

인공지능 기술의 대중화(AI Democratization)를 위한

TANGO 커뮤니티 제1회 컨퍼런스



신경망 요구사항 분석 및 시 각 프로그래밍 기술

성명 이윤규

소속 홍익대학교



주관 ETRI (TANGO) 주최 과학기술정보통신부 정보통신기획평가원 후원 LG W E O tesla System 한국인공지능협회 SNUH 서울대학교병원 고려대학교 KOREA UNIVERSITY 홍익대학교 HONGIK UNIVERSITY epw 중앙대학교 CHUNGANG UNIVERSITY Georgia Tech

목차

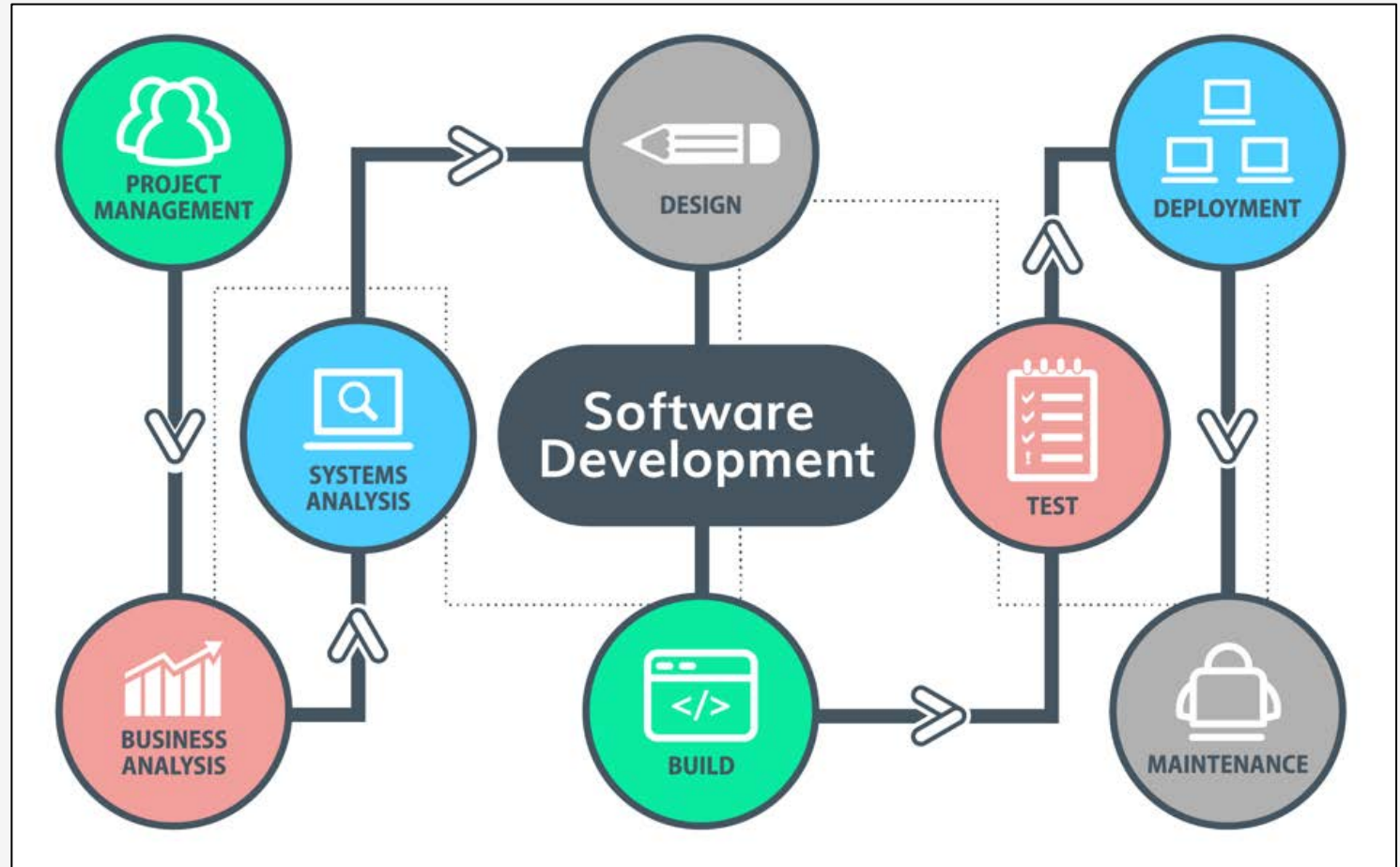
1. Overview
2. What to build
3. How to build
4. Concluding Remarks

인공지능 기술의 대중화(AI Democratization)를 위한
TANGO 커뮤니티 제1회 컨퍼런스

1. Overview

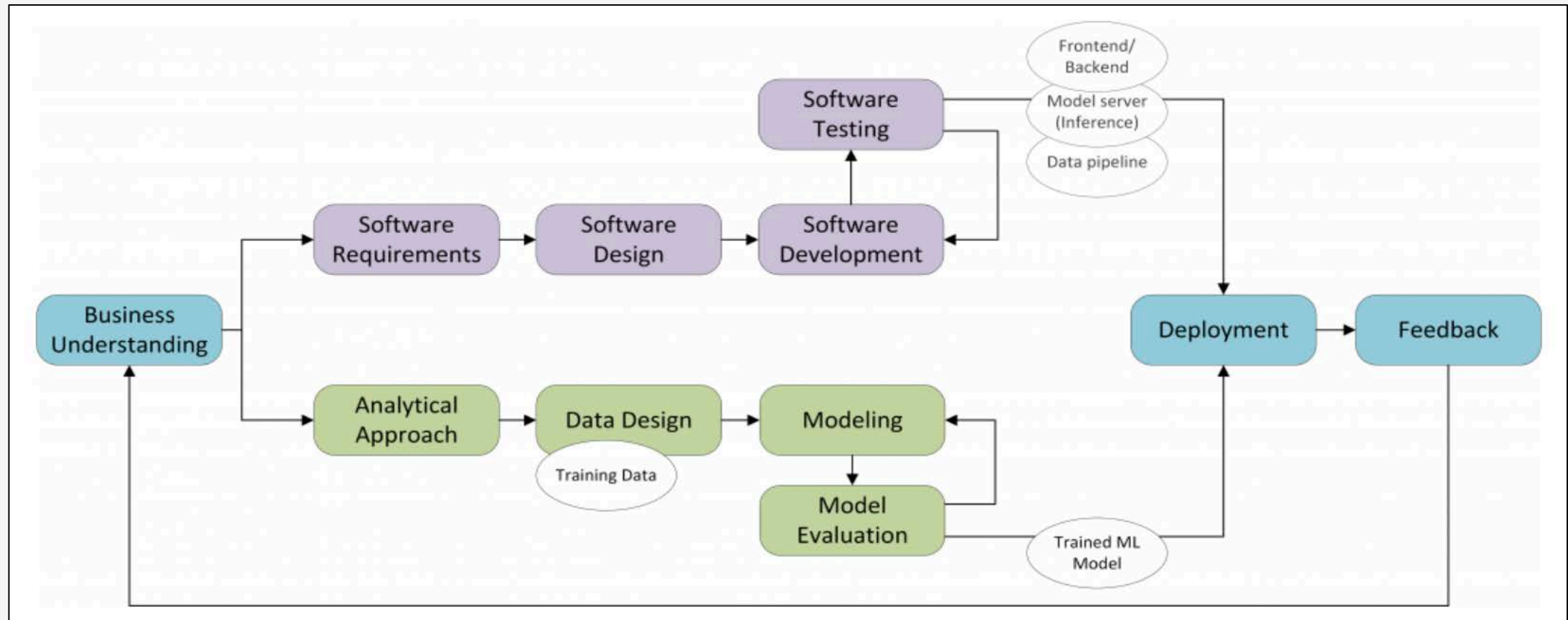
SW Development

- What to Build?
- How to Build?



1. Overview

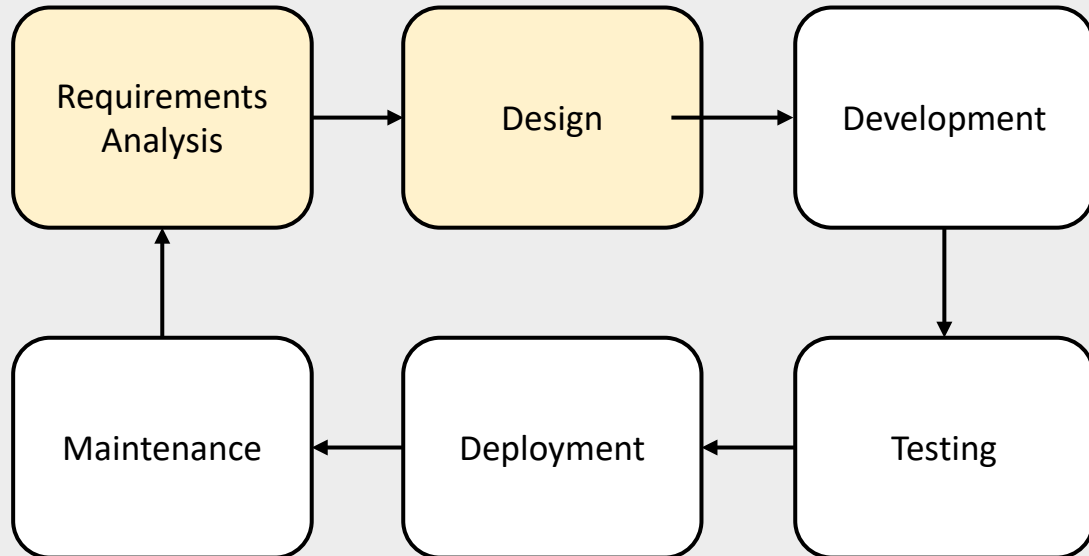
“Deep Learning” SW Development



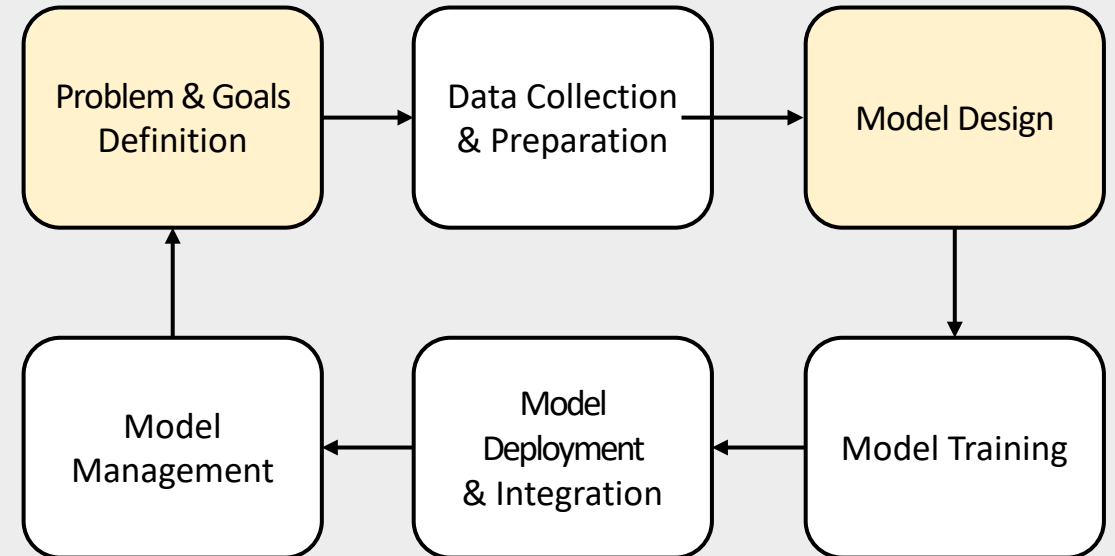
1. Overview

Traditional SW vs. Deep Learning SW

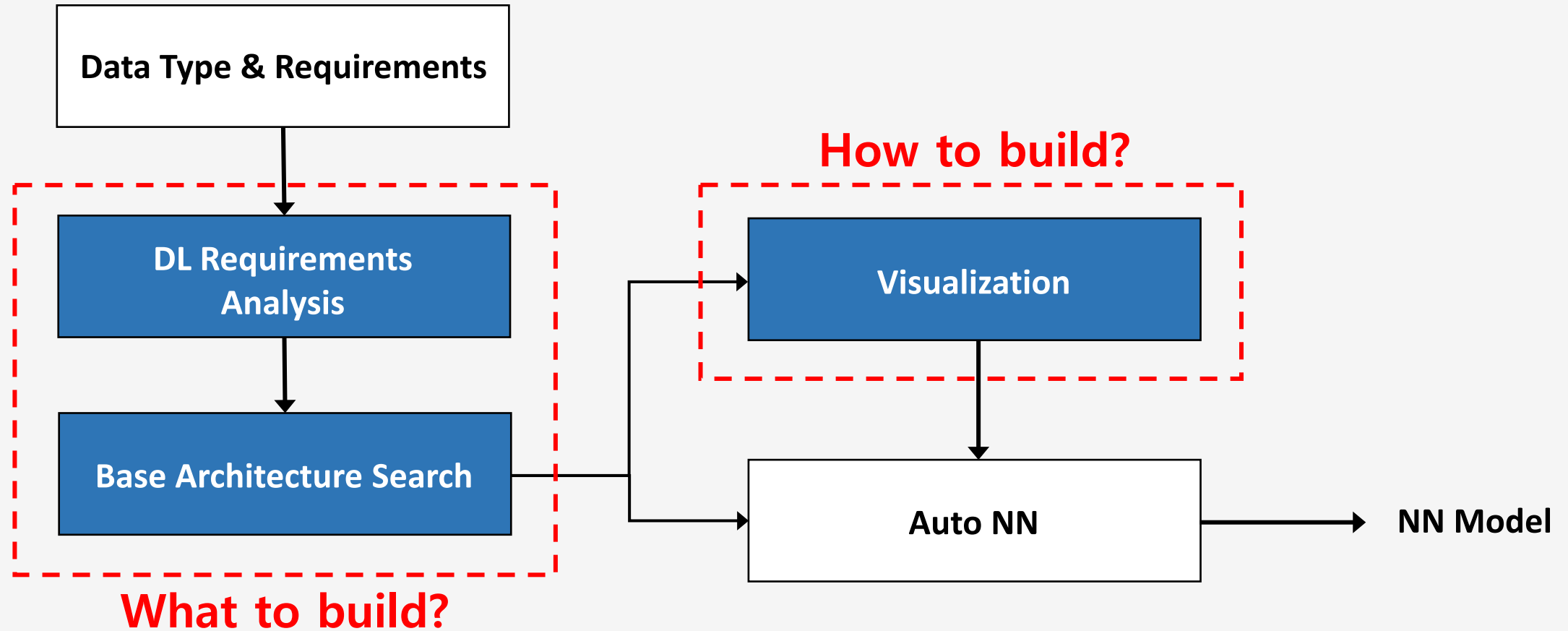
Traditional SW Development Process



Deep Learning SW Development Process



1. Overview

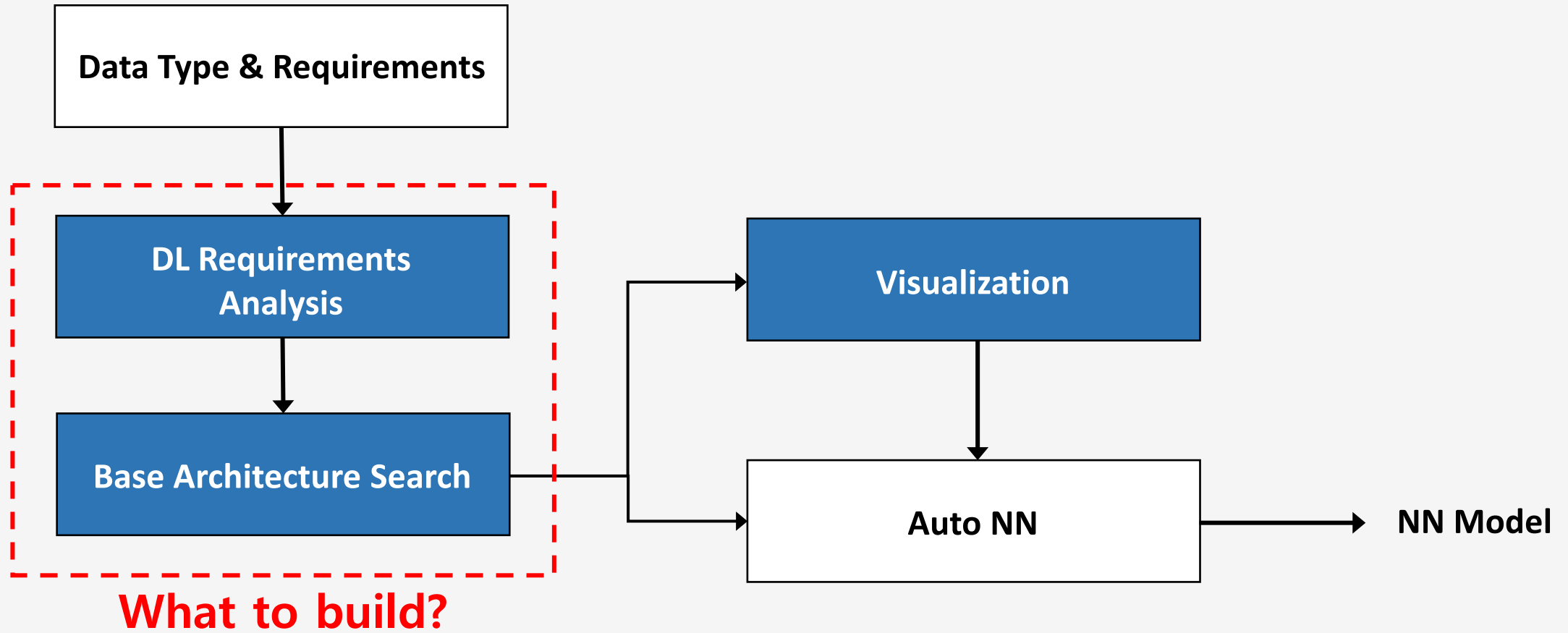


2. What to build

What to build

- 신경망 요구사항 분석 -

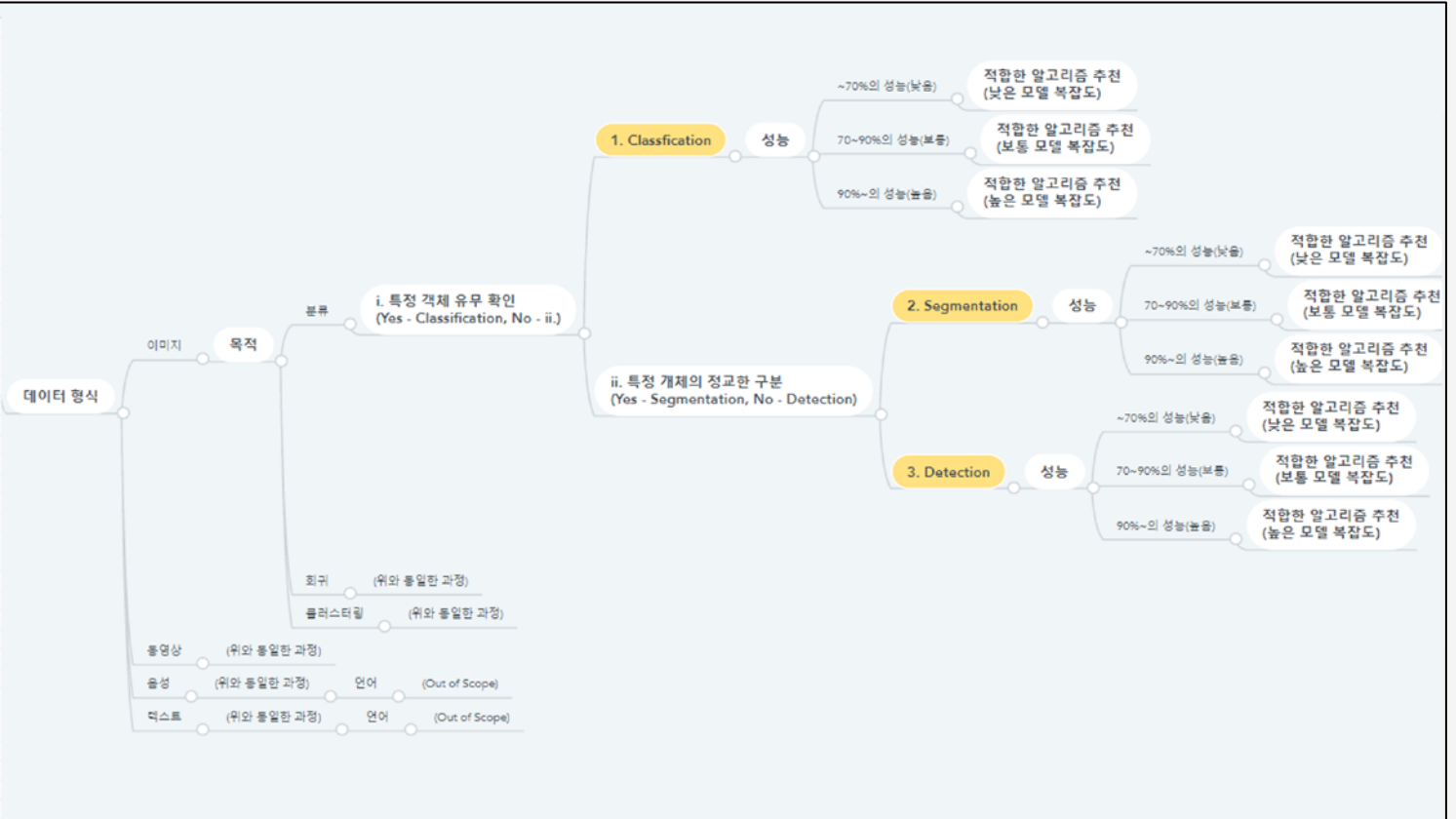
2. What to build



2. What to build

Rule-based taxonomy

산업군	활용 예시	인공지능 기술	비고	라벨
농업	정밀 농업	회귀, 분류, 클러스터링		이미지 - 1, 2, 3 동영상 - 1, 2, 3
	가축 모니터링	분류	각제 탐지	동영상 - 1, 2, 3
제조	실시간 공정 제어	회귀, 분류, 클러스터링	이상 탐지	동영상 - 1, 2, 3 이미지 - 1, 2, 3
	재고 관리 시스템	회귀		Out of Scope
에너지 (전기, 전력, 가스)	전력 수요 예측	회귀		이미지 - 1, 2, 3 동영상 - 1, 2, 3
	에너지 소비 이상 진단	회귀, 분류, 클러스터링	이상 탐지	이미지 - 1, 2, 3 동영상 - 1, 2, 3
건설	기상 예측	회귀		Out of Scope
	자재 및 활용 분석을 통한 실시간 견적 예측	회귀		이미지 - 1, 2, 3 동영상 - 1, 2, 3
	안전 사고 발생 예측	회귀		이미지 - 1, 2, 3 동영상 - 1, 2, 3
	이미지 처리를 활용한 CCTV 현장 안전관리	회귀, 분류, 클러스터링	이상 탐지	이미지 - 1, 2, 3 동영상 - 1, 2, 3
도매/소매	제품 추천 및 기획	분류, 회귀		이미지 - 1, 2, 3 동영상 - 1, 2, 3
	시각적 검색	분류, 클러스터링		이미지 - 1, 2, 3 동영상 - 1, 2, 3
운수/장고	가격 최적화	회귀		Out of Scope
	실시간 경로 최적화	회귀, 강화학습		Out of Scope
정보통신	운송 자량 관리	분류		이미지 - 1, 3
	네트워크 최적화	회귀		Out of Scope
금융	소셜미디어 감성 분석	분류, 클러스터링	자연어 처리	Out of Scope
	이상 통신 탐지	회귀, 분류, 클러스터링	이상 탐지	Out of Scope
전문 서비스 (법률, 회계)	고객 맞춤 서비스	분류, 클러스터링	자연어 처리	Out of Scope
	금융 시장 분석	회귀		Out of Scope
과학/기술 (연구개발, 건축, 엔지니어링, 기타 과학기술)	비즈니스 분석 및 보고	회귀		Out of Scope
	판례 분석	분류, 클러스터링	자연어 처리	Out of Scope
교육	계약 점검 및 관리	분류, 클러스터링	자연어 처리	Out of Scope
	소송 결과 예측	분류, 클러스터링	자연어 처리	Out of Scope
의료/헬스케어	건축 설계도면 분석	분류		이미지 - 3
	고통 완화 정보 예측	회귀		동영상 - 1, 3 Out of Scope
	변역 및 통역	분류, 클러스터링	자연어 처리	Out of Scope
	개인 맞춤형 교육 제공	분류, 클러스터링	자연어 처리	Out of Scope
자율주행	문제 분석	분류		Out of Scope
	환자 데이터 및 위험 분석	회귀, 분류, 클러스터링	이상 탐지, 자연어 처리	Out of Scope
국방	라이프스타일 관리 및 모니터링	회귀, 분류, 클러스터링	이상 탐지, 자연어 처리	Out of Scope
	계약 이상사실 최적화	분류	자연어 처리	Out of Scope
	자율주행자동차	분류, 강화학습	각제 탐지	동영상 - 2, 3
	무인 군사기	분류	각제 탐지	동영상 - 2, 3
	지능형 표적탐지레이더	분류	각제 탐지	동영상 - 2, 3



2. What to build

Architecture Selection Part

(서비스 이름)은 5개의 특성을 기준으로 사용자 맞춤형 모델을 생성합니다.
이를 위해, Architecture Selection Part에서는 Area, Data Type, Task, Complexity, Resource에 관한 세부 항목을 선택합니다.
화면에서 용도에 맞게 선택하시고, [다음] 버튼을 눌러 아키텍처를 생성하세요.

1. Area

Area는 산업 분야를 고려하여 적합한 아키텍처를 선정해주는 분류 기준입니다.
사용할 분야를 선택해주세요.

2. Data Type

Data Type은 인공지능 학습을 위해 사용할 데이터의 형식을 지정하는 분류 기준입니다.
사용할 데이터 형식을 선택해주세요.

3. Task

Task는 모델의 용도를 지정하는 분류 기준입니다.
아래의 내용을 참고하여 알맞은 용도를 선택해주세요.

Classification : 이미지 분류
Detection : 이미지 내의 객체 위치를 시각화 단위로 할지
Segmentation : 이미지 내의 객체 위치를 픽셀 단위로 할지
Regression : 연속적인 데이터를 기반으로 값 예측
Clustering : 정제되지 않은 데이터를 기반으로 하는 이미지 분류

4. Complexity

Complexity는 모델의 복잡도에 관한 분류 기준입니다.
복잡도가 높을수록 연산횟수가 많아집니다.
아래 내용을 참고하여 복잡도를 선택하세요.

Low : 0M ~ 10M
Mid : 10M ~ 75M
High : 75M ~

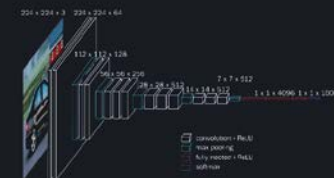
5. Resource

Resource는 모델을 학습 및 구동하는 데에 소요되는 FLOPs에 관한 분류 기준입니다.
FLOPs가 높을수록 학습 시간이 많이 소요되거나 좋은 성능을 얻을 수 있습니다.
아래 내용을 참고하여 Resource를 선택하세요.

Low : 0 ~ 5
Mid : 5 ~ 15
High : 15 ~

Architecture Selection Part

선택하신 특성들을 기반으로 아래의 아키텍처를 추천드립니다.
그대로 진행하시거나 [설정] 버튼을, 다시 선택하시려면 [이전] 버튼을 클릭해주세요.



VGG16

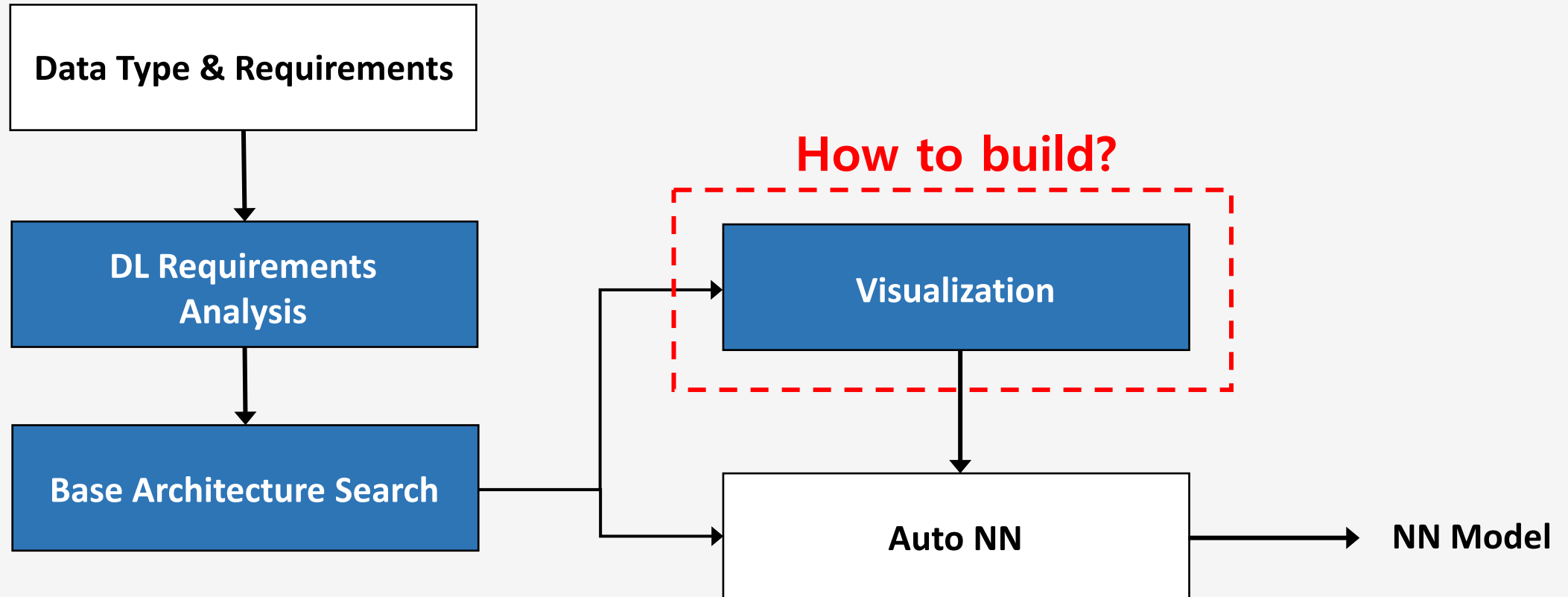
Area	Medical
Data Type	Image
Task	Classification
Complexity	Mid
Resource	Mid

3. How to build

How to build

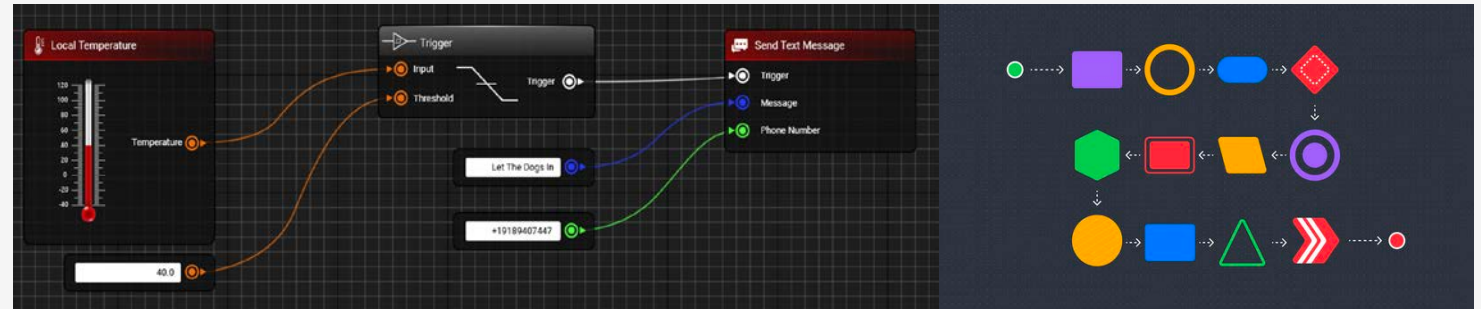
- 시각 프로그래밍 -

3. How to build



3. How to build

Visual Programming

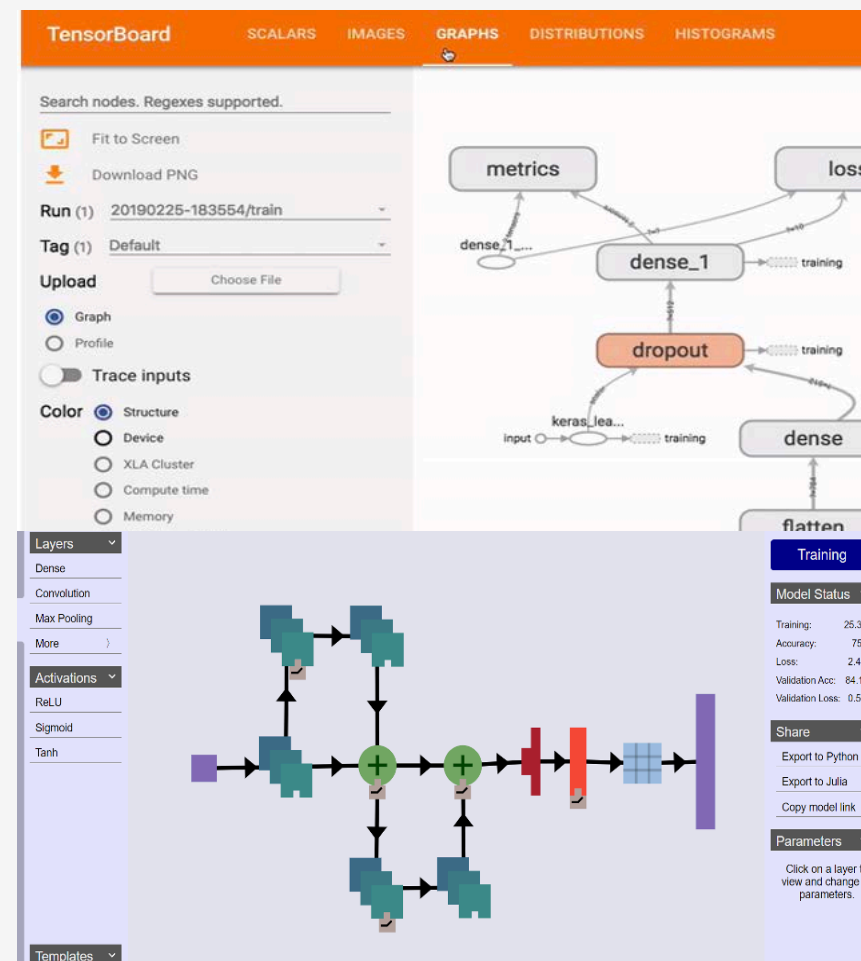


- 사용자가 텍스트 대신 그래픽적으로 구성 요소를 조작하여 프로그램 개발
- 직관적인 인터페이스를 통해 “No-Code” 플랫폼에서 효율적으로 작동
- 아이콘, 버튼, 기호와 같은 그래픽 구성 요소를 코딩 형태로 사용
- 초보자도 이해할 수 있는 방식으로 동작 및 메커니즘을 설명하기에 효율적

3. How to build

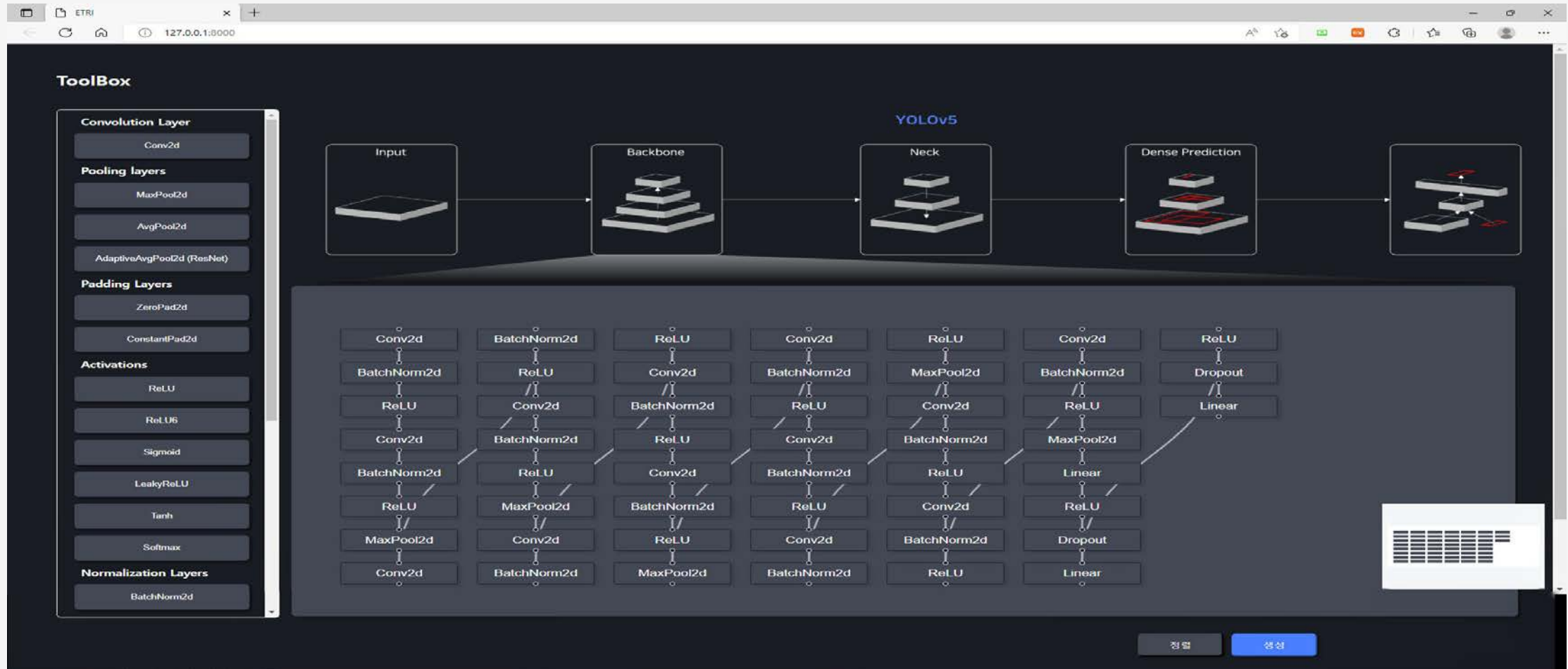
Visual Programming & Deep Learning

- 다수의 딥러닝 모델 구조 시각화 연구
 - 구조 편집, 파라미터 수정, 코드 자동 생성 미지원
- 다수의 딥러닝 시각 프로그래밍 연구
 - 특정 프레임워크만 지원
 - 제한적인 구조, 파라미터, 레이어 타입 지원



3. How to build

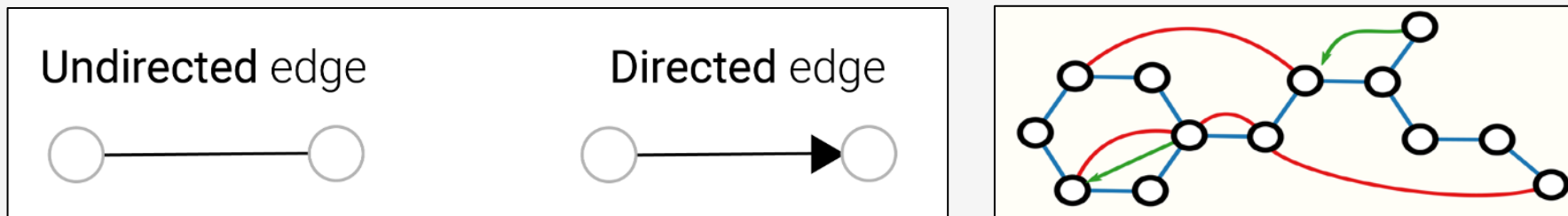
Vis2code



3. How to build

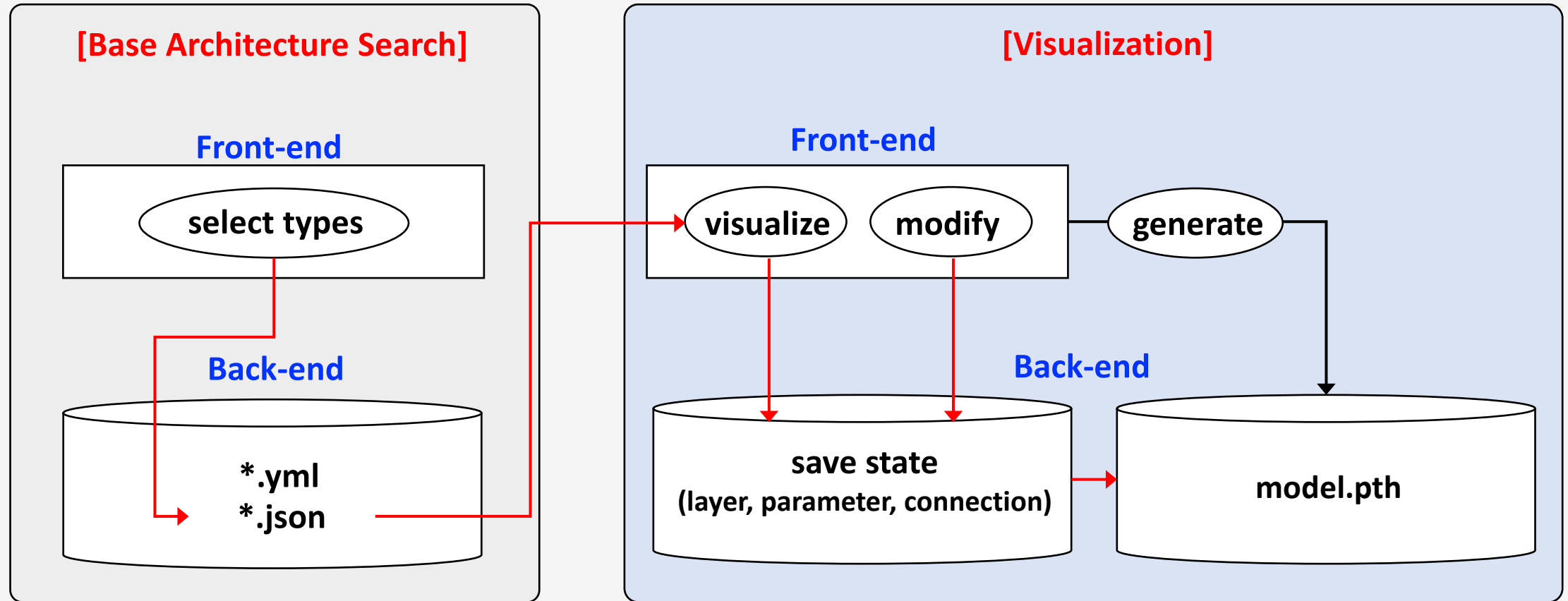
Vis2code

- 일반 프로그래밍: 텍스트 형태의 토큰 (1차원 관계)
- 시각 프로그래밍: 그래픽 형태의 토큰 (2차원 혹은 3차원 관계)
 - 토큰 직렬화를 통한 토큰 parsing
- SRG (Spatial Relations Graph): 그래픽 객체들의 관계 및 형태를 방향성 있게 표현



3. How to build

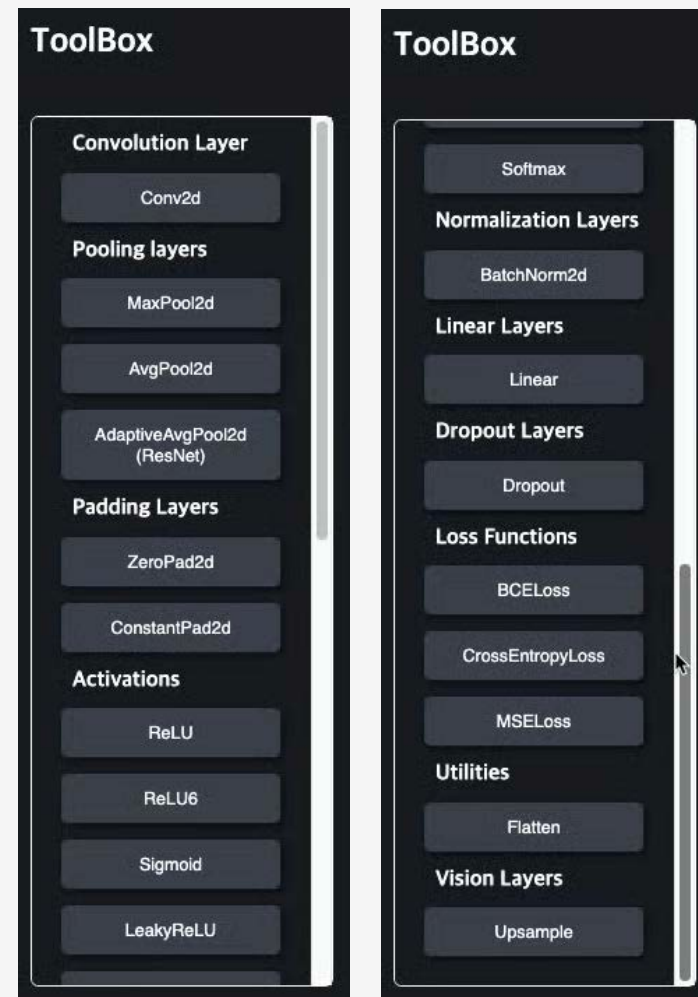
Vis2code



3. How to build

Vis2code (구성 요소)

- 19 종의 그래픽 구성 요소 정의
- 신경망 관련 산출물 분석을 통해 모델링 기본 단위 선별
- PyTorch 문법 기반 정의



3. How to build

Vis2code (구성 요소)

- PyTorch 문법을 기준으로 각 토큰(레이어, 기능) 별 코드 정의

```
[
  {
    "architecture": "yolov5",
    "yaml": "경로/yolov5.yaml",
    "json": "경로/yolov5.json"
    ...(생략)...
  }
]
```

< architecture >

```
[
  {
    "order": 1,
    "layer": "Conv2d",
    "parameters": "'in_channels': 3 Wn
    ...(생략)... 'padding': (1, 1)"
  },
  {
    "order": 2,
    "layer": "MaxPool2d",
    "parameters": "'kernel_size': (2, 2) Wn
    ...(생략)... 'ceil_mode': False"
  },
  ...(생략)...
]
```

< node >

```
[
  {
    "id": 1,
    "prior": 1,
    "next": 2
  },
  {
    "id": 2,
    "prior": 2,
    "next": 3
  },
  ...(생략)...
]
```

< edge >

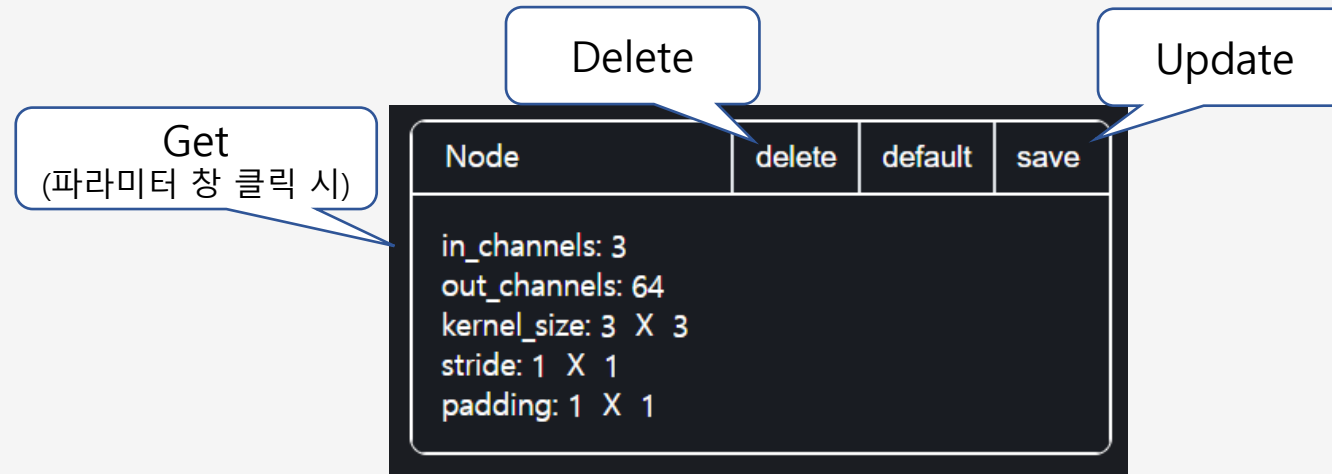
Vis2code (신경망 구성 - Layer, Connection)



3. How to build

Vis2code (신경망 구성 - Parameter)

- 각 레이어의 Parameter를 세부적으로 제어하는 동작 디자인



3. How to build

Vis2code (신경망 구성 - Parameter)



3. How to build

Vis2code (모델 코드 자동 생성)

The screenshot displays the Vis2code interface, which is used for automatically generating model code. On the left, a diagram illustrates the YOLOv5 architecture, showing the Backbone, Neck, Dense Prediction, and Sparse Prediction components. Below this, a grid of model components is shown, including Conv2d, BatchNorm2d, ReLU, MaxPool2d, and Linear layers, connected by arrows to represent the flow of the network. On the right, a list of generated PyTorch code blocks is shown, detailing the configuration for each layer, such as kernel_size, stride, padding, and activation functions. The code is organized into a Sequential model structure.

```
(49): ReLU(inplace=True)
(50): Dropout(p=0.5, inplace=False)
)
Linear
Sequential(
  (1): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (3): ReLU()
  (4): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (5): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (6): ReLU()
  (7): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=(0, 0), dilation=1, ceil_mode=False)
  (8): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (9): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (10): ReLU()
  (11): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (12): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (13): ReLU()
  (14): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=(0, 0), dilation=1, ceil_mode=False)
  (15): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (16): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (17): ReLU()
  (18): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (19): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (20): ReLU()
  (21): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (22): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (23): ReLU()
  (24): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=(0, 0), dilation=1, ceil_mode=False)
  (25): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (26): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (27): ReLU()
  (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (29): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (30): ReLU()
  (31): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (32): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (33): ReLU()
  (34): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=(0, 0), dilation=1, ceil_mode=False)
  (35): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (36): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (37): ReLU()
  (38): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (39): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (40): ReLU()
  (41): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (42): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (43): ReLU()
  (44): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=(0, 0), dilation=1, ceil_mode=False)
  (45): Linear(in_features=25088, out_features=4096, bias=True)
  (46): ReLU(inplace=True)
  (47): Dropout(p=0.5, inplace=False)
  (48): Linear(in_features=4096, out_features=4096, bias=True)
  (49): ReLU(inplace=True)
```

3. How to build

The screenshot shows a web browser window with the URL 127.0.0.1:8000. The page title is "Architecture Selection Part". Below the title, there is a paragraph explaining the purpose of the interface: "(서비스 이름)은 5개의 특성을 기준으로 사용자 맞춤형 모델을 생성합니다. 이를 위해, Architecture Selection Part에서는 Area, Data Type, Task, Complexity, Resource에 관한 세부 항목을 선택합니다. 하단에서 용도에 맞게 선택하시고, [다음] 버튼을 눌러 아키텍처를 생성하세요."

1. Area

Area는 산업 분야를 고려하여 적합한 아키텍처를 선정해주는 분류 기준입니다.
사용할 분야를 선택해주세요.

The "Area" section contains two rows of buttons for selection:

- Row 1: Agriculture, Medical, Manufacturing, Transportation, Human Resource, Law
- Row 2: Retail, Security, Finance

2. Data Type

Data Type은 인공지능 학습을 위해 사용할 데이터의 형식을 지정하는 분류 기준입니다.
사용할 데이터 형식을 선택해주세요.

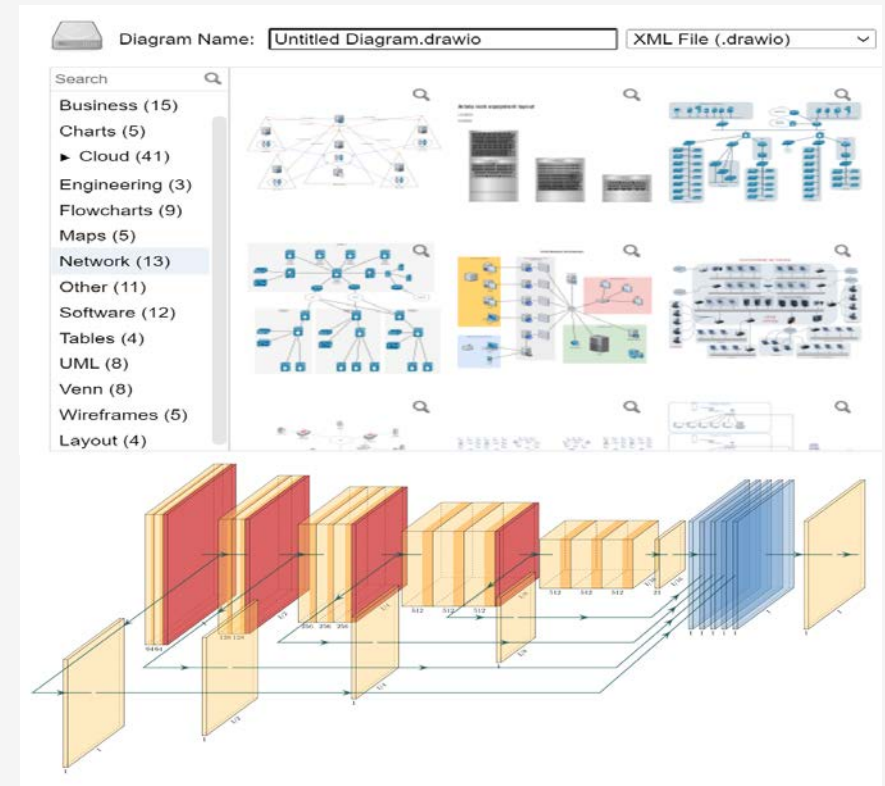
The "Data Type" section contains a row of buttons for selection:

- Row 1: Image, Video, Text, Voice

4. Concluding Remarks

Ongoing Research

- **신경망 요구사항 분석**
 - 요구사항 분석 프로세스 구체화 및 검증
 - 구조 탐색 알고리즘 구현 및 최적화
 - 사용자 피드백 및 GUI 개선
- **시각 프로그래밍**
 - 지원 아키텍처 및 구성 요소 다양화
 - 구조 및 파라미터 업데이트 메커니즘 다양화
 - 코드 자동 생성 기능 안정화
 - 사용자 피드백 및 GUI 개선



감사합니다.