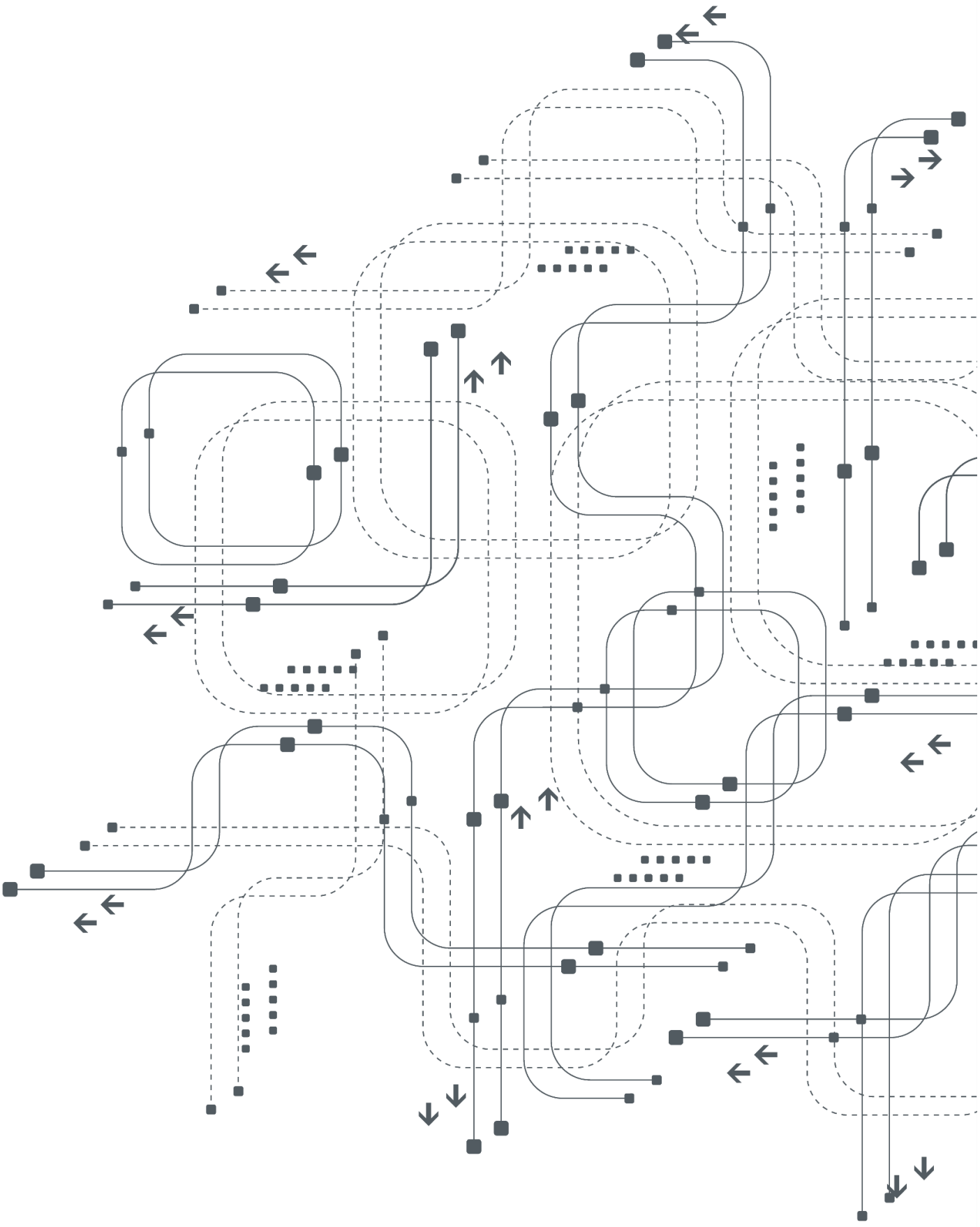


Tachy-Compile

가이드북

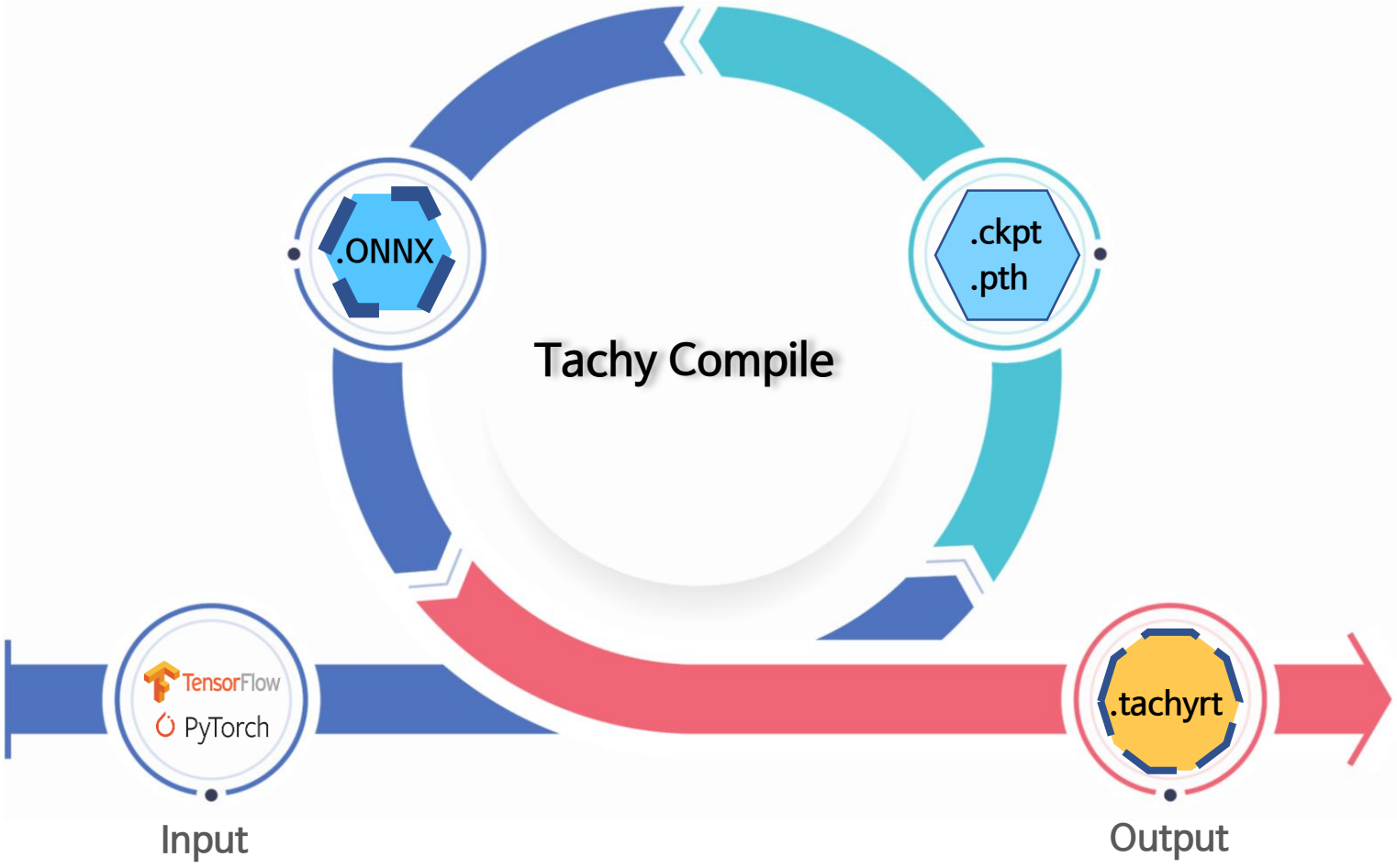
목차

1. 개요

1-1) Tachy Compile 란?

Tachy Compile은 다양한 딥러닝 프레임워크(TensorFlow, PyTorch 등)에서 학습된 모델을 고성능 NPU 환경에 최적화하여 실행할 수 있도록 변환하는 컴파일러입니다. 특히 ONNX(Open Neural Network Exchange) 포맷을 중심으로 모델을 표준화하고, 이를 TACHY 전용 포맷(.tachy)으로 변환함으로써, 경량화, 고속화, 실시간 추론을 지원합니다.

〈포맷 최적화 흐름도〉



1-2) 주요 특징 및 장점

→ 학습, 변환, 최적화, 배포 과정을 하나의 흐름으로 통합

→ 그래프 최적화 및 메모리 경량화

→ 다양한 플랫폼 대응 및 호환성

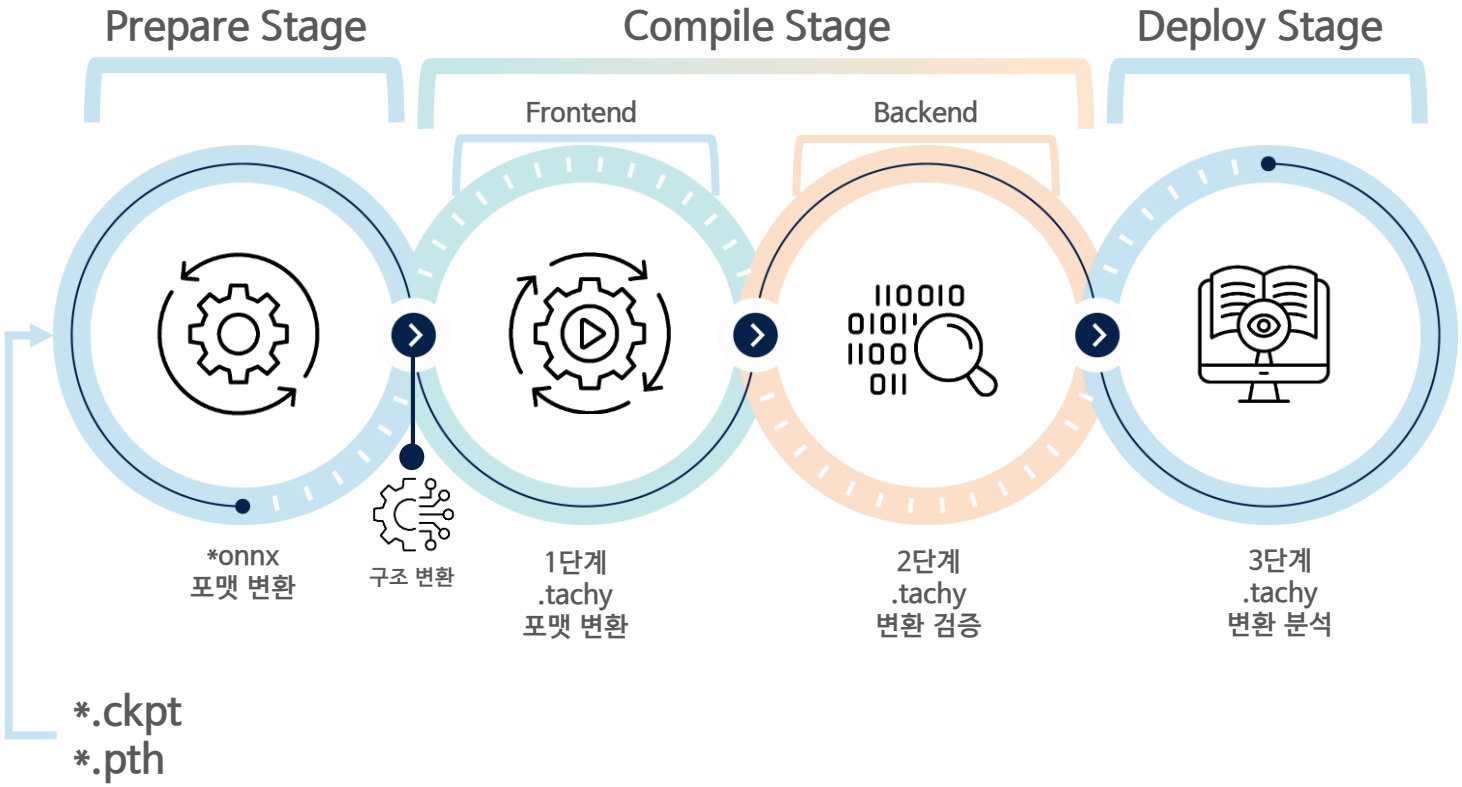
→ NPU 적용을 위한 최적화

2. 포맷 최적화 흐름

2-1) 모델 변환 개요

User는 TensorFlow나 PyTorch 등에서 XWN이 적용된 사전 학습된 모델을 준비하고, 이를 ONNX 포맷으로 변환한 뒤, Tachy Compiler를 통해 최종적으로 .tachy 파일을 생성하여 Deeper-I NPU 보드에 적용시키기 위한 파일 최적화 작업을 진행합니다.

2-2) ONNX 변환 과정 → Tachy 포맷 변환



학습된 모델을 ONNX로 변환한 후, ONNX Modifier를 통해 연산자 구조 수정 및 불필요한 노드 제거를 수행합니다. 최적화된 ONNX 모델은 XWN 경량화를 거쳐 실행 효율을 극대화합니다. 이후 Tachy Compiler가 최종적으로 메모리 구조와 연산 흐름을 재구성하여 .tachy 포맷으로 변환합니다.

3-1) 사전 준비

☐ 학습 환경 구축

GPU 및 CUDA 호환성 확인

```
$ nvidia-smi
```

자신의 GPU가 지원하는 CUDA 버전 확인.

☐ Dataset 준비

```
{dataset_path}/
├── train/
│   ├── images/
│   └── labels/
├── val/
│   ├── images/
│   └── labels/
└── test/
    ├── images/
    └── labels/
```

☐ 라벨 포맷 (YOLO 형식)

```
$ class x_center y_center width height
```

3-2) Main Training

☐ Dataset YAML 파일 생성

```
$ git clone http://asdf.com
$ cd yolov9
```

```
$ cd data
$ vi {DATASET_NAME.yaml}   ※ New 파일 생성
```

```yaml
path: /mnt/DPI-SHARED/tom/20240625_yolov9/dataset/FP # dataset root dir
train: train/images
val: val/images
test: test/images

# Classes
names:
  0: person
  1: car
  2: bus
  3: truck
```

☐ 모델 YAML 파일 생성

```
$ cd ../models/detect
$ cp bsnet-t.yaml {bsnet-t-DATASET_NAME.yaml}
$ vi {bsnet-t-DATASET_NAME.yaml}
```

수정: nc (number of classes) 값을 데이터셋에 맞게 변경

```yaml
# YOLOv9

# parameters
nc: 4  # number of classes
```

Example {bsnet-t-DATASET_NAME.yaml}

□ 학습 스크립트 설정

```
$ cd ../..
$ vi train    ※ New 파일 생성
```

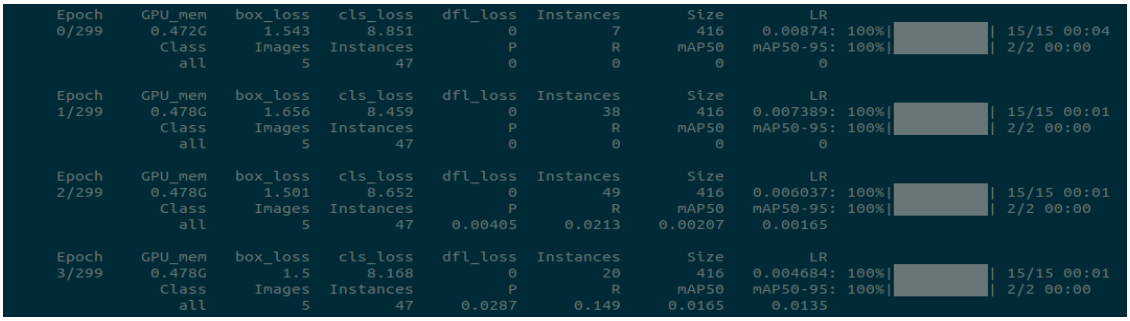
예) train 명령어 - train 파일 열고, 하단에 있는 코드 붙여넣기

```
python train.py --workers 8 ₩
                --device 0 ₩
                --batch 8 ₩
                --data data/{DATASET_NAME.yaml} ₩
                --img 416 ₩
                --cfg models/detect/bsnet-t-{DATASET_NAME.yaml} ₩
                --weights '' ₩
                --name {guide} ₩
                --hyp hyp.pretrained.yaml ₩
                --min-items 0 ₩
                 --epochs 300 ₩
                --close-mosaic 15
```

□ 학습 실행

```
$ ./train    파일 실행
```

결과: runs/train/{guide}/weights/best.pt 생성



3-3) 학습 결과 검증

□ 검증 스크립트 설정 및 실행

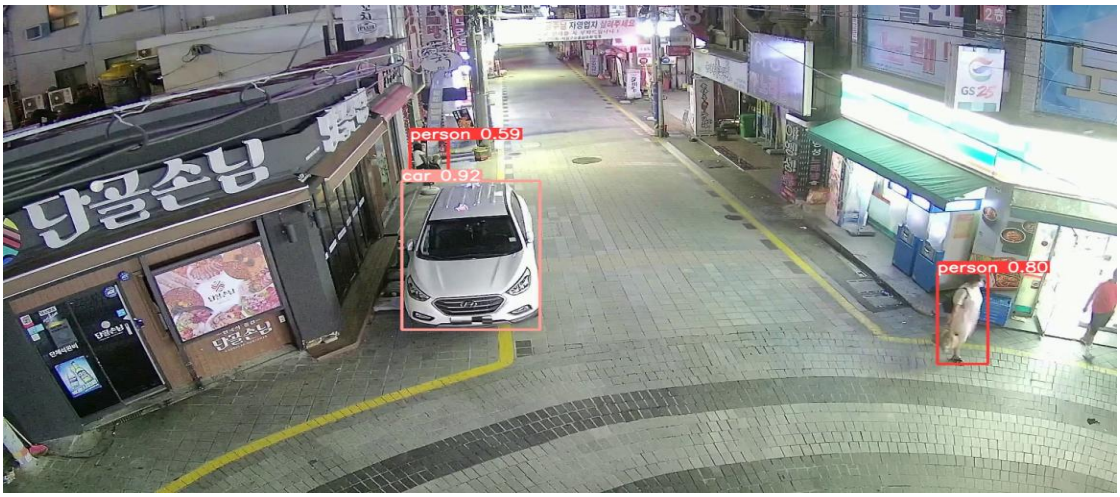
```
$ vi test    ※ New 파일 생성
```

예) test 명령어 - test 파일 열고, 하단에 있는 코드 붙여넣기

```
python detect.py --source {dataset_path}/test/images ₩
                 --img 416 ₩
                 --device 0 ₩
                 --weights runs/train/{guide}/weights/best.pt ₩
                 --name {guide} ₩
                 --save-txt
```

```
$ ./test    파일 실행
```

결과: runs/detect/{guide} 디렉터리에 검출 결과 저장



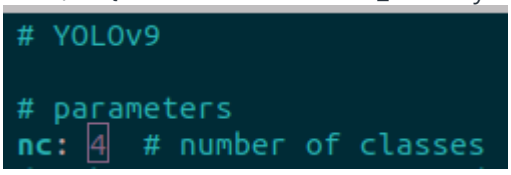
Example detect image

3-4) 최적화 Training (옵션)

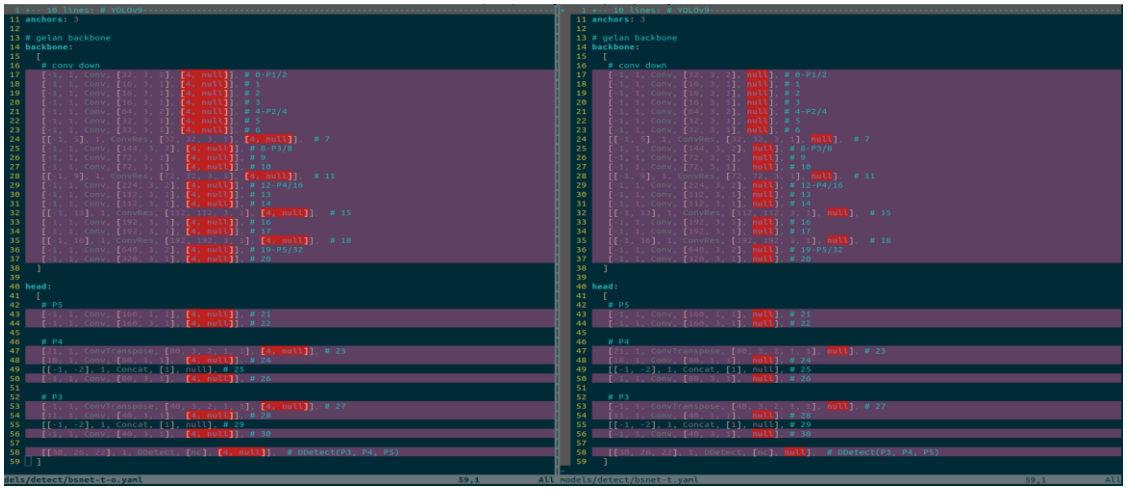
(선택 사항: 양자화/최적화 Training을 진행할 경우)

☐ 최적화용 모델 YAML 생성

```
$ cd ../models/detect
$ cp bsnet-t-o.yaml {bsnet-t-o-DATASET_NAME.yaml}
$ vi {bsnet-t-o-DATASET_NAME.yaml}
```



Example {bsnet-t-o-DATASET_NAME.yaml}


diff main training model.yaml, opt training model.yaml

opt training differ bit : int / 4 / Quantization range (bit-1 ** 2)

□ 최적화 학습 스크립트 설정

```
$ cd ../..
$ vi train    ※ New 파일 생성
```

예) train 명령어 - train 파일 열고, 하단에 있는 코드 붙여넣기

```
python train.py --workers 8 ₩
        --device 0 ₩
        --batch 8 ₩
        --data data/ {DATASET_NAME.yaml} ₩
        --img 416 ₩
        --cfg models/detect/ {bsnet-t-o-DATASET_NAME.yaml} ₩
        --weights 'runs/train/{guide}/weights/best.pt' ₩
        --name {guide-opt} ₩
        --hyp hyp.optimization.yaml ₩
        --min-items 0 ₩
         --epochs 100 ₩
        --close-mosaic 15
```

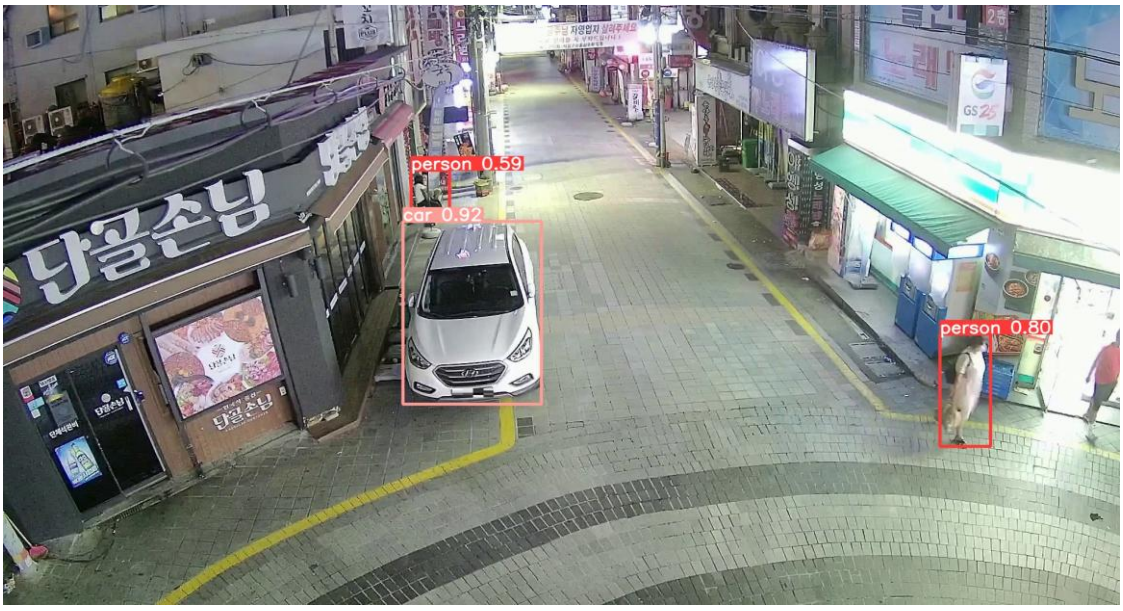
* Pay attention to yaml file name in --data and --cfg options

□ 최적화 학습 및 검증

```
$ ./train    파일 실행
```

□ 최적화된 결과를 다시 검증

```
$ vi test
python detect.py --source {/dataset_path/test/images} ₩
        --img 416 –device 0 ₩
        --weights 'runs/train/{guide-opt}/weights/best.pt' ₩
        --name {guide-opt} –save-txt
$ ./test    파일 실행
```



3-5) 모델 변환 (PyTorch → ONNX → TACHY)

Compile 이란?

pt 등으로 학습된 yolov9 모델을 tachy-bs npu 가 이해할 수 있는 tachy runtime 패키지(.tachyrt, param.bin 등)로 자동 컴파일 하는 일괄 스크립트 입니다.

☐ compile 스크립트 수정

yolov9 프로젝트 내부의 compile 스크립트를 수정합니다.

vi compile ※ 파일 열기

〈compile 스크립트 내 주요 변수 설명〉 ※ 필수설정

config	default	explanation
입력경로		
PRE_PARAM_DIR	runs/golden/drone/bsnet-t-drone3c/weights	**최적화 전** 모델 가중치(.pt)가 위치한 디렉터리 경로
OPT_PARAM_DIR	runs/golden/drone/bsnet-t-o-drone3c/weights	**최적화 후** 모델 가중치(.pt)가 위치한 디렉터리 경로
사이즈		
ONNX_INPUT_SHAPE	"1 3 320 416"	ONNX 변환 시 입력 텐서 형태 *(B, C, H, W)*
TACHY_INPUT_SHAPE	"1 320 416 3"	TACHY-Compiler용 입력 텐서 형태 *(B, H, W, C)*

〈옵션설정〉

config	default	explanation
TRAIN_FILE_TYPE	"pt_yolov9"	배포 스크립트에서 사용할 학습 파일 형식 (PyTorch-YOLOv9)
ONNX_FILE_TYPE	"pt2onnx"	PyTorch → ONNX 변환 스크립트 유형
SRC_PRE_PATH	os.path.join(PRE_PARAM_DIR, "best.pt")	최적화 전 가중치 파일 전체 경로
SRC_OPT_PATH	os.path.join(OPT_PARAM_DIR, "best.pt")	최적화 후 가중치 파일 전체 경로
WORK_DIR	datetime.datetime.now().strftime("20%y%m%d_%H%M%S")	실행 시각으로 생성되는 작업 디렉터리 이름 (예: 20250429_153240)
TEMP_DIR	os.path.join(WORK_DIR, "temp")	작업 디렉터리 내부의 임시 파일 저장 폴더
DST_EVAL_PATH	os.path.join(WORK_DIR, "eval.pt")	평가용(미압축) PyTorch 가중치 저장 경로
DST_COMP_PATH	os.path.join(WORK_DIR, "comp.pt")	압축(컴파일)된 PyTorch 가중치 저장 경로
DST_TEMP_PATH	"" (빈 문자열)	임시 결과물이 필요할 때 채워지는 경로 (현재 미사용)
ONNX_EVAL_PATH	os.path.join(WORK_DIR, "eval.onnx")	EVAL ONNX 모델 저장 경로
ONNX_COMP_PATH	os.path.join(WORK_DIR, "comp.onnx")	COMP ONNX 모델 저장 경로
LAYER_EVAL_PATH	os.path.join(WORK_DIR, "eval.tachy")	Layer-level EVAL TACHY 파일 경로
LAYER_COMP_PATH	os.path.join(WORK_DIR, "comp.tachy")	Layer-level COMP TACHY 파일 경로
L_DEFAULT_PAD_MODE	--default_pad_mode=dynamic	레이어 컴파일 시 패딩 모드 설정 플래그
BLOCK_COMP_PATH	os.path.join(WORK_DIR, "block.tachy")	Block-level 압축 TACHY 파일 경로
B_DEFAULT_PAD_MODE	--default_pad_mode=dynamic	블록 컴파일 시 패딩 모드 설정 플래그
DEFAULT_PAD_ORDER	--default_pad_order=up_left	패딩을 적용할 기본 방향 지정
LOGIT_ORDER	--logit_order 5 3 0 4 2 1	출력 로짓(클래스) 순서 재정렬 옵션

□ 파일 실행

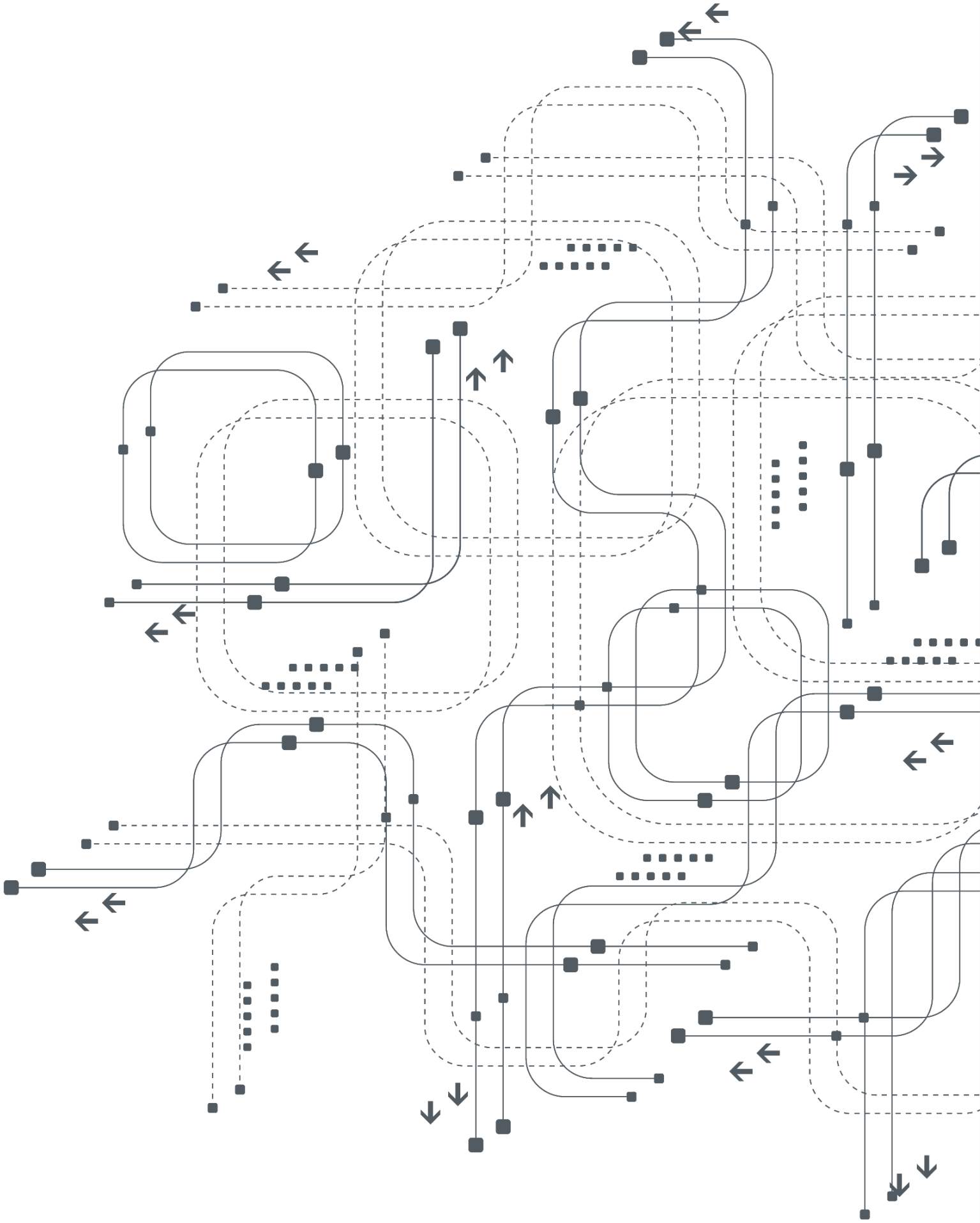
```
./compile
```

□ 결과 확인

```
ls 20250429_152340/*
```

□ 예시 출력

```
model_224x224x3_inv-t.tachyrt inst.bin param.bin inst.json
```



고객 지원 (Customer Support)

- **문의처**: partner@deeper-i.ai
- **웹사이트**: https://www.deeper-i.ai