

소프트웨어 사용 가이드

Neuro-R 기본 개념 및 활용 방법 설명

NEUROCLE

Deep Learning Vision Software

딥러닝 모델 학습용 소프트웨어

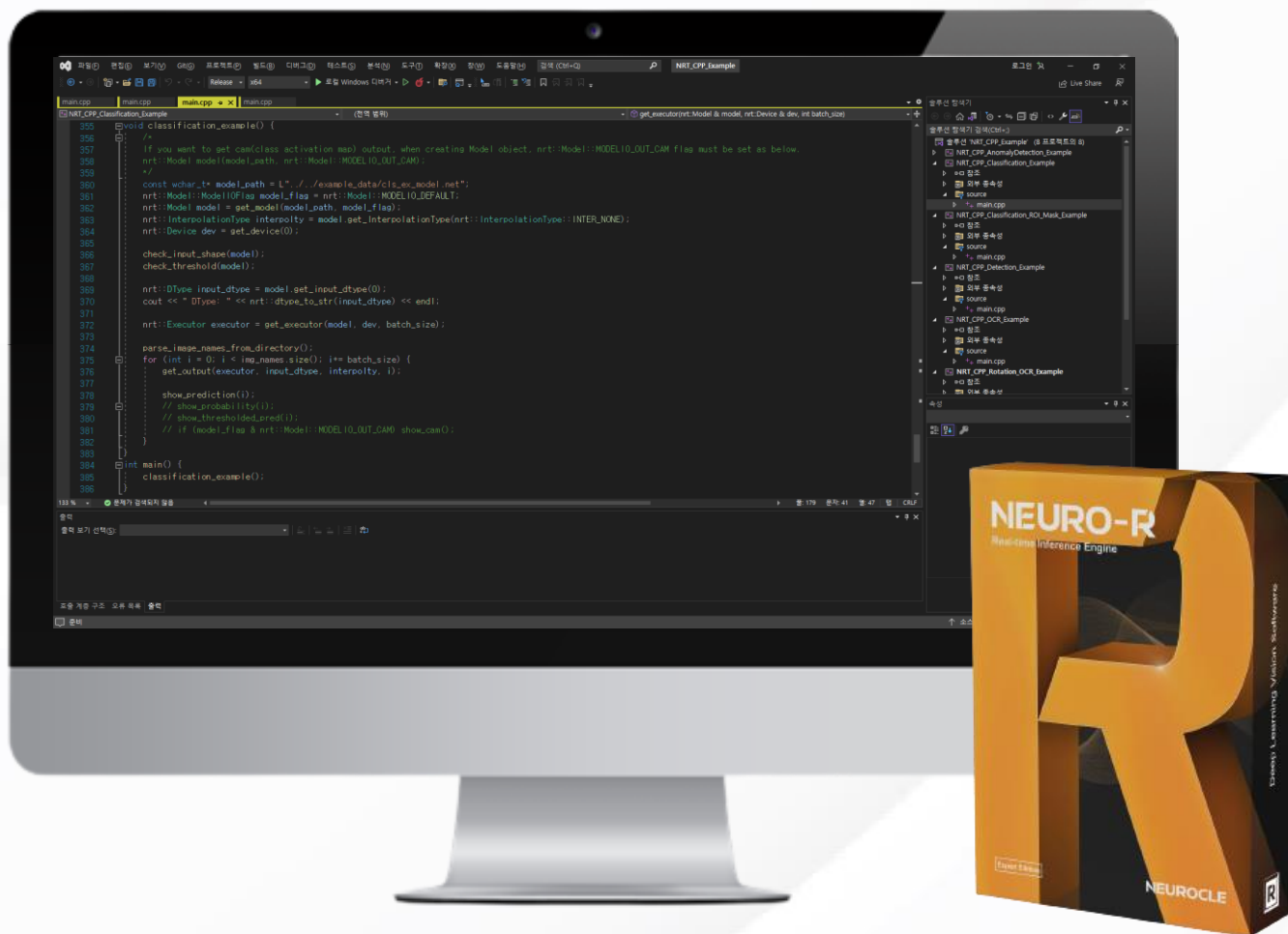


NEURO-T

GUI 기반의 No-Code 소프트웨어

- 누구나 고성능의 딥러닝 모델을 손쉽게 생성할 수 있는 오토 딥러닝 알고리즘 제공
- 이미지 편집부터 모델 학습까지 모든 것이 가능한 올인원 플랫폼

런타임 라이브러리



NEURO-R

Neuro-T에서 생성한 모델을 활용하기 위해 필요한
런타임 API

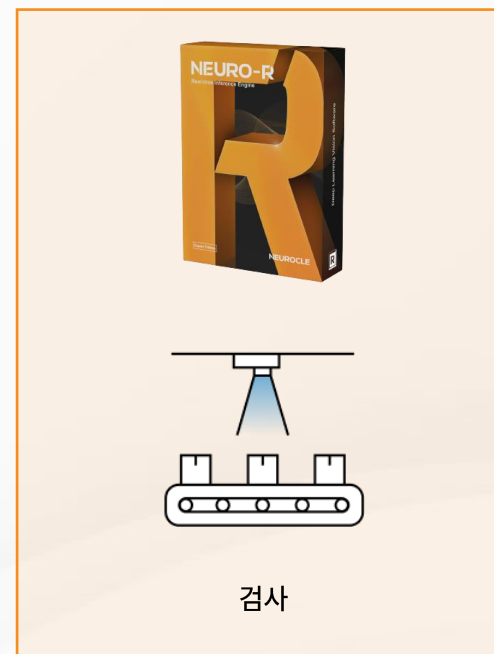
- 폭넓은 프로그래밍 언어 지원 (C++, C#, Python)
- 임베디드 장비부터 PC까지 다양한 환경을 지원
- 목표 검사 속도 달성이 가능한 빠른 검사 속도

비전검사 워크플로우

NEURO-T



NEURO-R



Neuro-T에서 생성된 모델을 효과적으로 활용하기 위해서는 Neuro-R이 반드시 필요합니다.

목차

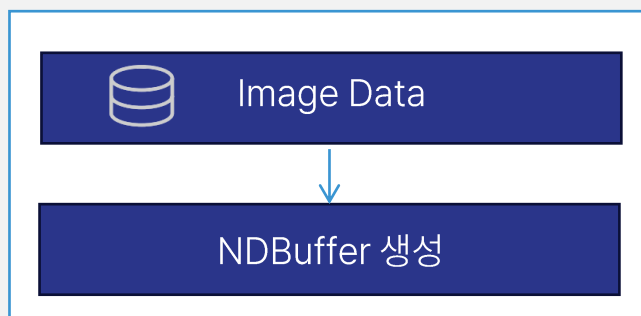
1. 개념 설명
2. 적용 및 활용
3. 샘플 코드 분석

01. 개념 설명

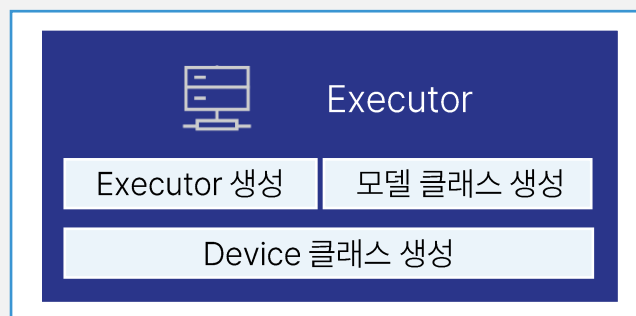
Neuro-R 변경사항

~v3.2

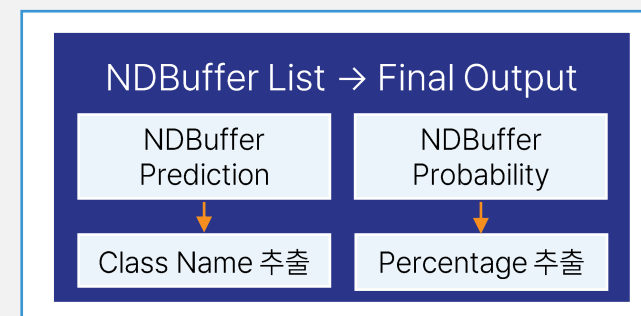
Input



Processing

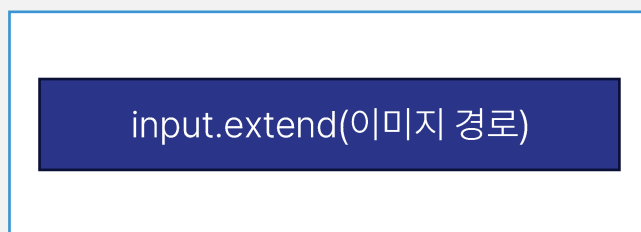


Output

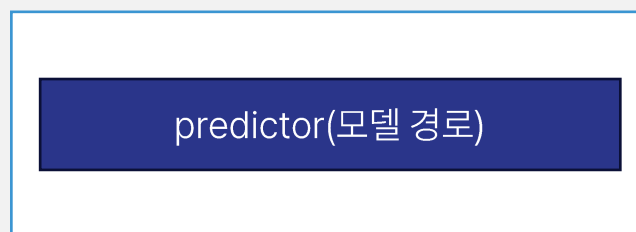


v4.0

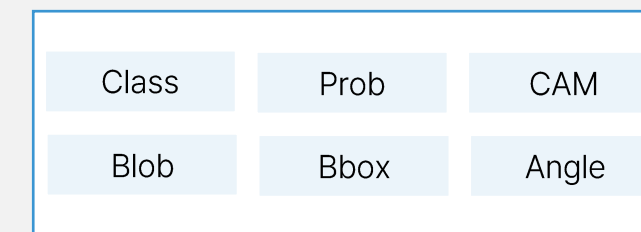
Input



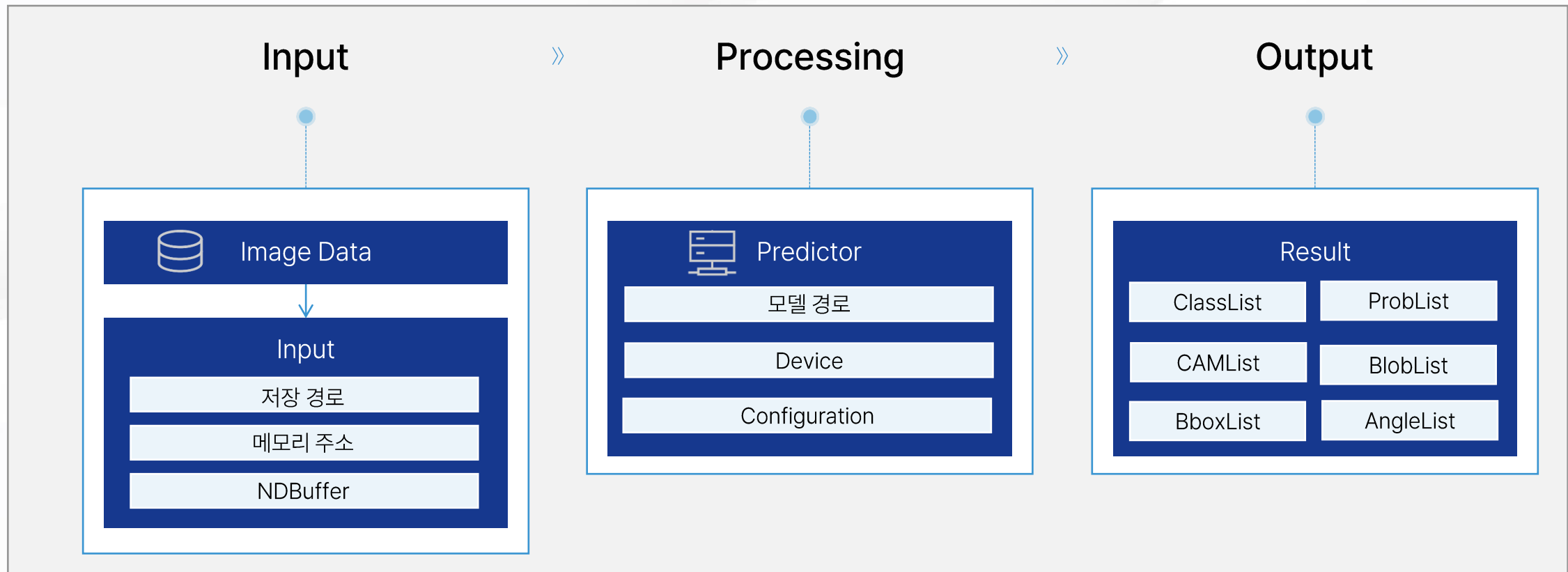
Predictor



Result



Neuro-R 워크플로우



Input



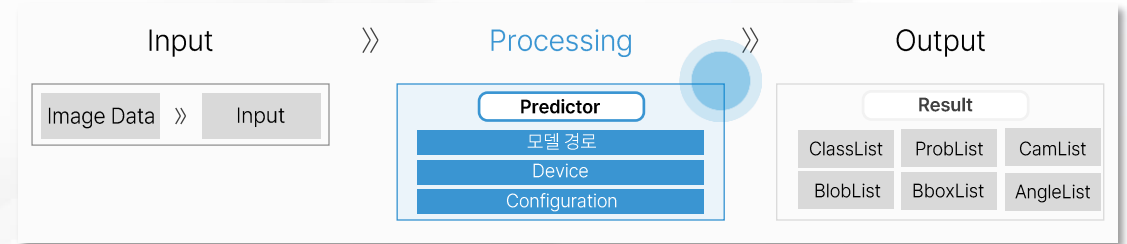
데이터 추가 방법에 따른 API 사용법

1. 저장 경로로 추가하는 법 **API →** `input.extend (이미지 저장 경로)`
2. 데이터 메모리 주소로 추가하는 법 **API →** `input.extend(메모리주소, nrt::Shape(h,w,c), nrt::DTYPE_UINT8)`
3. NDBuffer*로 변환하여 추가하는 법 **API →** `Input.extend(ndbuffer)`

*NDBuffer란?

OpenCV의 Mat, 혹은 NumPy의 ndarray와 같이 데이터의 모양 및 유형을 포함하고 있는 이미지 매트릭스 제어 클래스입니다.

Processing



Predictor 입력 매개변수

딥러닝 모델

1. 모델 경로

2. 모델 I/O Flag

- MODELIO_DEFAULT
- MODELIO_OUT_PRED
- MODELIO_OUT_PROB
- MODELIO_OUT_CAM

+

Device Flag

- ✓ GPU
- ✓ CPU

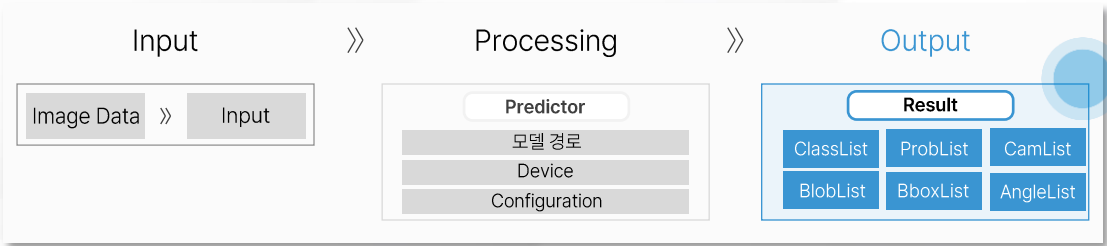
+

설정

- ✓ 배치 사이즈
- ⚙ FP 16 설정
- ✓ Threshold 설정

Result

추론 결과를 리스트 형태로 출력합니다.






모델 종류	Classification & Anomaly Classification	Segmentation & Anomaly Segmentation	Object Detection & OCR	Rotation
Result	<div>ClassList</div> <div>ProbList (Anomaly Score)</div> <div>CAMList</div>	<div>BlobList</div> <div>Blob class</div> <div>Blob 좌표</div> <div>Blob Probability (Anomaly score)</div>	<div>BoxList</div> <div>Box class</div> <div>Box 좌표</div> <div>Angle List (OCR)</div> <div>ProbList</div>	<div>Angle List</div>


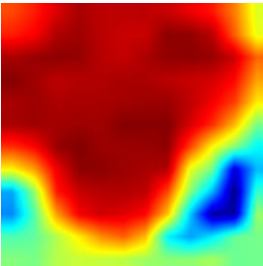
Outputs

Classification과 Anomaly Classification의 결과입니다.

Class & Probability(Anomaly Score)

Image			
Class (Prediction)	0 (Good)	0 (Good)	1 (Bad)
Probability (Anomaly Score)	0.9887	0.8830	0.0001
	0.0012	0.1170	0.9999

CAM

Input Image	
CAM Image	

Outputs

Segmentation과 Anomaly Segmentation의 결과입니다.

모델 추론 단위 및 출력 방식

픽셀 단위로 Class 출력 후 같은 클래스를 Blob으로 그룹화

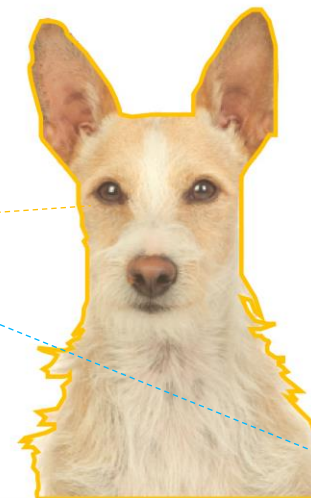
0	0	0	0	1	1	1	1
0	0	0	0	0	1	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0
2	2	0	0	0	0	0	0
2	2	0	0	0	0	0	0

☐ Background ☒ Cat ☒ Dog

Blob 정보 출력

Blob의 Contour, Class, 좌표, 면적, Probability 등 출력 가능

Class : Cat, x : 80, y : 240, Area : 7590



Class : Cat, x : 150, y : 180, Area : 6320



모델 추론 단위 및 출력 방식

Bounding Box에 담긴 정보 값들

BCYXHW

B : Batch_idx

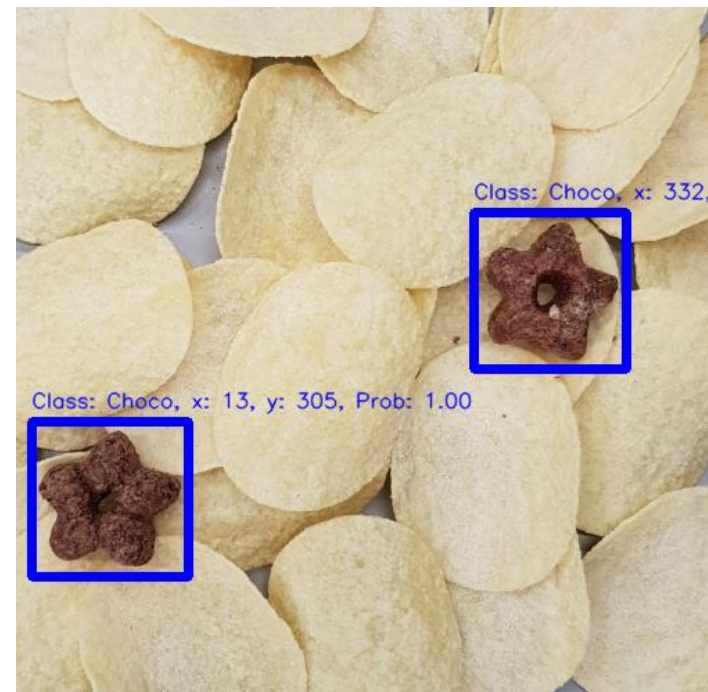
Y, X : Center Y, X

C : Class

H, W : Height, Width

B	C	Y	X	H	W
..

Bbox 출력



Outputs

Rotation의 결과입니다.

Rotation 전



89°

모델 예측 결과
89도 회전 적용

Rotation 후



02. 적용 및 활용

Neuro-R 활용 예시

1

Flowchart

2

Threshold

3

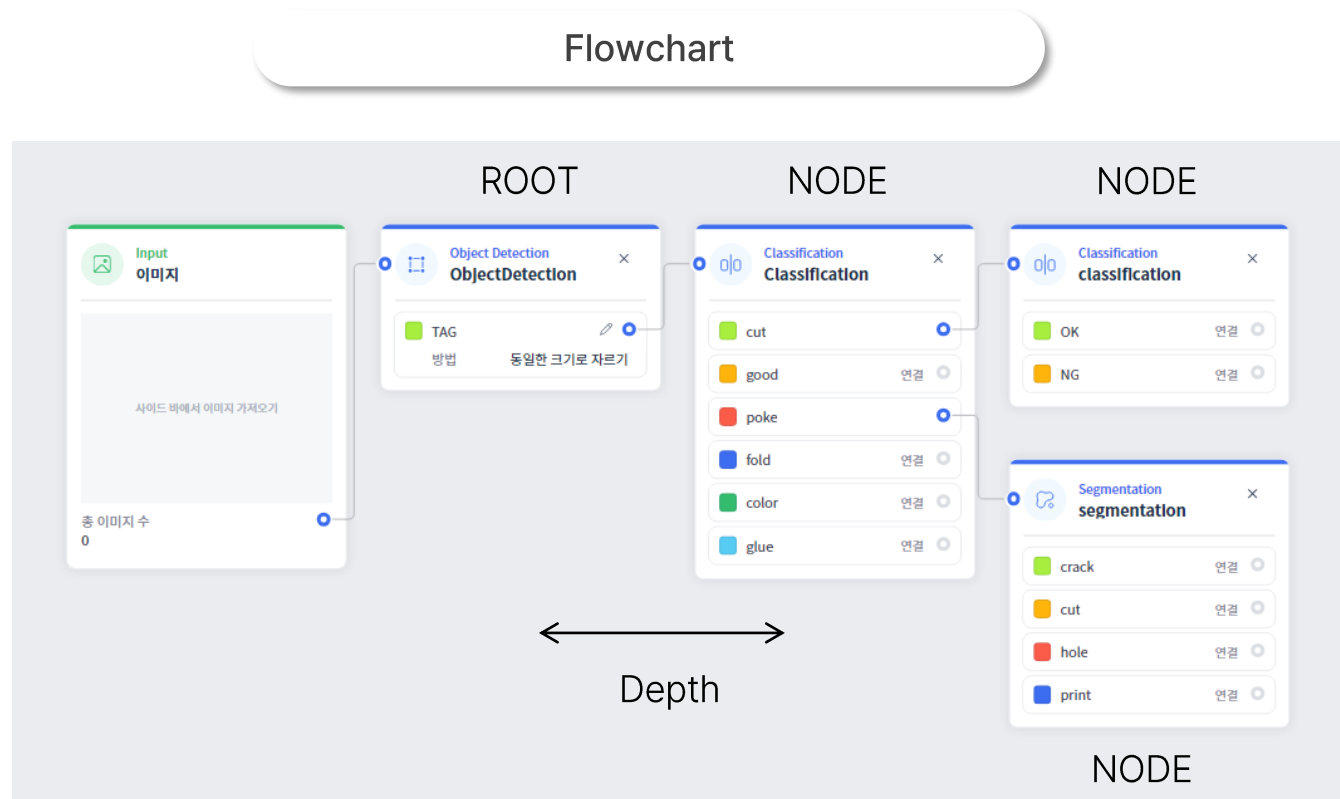
ROI / Mask

4

Patch Mode

Flowchart

- Inference Center에서 Flowchart 생성
- Predictor 클래스 대신 Flowchart 클래스 사용



Threshold

- Threshold_flag = true 적용 시 자동 적용
- Unknown 방식과 차선 클래스 출력 방식 → Neuro-T 에서 생성
- Neuro-R 4.0에서 Threshold를 직접 만들 경우 → 출력부에서 조건문 작성

Probability Threshold

```
float prob = res.probs.get(i, cla.idx);  
float threshold = 0.8;  
  
if (prob >= threshold)  
{  
    // 코드작성  
}
```

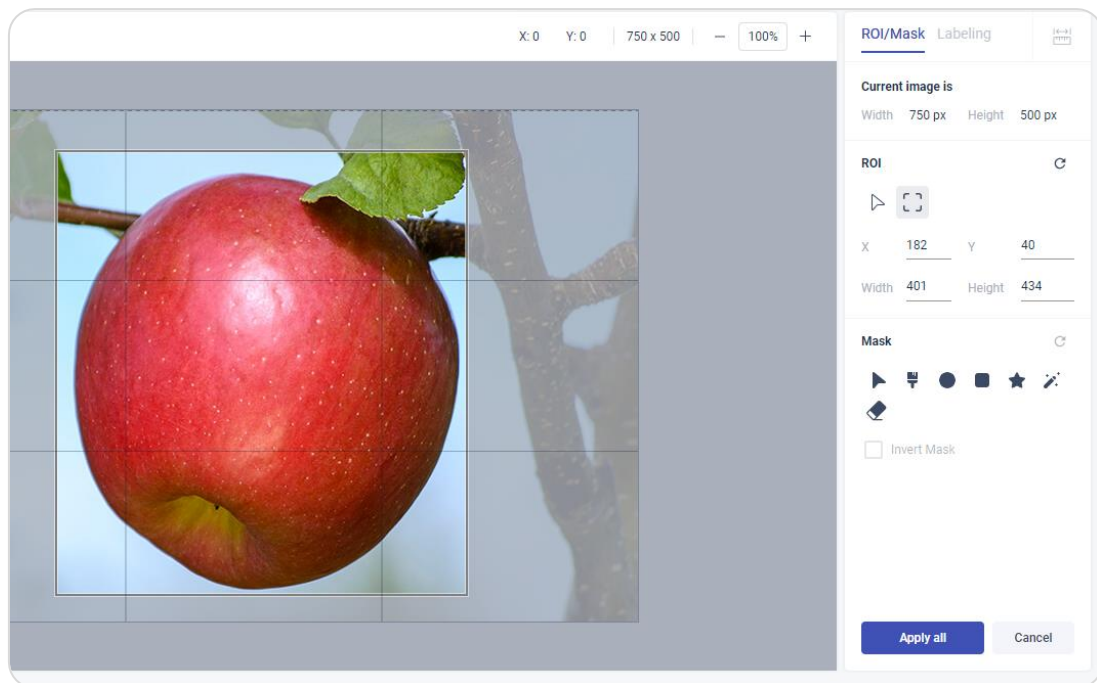
Size Threshold

```
nrt::Blob blob = res.blobs.get(i);  
  
if (blob.rect.width >= 100 && blob.rect.height >= 100)  
{  
    // 코드 작성  
}
```

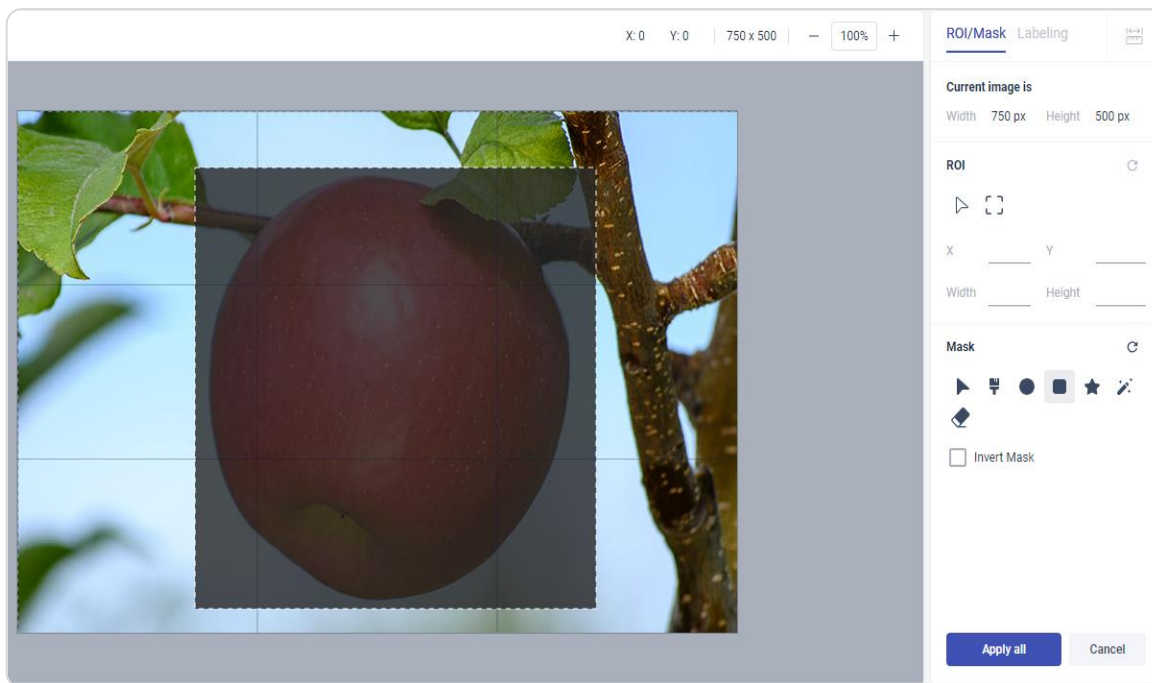
ROI/ Mask

- ROI/ Mask는 Flag 없이 자동 적용
- ROI는 원본 이미지 크기에 대한 ROI 좌표 정보가 저장되기 때문에, 추론 시에도 이미지 크기와 객체 위치를 고려해야 함.

ROI



MASK

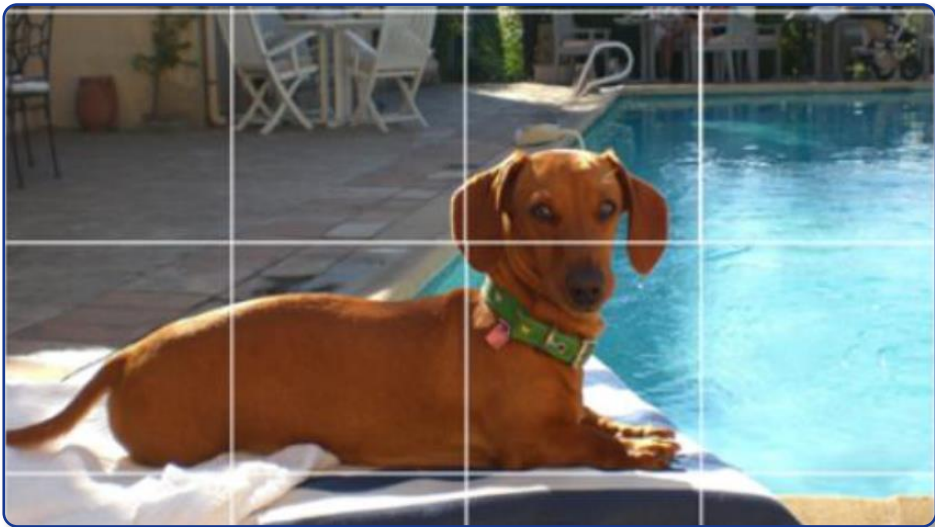


Patch Mode (for Segmentation)

- 다른 모델 유형과 다르게, Segmentation 모델은 이미지를 분할하여 학습에 사용
- Patch Mode에서 Predictor가 이미지 한 장을 한 번에 인퍼런스 하기 위해서는 패치로 분할된 이미지 수를 한 번에 추론해야 함.

EX) 1024X1024 이미지를 512X512 크기의 패치로 자를 경우 9장의 패치가 생성되며 이 9장을 한 번에 추론한다.

01. 이미지 분할 크기 선택





02. 분할된 이미지 Inference



03. 샘플 코드 분석

1. 경로 설정(사용자 환경에 따라 변경 필요)

이미지 데이터가 들어있는 폴더 경로 설정 및
Neuro-T에서 "모델 내보내기"를 통해 추출된 모델(.net) 파일 경로 설정

 cls_ex_img	2024-01-22 오후 5:08	파일 폴더	
 cls_ex_model.net	2023-09-14 오후 1:35	NET 파일	5,539KB



경로 적용

```
def main():  
    image_dir = "../example_data/cls_ex_img/"  
    model_path = "../example_data/cls_ex_model.net"  
    predictor_path = "../example_data/cls_ex_predictor.nrpd"  
    batch_size = 3  
    fp16_flag = False  
    threshold_flag = False
```

Predictor_path의 경우 파일 생성 위치로 작성해주시면 됩니다.

2. Predictor 생성 및 불러오기

```
# 1 step. Create the Predictor using '.net' or '.nrpd' file.  
# CPU's device_idx = -1, GPU's device_idx = [0, num of device)  
device_idx = 0  
  
# If you have used the Predictor in the same Device and Model environment and saved the Predictor as a file(.nrpd)  
# you can reduce the optimization time by loading that file for use.  
if device_idx >= 0 and os.path.isfile(predictor_path):  
    print("Load the Predictor using the previously optimized Predictor file.")  
    predictor = nrt.Predictor(device_idx, predictor_path)  
else:  
    print(  
        "Optimizing the Predictor for the Model and Device... It may take a few minutes."  
    )  
    # The default device setting for the Predictor is CPU(-1).  
    # If you want more diverse options, you can use it as shown in the comment below.  
    predictor = nrt.Predictor(model_path)  
    predictor = nrt.Predictor(  
        model_path,  
        nrt.Model.MODELIO_OUT_CAM,  
        device_idx,  
        batch_size,  
        fp16_flag,  
        threshold_flag,  
    )  
  
    # Save the predictor information optimized for this device,  
    # so if the same model is used in the same device environment later, the optimized predictor can be reused.  
    if device_idx >= 0:  
        print("Save the information of the optimized Predictor to a file(.nrpd)")  
        status = predictor.save_predictor(predictor_path)  
        if status != nrt.STATUS_SUCCESS:  
            raise Exception(  
                "Predictor initialization failed. : " + nrt.get_last_error_msg()  
            )  
    if predictor.get_status() != nrt.STATUS_SUCCESS:  
        raise Exception(  
            "Predictor initialization failed. : " + nrt.get_last_error_msg()  
        )  
    print("Predictor initialization complete...\n")
```

3. 이미지를 모델 내 Input 및 추론 시작

```
# 2 Step, Get image to predict
inputs = nrt.Input()
img_files = glob.glob(f"{image_dir}/*.jpg")
img_files.extend(glob.glob(f"{image_dir}/*.png"))
batch_size = predictor.get_batch_size()
cur_batch = 0
is_last = False
for i, img in enumerate(img_files):
    is_last = i == (len(img_files) - 1)
    status = inputs.extend(img) → 이미지 Input
    inf_files.append(img)
    if status != nrt.STATUS_SUCCESS:
        raise Exception("Extend the input failed. : " + nrt.get_last_error_msg())
    cur_batch += 1
# 3 Step, If the inputs have reached the batch size, start the prediction.
if (cur_batch == batch_size) or is_last:
    print("Acquired input for the batch size. Now starting the prediction.")
    start = time.time()
    results = predictor.predict(inputs) → 모델 추론 및 결과 데이터를 results 변수에 저장
    end = time.time()
    if results.get_status() != nrt.STATUS_SUCCESS:
        raise Exception("Predict failed. : " + nrt.get_last_error_msg())
    print(
        f"duration per batch ({cur_batch} images) : {round((end - start) * 1000, 2)} ms"
    )
    print_cla_output(results, predictor) → 결과 데이터 시각화 함수
    inputs.clear()
    inf_files.clear()
    cur_batch = 0
```

4. 결과 출력 함수

```
def print_cla_output(results: nrt.Result, predictor: nrt.Predictor):
    print(results.classes)
    for i in range(results.classes.get_count()):
        cla = results.classes.get(i)
        prob = results.probs.get(i, cla.idx) → 확률 값(Probability)
        print(f"File name : {inf_files[i]} ", end="")
        print(f"- Class : {predictor.get_class_name(cla.idx)}, Prob : {prob}")
        → 클래스 이름(class_name)
    if not results.cams.empty():
        originImg = cv2.imread(inf_files[i])
        cam = results.cams.get(i)
        mat_cam = cam.cam_to_numpy()
        mat_cam = mat_cam.reshape([cam.get_height(), cam.get_width(), 3])
        mat_cam = cv2.applyColorMap(mat_cam, cv2.COLORMAP_JET)
        cv2.addWeighted(originImg, 0.4, mat_cam, 0.6, 0, mat_cam)
        cv2.imshow("cam", mat_cam) → 이미지 결과 시각화
        cv2.waitKey(0)
        cv2.destroyAllWindows()
```


Q & A

감사합니다.