인공지능 기술의 대중화(AI Democratization)를 위한

TANGO 커뮤니티 제1회 컨퍼런스

TANGO

신경망 배포 탑재 기술

성 명 이경희 소 속 한국전자통신연구원

주관 ETRI 한국전자통신연구원 (TANGO) 주최 과학기술정보통신부 IITP 정보통신기획평가원 후원 Lablup weda tesla System 한국인공지능협회 SNUH 서울대학교병원 고려대학교 KOREA UNIVERSITY 홍익대학교 HONGIK UNIVERSITY CAU 중앙대학교 Georgia Tech

TANGO

목차

1. 배포탑재 기술의 개요 – 비전

다양한 타겟 환경지원

- **OS의 다양성 지원**
  - Linux
  - Windows
- **HW의 다양성 지원**
  - x86(windows, Linux), RK3399Pro/OdroidN2/M1, Google Cloud, etc
  - CUDA, NPU, Mali 등 다양한 가속환경 지원
- **추론엔진의 다양성 지원**
  - PyTorch
  - ACL(PyArmNN), RKNN

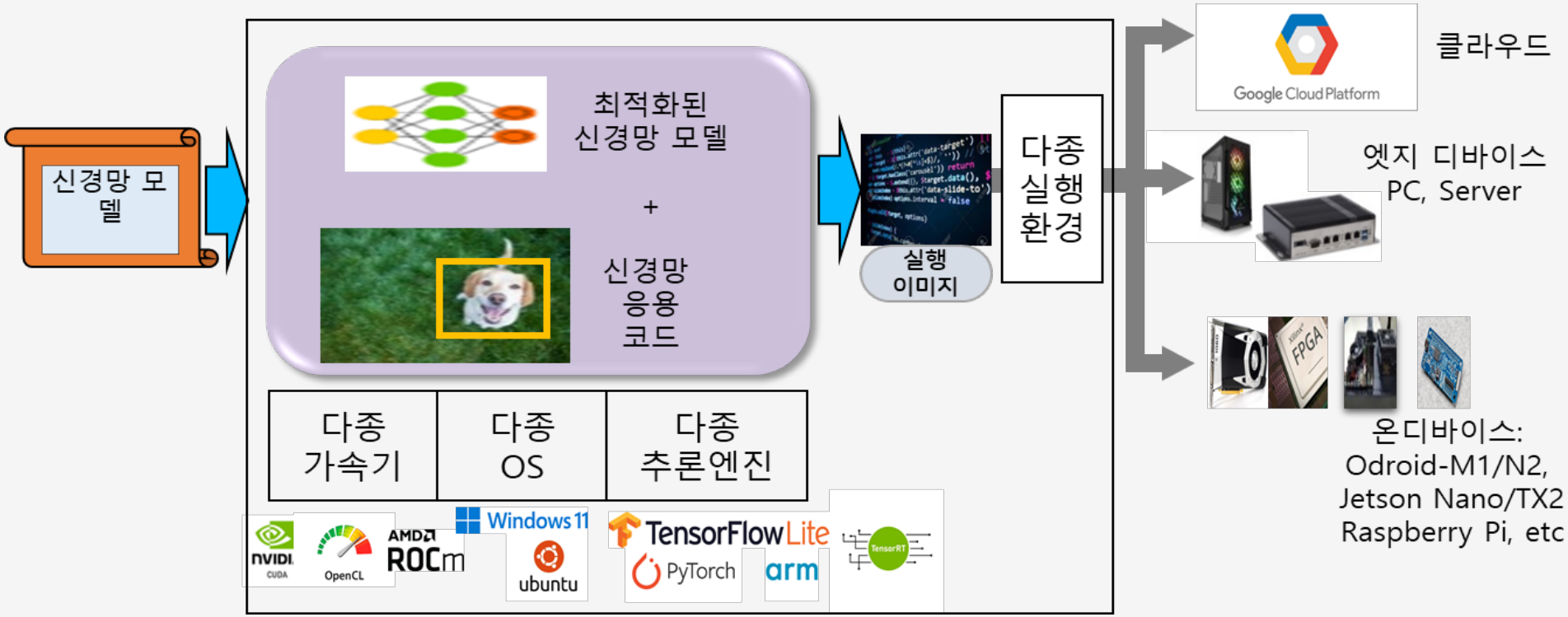
1. 배포탑재 기술의 개요 – 비전

응용 개발 및 실행의 편이성 지원

- **배포탑재의 편이성 지원**
  - docker
  - bin. file copy/transfer
- **서비스 개발의 편리성 지원**
  - 신경망 모델기반 응용 개발의 용이성(응용 템플릿)
  - Python 지원
- **DevOps 지원 고려**
  - 배포 프로세스 마이크로 서비스화, 개발 history 관리 등등
  - 문제발견 수정/학습데이타변경/요구사항 변경 후 빠른 deploy

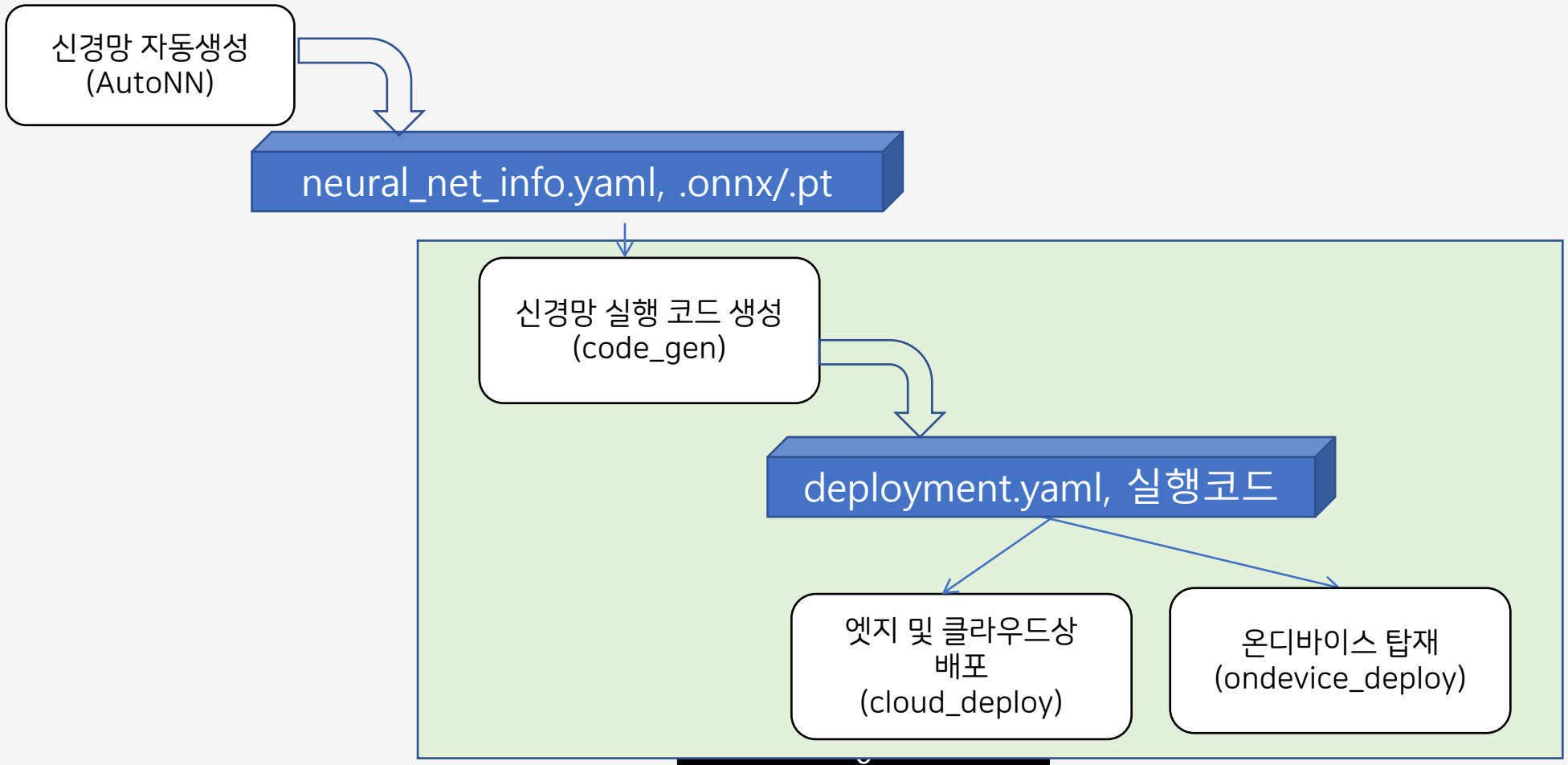
1. 배포탑재 기술의 개요 – 비전

클라우드, 엣지, 온디바이스 및 다종 가속 환경 통합 지원



1. 배포탑재 기술의 개요 – 시나리오

동작 시나리오



2. 구현 현황

소스 트리 구조

- /tango/
  - deployment/
    - code_gen
    - target_deploy
      - cloud_deploy/
      - ondevice_deploy/

폴더명	기능
code_gen	신경망 실행에 필수적인 코드 생성
cloud_deploy	클라우드, 엣지 클라우드 대상 신경망 실행 이미 생성 및 배포
ondevice_deploy	온디바이스용 이미지 복사/다운로드

2. 구현 현황 – 입력

neural_net_info.yaml (예)

```
# NN Model
class_file: 'bestmodel.py' # for pytorch model
class_name: 'TheBestmodelClass()' # for pytorch model
weight_file: [bestmodel.pt, yolo5s.onnx]

# Label
nc: 80 # number of classes
label_info_file: labelmap.yaml

# Input
input_tensor_shape: [1, 3, 640, 640]
input_data_type: fp32 # fp32, fp16, int8, etc
# anchors: 3
anchors:
  - [10,13, 16,30, 33,23]  # P3
  - [30,61, 62,45, 59,119]  # P4
  - [116,90, 156,198, 373,326]  # P5
```

2. 구현 현황 - 코드 생성

code_gen

- 신경망 생성 모델의 실행을 위한 전처리/후처리 코드 생성
  - 전처리: 이미지 resize/crop, 평균치 조정, 신경망 입력용 텐서 생성(NCHW)
  - 후처리: 신경망 모델의 출력 해석 (인식 객체명, 확률값)

- 신경망 실행을 위한 입출력 코드 생성
  - 입력: 파일/동영상/카메라 입력 코드 생성
  - 출력: 동영상, 화면, 텍스트 등으로 결과 출력 코드 생성

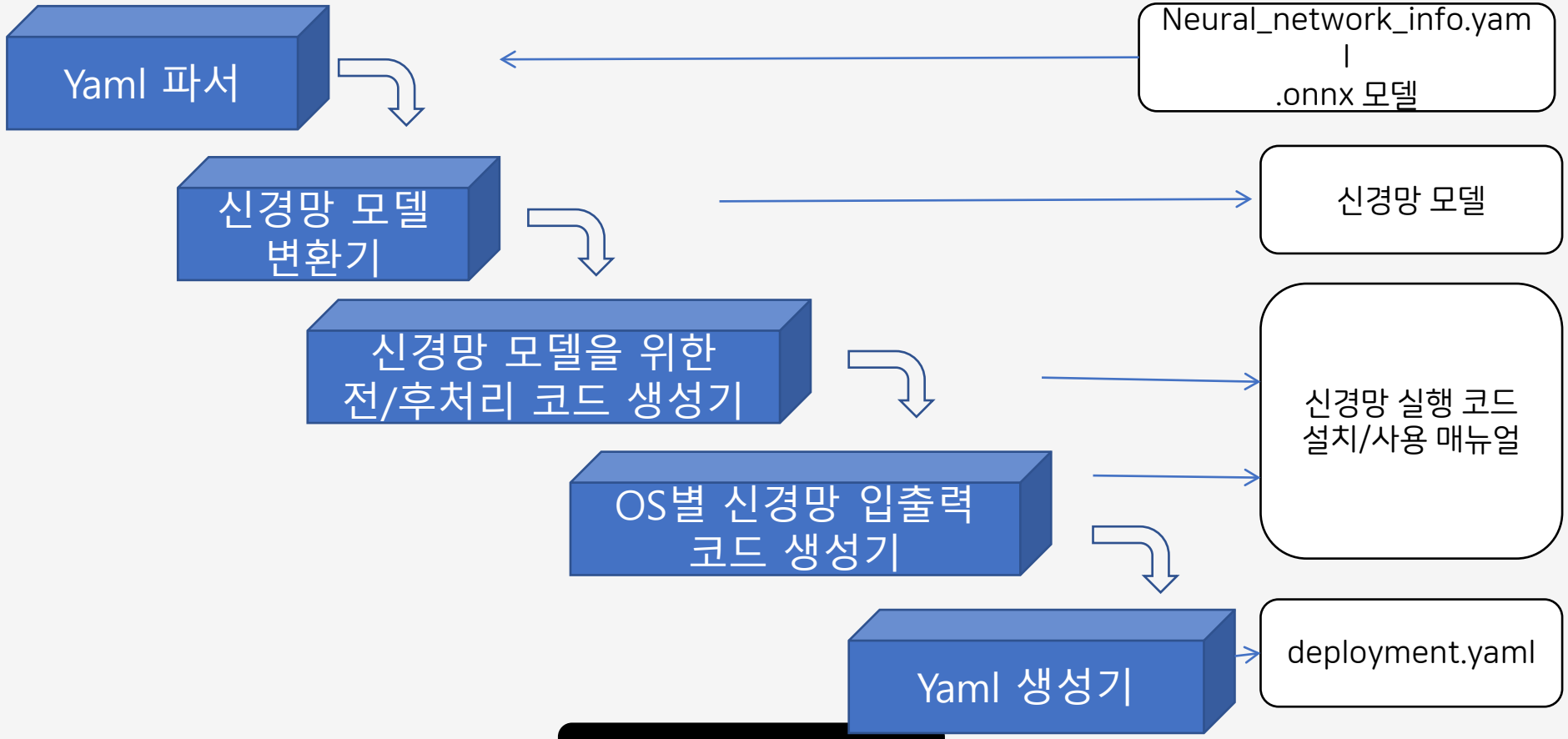
- 배포탑재 및 실행을 위한 정보 파일 해석 및 생성
  - AutoNN과 Project manager에서 생성한 yaml화일 해석
  - 클라우드/엣지클라우드/온디바이스상 신경망 배포 실행을 위한 yaml 파일 생성

2. 구현 현황 – 코드 생성

code_gen

2. 구현 현황 – 코드 생성

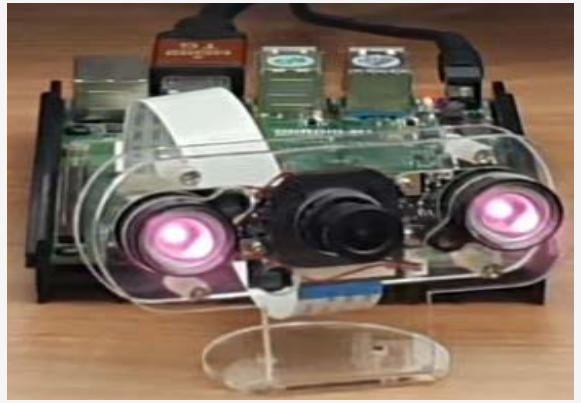
code_gen



2. 구현 현황 - 코드 생성

ArmNN 생성 코드의 예

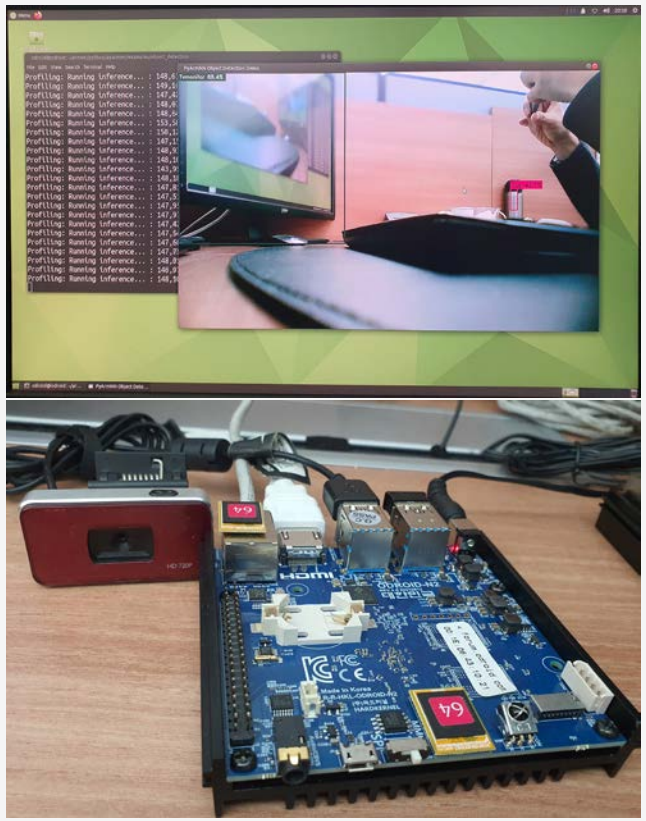
```
def main(file_name):
    enable_profile = def_profiling_enabled == "true"
    action_profiler = Profiling(enable_profile)
    overall_profiler = Profiling(enable_profile)
    overall_profiler.profiling_start()
    action_profiler.profiling_start()
    exec_input_args = (def_model_file_path,
                       def_preferred_backends)
    executor = ArmnnNetworkExecutor(*exec_input_args)
    action_profiler.profiling_stop_and_print_us("Executor initialization")
    action_profiler.profiling_start()
    video, frame_count = init_video_file_capture(file_name)
    process_output = yolo_processing
    resize_factor = yolo_resize_factor(video, executor.get_shape())
    action_profiler.profiling_stop_and_print_us("Video initialization")
.............
```



2. 구현 현황 - 코드 생성

RKNN 생성 코드의 예

```
if __name__ == '__main__':
    if len(sys.argv) < 2:
        print('%s%s' % ('input_file = ', input_file))
    else:
        input_file = sys.argv[1]
    rknn_lite = RKNNLite()    # load RKNN model
    print('--> Load RKNN model')
    ret =      rknn_lite.load_rknn(rknn_model_file)
    if ret != 0:
        print('Load RKNN model failed')
        exit(ret)
    print('done')
.....
```



2. 구현 현황 – 배포탑재

deployment.yaml의 예

```
build:
    accelerator: cpu
    architecture: linux/amd64
    components:
        custom_packages:
            atp:
                - vim
                - python3.9
            pypi: []
    os: ubuntu
    target_name: yolov3:latest
    workdir: /yolov3
deploy:
    entrypoint:
        - python3
        - deploy_server.py
    network:
        service_container_port: 5051
        service_host_ip: 0.0.0.0
        service_host_port: 8887
```

2. 구현 현황 – 배포탑재

ondevice_deploy

- **배포탑재 및 실행을 위한 정보 파일 해석**
  - code_gen에서 전달받은 deployment.yaml 파일 해석

- **신경망 실행 코드 전달**
  - 신경망 실행 코드 압축
  - 설치 및 사용 매뉴얼

2. 구현 현황 – 배포탑재

ondevice_deploy

3. 개발 관련 이슈

지원 범위

- 코드 자동 생성의 범위
  - 사용자 요구 명세의 범위(GUI, 응용의 기능 명세)
  - 지원 라이브러리 (그래픽 라이브러리의 다양성)
- 온디바이스 지원 범위
  - 온디바이스를 위한 cross compile 환경 지원 여부
  - 온비바이스상 탑재 지원 방법
- 외부 IDE와의 연동 지원

4. 향후 계획

2023년 릴리즈 계획

- 배포탑재 모듈의 프레임워크화 강화
  - 추론엔진/타겟환경별 코드 생성/배포/탑재 모듈의 등록/삭제/관리를 지원하는 배포탑재 프레임워크 강화
- 지원 타겟 환경 확대
  - Jetson Nana, android
  - ondevice에 대한 cross compile 기능 제공
- 분산 실행 환경 지원
  - Kubernetes 등을 통한 분산 실행 환경 지원
- 코드 생성 범위 확대 및 외부 IDE 연동 지원
  - 사용자가 원하는 수준의 코드 자동생성 및 배포/탑재 방식 지원

감사합니다.

주관 ETRI 한국전자통신연구원 (TANGO) 주최 과학기술정보통신부 IITP 정보통신기획평가원 후원 lablup weda tesla System 한국인공지능협회 SNUH 서울대학교병원 고려대학교 KOREA UNIVERSITY 홍익대학교 HONGIK UNIVERSITY CAU 중앙대학교 Georgia Tech