02.544620 onnx_print_graph_summary.py:199 [PrintGraphSummary] ***** Graph summary *****
INFO: 10/25/2024 11:06:02.544689 onnx_print_graph_summary.py:353 [PrintGraphSummary] Total operators: 166
INFO: 10/25/2024 11:06:02.544758 onnx_print_graph_summary.py:354 [PrintGraphSummary] Total (non-constant) operators: 166
INFO: 10/25/2024 11:06:02.544825 onnx_print_graph_summary.py:358 [PrintGraphSummary] Primary inputs (1):
INFO: 10/25/2024 11:06:02.544947 onnx_print_graph_summary.py:360 [PrintGraphSummary] -> Tensor(images, shape=[1 x 3 x 640 x 640], dtype=float32)
INFO: 10/25/2024 11:06:02.545018 onnx_print_graph_summary.py:362 [PrintGraphSummary] Primary outputs (1):
INFO: 10/25/2024 11:06:02.545102 onnx_print_graph_summary.py:364 [PrintGraphSummary] -> Tensor(model/model.77/concat_3_output_0, shape=[1 x 25200 x 85], dtype=float32)
INFO: 10/25/2024 11:06:02.545813 onnx_print_graph_summary.py:199 [PrintGraphSummary] *****
  
```

✓ 透過 inference.ipynb 將修正後的 onnx 輸出接出，並加入 NMS 在本機上推論進行結果驗證，確認修改的模型沒有問題：



## Step 3: Modified ONNX -> amba bin

### input image

- ✓ 產生作為 IPCAM 的 input image bin 檔
- ✓ /dra\_img/ 裡面裝一張圖片就好，不然後面 onnxparser 會噴錯
- ✓ 執行完會產生出紀錄圖片路徑的 txt 檔和轉成 bin 檔的圖片

```
$ cd /usr/local/amba-cv-tools-basic.2.5.6.0.1245.ubuntu-  
20.04/cv2/tv2/release/CommonCnnUtils/cv2.basic.2.5.6.0.1245.ubu2004  
  
$ python3 gen_image_list.py \  
  
-f /home/lilin/cnngen_demo/onnx/demo_networks/yolov7_tiny/dra_img \  
  
-o  
/home/lilin/cnngen_demo/out/demo_networks/onnx_yolov7s_tiny/dra_image_bin/img_list.  
txt> -ns \  
  
-e jpg -c 0 -d 0,0 -r 640,640 \  
  
-bf /home/lilin/cnngen_demo/out/demo_networks/onnx_yolov7s_tiny/dra_image_bin \  
  
-bo  
/home/lilin/cnngen_demo/out/demo_networks/onnx_yolov7s_tiny/dra_image_bin/dra_bin_  
list.txt
```

### Parser

- ✓ 接下來產生很多 weight, layer bin 檔在 /dra\_image\_bin/ 中，也會包含 onnx\_yolov7t.json 檔

```
$ cd /usr/local/amba-cv-tools-basic.2.5.6.0.1245.ubuntu-  
20.04/cv2/tv2/release/AmbaCnnUtils/cv2.basic.2.5.6.0.1245.ubu2004/parser/onnx  
  
$ python3 onnxparser.py \  
  
-m "/usr/local/amba-cv-tools-basic.2.5.6.0.1245.ubuntu-  
20.04/cv2/tv2/release/CommonCnnUtils/cv2.basic.2.5.6.0.1245.ubu2004/version3.onnx" \  
  
-isrc  
"is:1,3,640,640|i:1|idf:0,0,8,0|i:images=/home/lilin/cnngen_demo/out/demo_networks/onnx_  
_yolov7s_tiny/dra_image_bin/dra_bin_list.txt" \  
  
-o "onnx_yolov7t" \  
  
-of "/home/lilin/cnngen_demo/out/demo_networks/onnx_yolov7s_tiny/dra_image_bin/"
```



```
-odst "o:/model/model.77/Concat_3_output_0|odf:fp32"
```

✓ 出現以下畫面表示成功

```
[info] drc proc time = 2.382 s @ 100 %
[info] drc num frames = 1
[info] drc proc time = 2.382277 s
[info] drc pproc time = 0.000652 s
[info] drc total time = 4.842634 s
[info] dr2 time = ( 2.591 calc + 3.369 stat) s x 1 frames
```

## Acinference

✓ 執行完以下指令後，即可產生最終的 weight bin 檔：onnx\_yolov7t\_cavalry.bin

```
$ cd /home/lilin/cnngen_demo/out/demo_networks/onnx_yolov7s_tiny/dra_image_bin
```

```
$ acinference onnx_yolov7t.json \
```

```
-i
```

```
images=/home/lilin/cnngen_demo/out/demo_networks/onnx_yolov7s_tiny/dra_image_bin/
dra_bin_list.txt -o output
```

```
$ vas -auto -show-progress onnx_yolov7t.vas
```

```
$ cavalry_gen -V 2.2.8.1 -d vas_output/ -f onnx_yolov7t_cavalry.bin -v
```

```
lilin@lilin-VirtualBox:~/cnngen_demo/out/demo_networks/onnx_yolov7s_tiny/dra_image_bin$ acinference onnx_yolov7t.json -i images=/home/lilin/cnngen_demo/out/demo_networks/onnx_yolov7s_tiny/dra_image_bin/dra_bin_list.txt -o output
acinference: version cv2.basf:2.5.6.0.1245-ide0004 (9322cec70)
acinference: started at Fri Oct 25 11:41:35 2024
acinference: inferecing 'onnx_yolov7t.json', cpu-only:
[info] fwd time = 2.923 s x 1 frames
[info] fwd total time = 2.923179 s
acinference: end of inference.
lilin@lilin-VirtualBox:~/cnngen_demo/out/demo_networks/onnx_yolov7s_tiny/dra_image_bin$ vas -auto -show-progress onnx_yolov7t.vas
onnx_yolov7t: EXPAND 100 (expansion progress: 625)
onnx_yolov7t: EXPAND 161 (expansion progress: 1000)
onnx_yolov7t: DONE expansion
onnx_yolov7t: DONE validation
onnx_yolov7t: DONE analyzing
onnx_yolov7t: DONE opt
onnx_yolov7t: DONE buffering
onnx_yolov7t: SPLIT 0 (split progress: 145)
onnx_yolov7t: SPLIT 1 (split progress: 265)
onnx_yolov7t: SPLIT 2 (split progress: 285)
onnx_yolov7t: SPLIT 3 (split progress: 505)
onnx_yolov7t: SPLIT 4 (split progress: 525)
onnx_yolov7t: SPLIT 5 (split progress: 665)
onnx_yolov7t: SPLIT 6 (split progress: 815)
onnx_yolov7t: SPLIT 7 (split progress: 865)
onnx_yolov7t: SPLIT 8 (split progress: 1005)
onnx_yolov7t: DONE split
onnx_yolov7t: DONE all
lilin@lilin-VirtualBox:~/cnngen_demo/out/demo_networks/onnx_yolov7s_tiny/dra_image_bin$ cavalry_gen -V 2.2.8.1 -d vas_output/ -f onnx_yolov7t_cavalry.bin
Using cavalry_gen version 2.2.8.1
Save to file onnx_yolov7t_cavalry.bin with file size 6355704.
```

## Step 4: amba bin 輸出驗證

✓ 參考 amba 文件 (可搜關鍵字 test\_nnctrl)：

- Ambarella\_CV2x\_UG\_Flexible\_Linux\_SDK3.0\_CNNGen\_Development\_Flow
- Ambarella\_CV2x\_DG\_Flexible\_Linux\_SDK3.0\_CVflow\_FAQ

✓ 上傳到 IPCAM 推論單張圖片

```
# IPCAM
```

```
$ killall -9 inetd; rm /etc/inetd.conf; echo 221 stream tcp nowait root ftpd ftpd -w / >
```

```
/etc/inetd.conf ;inetd
```

```
$ killall -9 monitor_ipcam; killall -9 ipcam_service
```

```
$ tftp 192.168.50.200 -gr onnx_yolov7t_cavalry.bin
```

```
$ tftp 192.168.50.200 -gr bus.bin
```

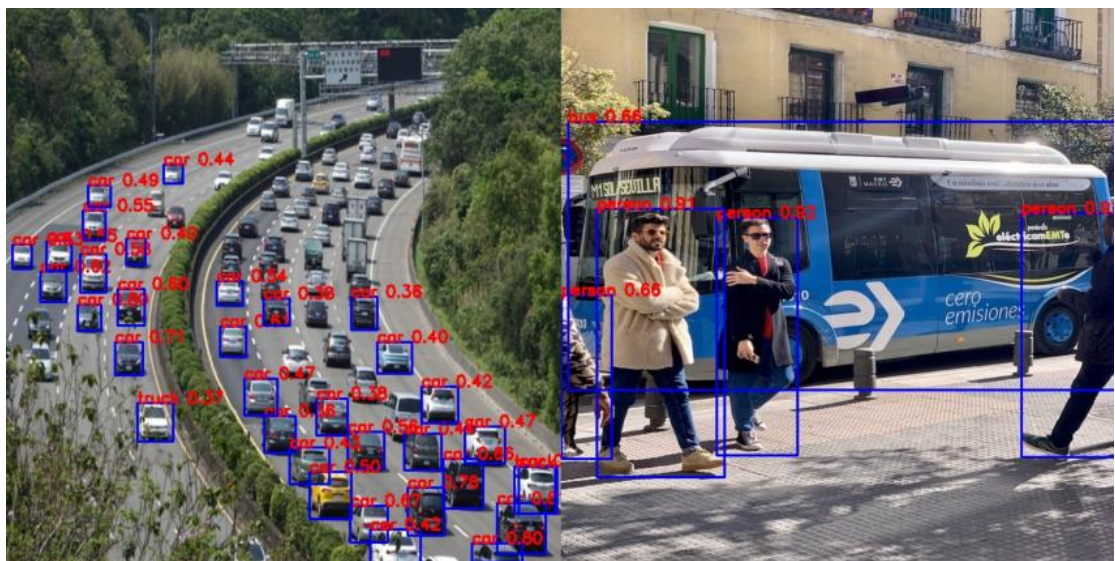
```
$ modprobe cavalry;/usr/local/bin/cavalry_load -f /lib/firmware/cavalry.bin -r
```

```
$ test_nnctrl -b onnx_yolov7t_cavalry.bin --in images=bus.bin --out
```

```
/model/model.77/Concat_3_output_0=output.bin -e
```

```
$ tftp -pr output.bin 192.168.50.200
```

- ✓ 就可以看到輸出一個 output.bin 檔，再透過 inference.ipynb 將模型的輸出接入 NMS 並產生畫出 Bounding Box 的結果圖



### LILIN\_SDK3.0.9

- ✓ 如果要調整 test\_nnctrl 就重新編譯後上傳到 IPCAM

```
# Enter VM SDKKK
```

```
$ cd
```

```
/home/lilin/Desktop/sdk_build/cv2x_sdk30_patch9/ambarella/boards/cv22_walnut
```

```
$ source ../../build/env/aarch64-linux5.4-gcc.env
```

```
$ make test_nnctrl
```

```
$ sudo cp ../../out/cv22_walnut/fakeroot/usr/local/bin/test_nnctrl ~/Desktop/sdk/
```

```
# IPCAM
```

```
$ tftp 192.168.50.200 -gr test_nnctrl
```

```
$ chmod 777 test_nnctrl
```

```
./test_nnctrl -b onnx_yolov7t_cavalry.bin --in images=bus.bin --out
```

```
/model/model.77/Concat_3_output_0=output.bin -e
```



```
Build libnnctrl.so Done.  
Build libcavalry_mem.so Done.  
Build libvproc.so Done.  
Build test_nnctrl_live Done.
```

- ✓ 模型輸出的內容為 structure io\_match 的 \*virt\_addr，可以將 \*virt\_addr 轉成 float 型態印出查看，應與上面 output.bin 內容一樣：

```
void print_data_as_float(uint8_t *virt_addr, uint32_t data_count, struct io_data_fmt  
*data_fmt) {  
    float *f32_ptr;  
    uint32_t i;  
  
    if (data_fmt->size == 2) { // 32-bit float  
        f32_ptr = (float *)virt_addr;  
        printf("Data as 32-bit float:\n");  
        for (i = 0; i < data_count; i++) {  
            printf("%f ", f32_ptr[i]);  
        }  
        printf("\n");  
    } else {  
        printf("Unsupported data size for floating-point format.\n");  
    }  
}
```