

대분류/20  
정보통신

중분류/01  
정보기술

소분류/07  
인공지능

세분류/05  
인공지능서비스구현

능력단위/04

NCS 학습모듈

# 인공지능서비스 애플리케이션 개발

LM2001070504\_20v1



교육부

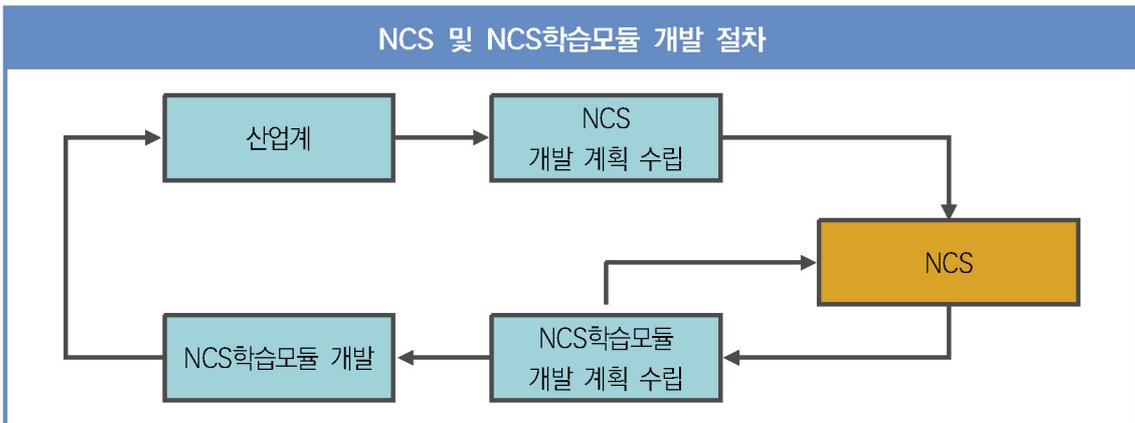
NCS 학습모듈은 교육훈련기관에서 출처를 명시하고 교육적 목적으로 활용할 수 있습니다. 다만 NCS 학습모듈에는 국가(교육부)가 저작권 일체를 보유하지 않은 저작물들(출처가 표기되어 있는 도표, 사진, 삽화, 도면 등)이 포함되어 있으므로 이러한 저작물들의 변형, 복제, 공연, 배포, 공중 송신 등과 이러한 저작물들을 활용한 2차 저작물의 생성을 위해서는 반드시 원작자의 동의를 받아야 합니다.

# NCS학습모듈의 이해

※ 본 NCS학습모듈은 「NCS 국가직무능력표준」사이트(<http://www.ncs.go.kr>) 에서 확인 및 다운로드할 수 있습니다.

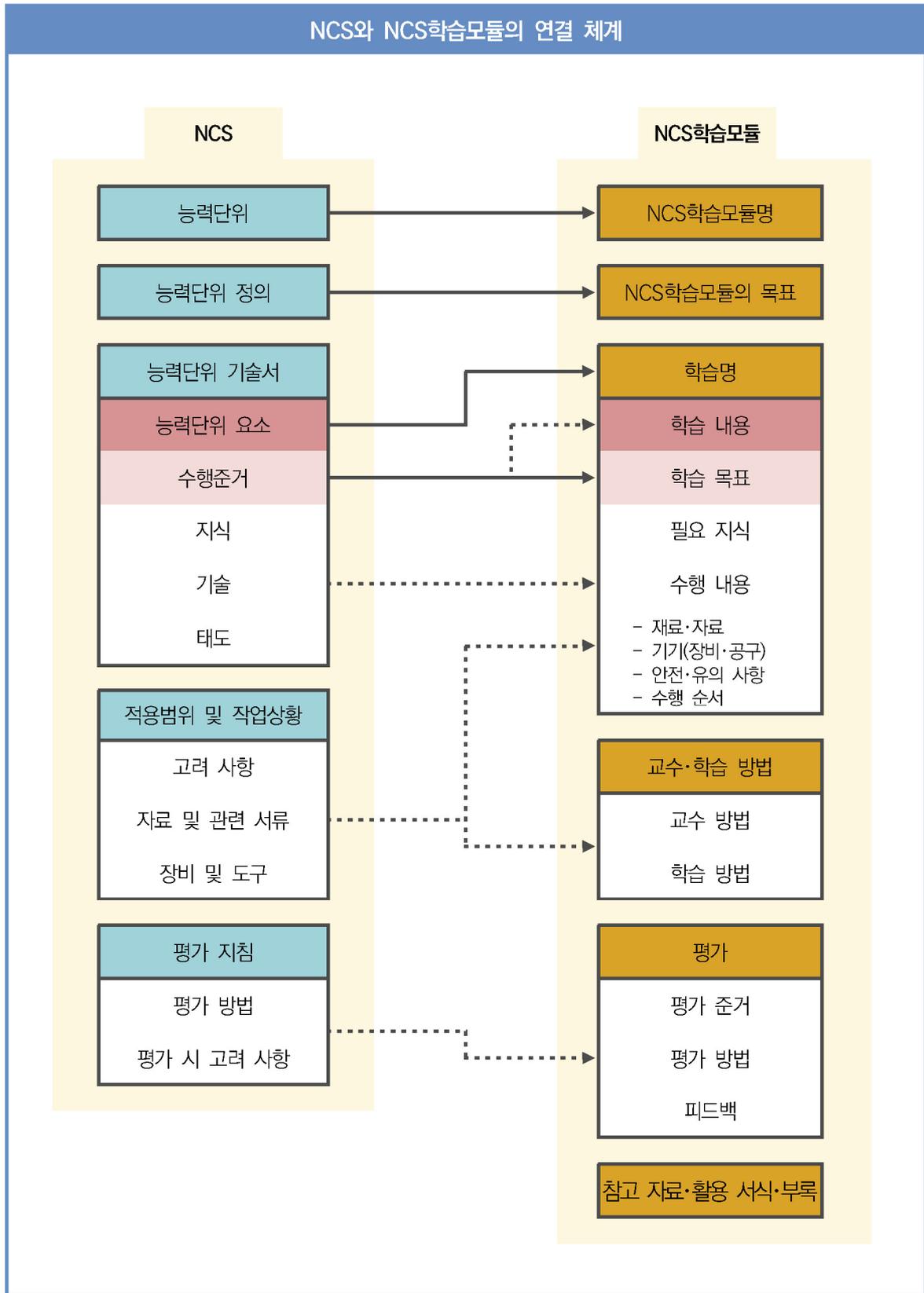
## I NCS학습모듈이란?

- 국가직무능력표준(NCS: National Competency Standards)이란 산업현장에서 직무를 수행하기 위해 요구되는 지식·기술·소양 등의 내용을 국가가 산업부문별·수준별로 체계화한 것으로 산업현장의 직무를 성공적으로 수행하기 위해 필요한 능력(지식, 기술, 태도)을 국가적 차원에서 표준화한 것을 의미합니다.
- 국가직무능력표준(이하 NCS)이 현장의 ‘직무 요구서’라고 한다면, **NCS학습모듈은 NCS의 능력단위를 교육훈련에서 학습할 수 있도록 구성된 ‘교수·학습 자료’입니다.** NCS학습모듈은 구체적 직무를 학습할 수 있도록 이론 및 실습과 관련된 내용을 상세하게 제시하고 있습니다.



- **NCS학습모듈은 다음과 같은 특징을 가지고 있습니다.**
  - 첫째, NCS학습모듈은 산업계에서 요구하는 직무능력을 교육훈련 현장에 활용할 수 있도록 성취목표와 학습의 방향을 명확히 제시하는 가이드라인의 역할을 합니다.
  - 둘째, NCS학습모듈은 특성화고, 마이스터고, 전문대학, 4년제 대학교의 교육기관 및 훈련기관, 직장교육기관 등에서 표준교재로 활용할 수 있으며 교육과정 개편 시에도 유용하게 참고할 수 있습니다.

○ NCS와 NCS학습모듈 간의 연결 체계를 살펴보면 아래 그림과 같습니다.



## II NCS학습모듈의 체계

○ NCS학습모듈은 1. NCS학습모듈의 위치, 2. NCS학습모듈의 개요, 3. NCS학습모듈의 내용 체계, 4. 참고 자료, 5. 활용서식/부록 으로 구성되어 있습니다.

### 1. NCS학습모듈의 위치

○ NCS학습모듈의 위치는 NCS 분류 체계에서 해당 학습모듈이 어디에 위치하는지를 한 눈에 볼 수 있도록 그림으로 제시한 것입니다.

[NCS-학습모듈의 위치]		
대분류	문화·예술·디자인·방송	
중분류	문화콘텐츠	
소분류	문화콘텐츠제작	
<b>세분류</b>		
<b>방송콘텐츠제작</b>	<b>능력단위</b>	<b>학습모듈명</b>
영화콘텐츠제작	프로그램 기획	프로그램 기획
음악콘텐츠제작	아이템 선정	아이템 선정
광고콘텐츠제작	자료 조사	자료 조사
게임콘텐츠제작	프로그램 구성	프로그램 구성
애니메이션 콘텐츠제작	캐스팅	캐스팅
만화콘텐츠제작	<b>제작계획</b>	<b>제작계획</b>
캐릭터제작	방송 미술 준비	방송 미술 준비
스마트문화앱 콘텐츠제작	방송 리허설	방송 리허설
영사	야외촬영	야외촬영
	스튜디오 제작	스튜디오 제작
	...	...

**학습모듈은**

NCS 능력단위 1개당 1개의 학습모듈 개발을 원칙으로 합니다. 그러나 필요에 따라 고용단위 및 교과단위를 고려하여 능력단위 몇 개를 묶어 1개 학습모듈로 개발할 수 있으며, NCS 능력단위 1개를 여러 개의 학습모듈로 나누어 개발할 수도 있습니다.

## 2. NCS학습모듈의 개요

○ NCS학습모듈의 개요는 학습모듈이 포함하고 있는 내용을 개략적으로 설명한 것으로

학습모듈의 목표, 선수학습, 학습모듈의 내용 체계, 핵심 용어 로 구성되어 있습니다.

학습모듈의 목표	해당 NCS 능력단위의 정의를 토대로 학습 목표를 작성한 것입니다.
선수학습	해당 학습모듈에 대한 효과적인 교수·학습을 위하여 사전에 이수해야 하는 학습모듈, 학습 내용, 관련 교과목 등을 기술한 것입니다.
학습모듈의 내용 체계	해당 NCS 능력단위요소가 학습모듈에서 구조화된 체계를 제시한 것입니다.
핵심 용어	해당 학습모듈의 학습 내용, 수행 내용, 설비·기자재 등 가운데 핵심적인 용어를 제시한 것입니다.

### 제작계획 학습모듈의 개요

#### 학습모듈의 목표

본격적인 촬영을 준비하는 단계로서, 촬영 대본을 확정하고 제작 스태프를 조직하며 촬영 장비와 촬영 소품을 준비할 수 있다.

#### 선수학습

제작 준비(LM0803020105\_13v1), 섭외 및 제작스태프 구성(LM0803020104\_13v1), 촬영 제작(LM0803020106\_13v1), 촬영 장비 준비(LM0803040204\_13v1.4), 미술 디자인 협의하기(LM0803040203\_13v1.4)

#### 학습모듈의 내용체계

학습	학습 내용	NCS 능력단위 요소	
		코드번호	요소 명칭
1. 촬영 대본 확정하기	1-1. 촬영 구성안 검토와 수정	0803020114_16.3.1	촬영 대본 확정하기
	2-1. 기술 스태프 조직		0803020114_16.3.2
2. 제작 스태프 조직하기	2-2. 미술 스태프 조직	제작 스태프 조직하기	
	2-3. 전문 스태프 조직		
3. 촬영 장비 계획하기	3-1. 촬영 장비 점검과 준비	0803020114_16.3.3	촬영 장비 계획하기
	4-1. 촬영 소품 목록 작성		0803020114_16.3.4
4. 촬영 소품 계획하기	4-2. 촬영 소품 제작 의뢰	촬영 소품 계획하기	

#### 핵심 용어

촬영 구성안, 제작 스태프, 촬영 장비, 촬영 소품

#### 학습모듈의 목표는

학습자가 해당 학습모듈을 통해 성취해야 할 목표를 제시한 것으로, 교수자는 학습자가 학습모듈의 전체적인 내용흐름을 파악하도록 지도할 수 있습니다.

#### 선수학습은

교수자 또는 학습자가 해당 학습모듈을 교수·학습하기 이전에 이수해야 하는 교과목 또는 학습모듈(NCS 능력단위) 등을 표기한 것입니다. 따라서 교수자는 학습자가 개별 학습, 자기 주도 학습, 방과 후 활동 등 다양한 방법을 통해 이수할 수 있도록 지도하는 것을 권장합니다.

#### 핵심 용어는

해당 학습모듈을 대표하는 주요 용어입니다. 학습자가 해당 학습모듈을 통해 학습하고 평가받게 될 주요 내용을 알 수 있습니다. 「NCS 국가직무능력표준」 사이트(www.ncs.go.kr)의 색인(찾아보기) 중 하나로 이용할 수 있습니다.

### 3. NCS학습모듈의 내용 체계

○ NCS학습모듈의 내용은 크게 **학습**, **학습 내용**, **교수·학습 방법**, **평가**로 구성되어 있습니다.

<b>학습</b>	해당 NCS 능력단위요소 명칭을 사용하여 제시한 것입니다. 학습은 크게 학습 내용, 교수·학습 방법, 평가로 구성되며 해당 NCS 능력단위의 능력단위 요소별 지식, 기술, 태도 등을 토대로 내용을 제시한 것입니다.
<b>학습 내용</b>	학습 내용은 학습 목표, 필요 지식, 수행 내용으로 구성되며, 수행 내용은 재료·자료, 기기(장비·공구), 안전·유의 사항, 수행 순서, 수행 tip으로 구성된 것입니다. 학습모듈의 학습 내용은 실제 산업현장에서 이루어지는 업무활동을 표준화된 프로세스에 기반하여 다양한 방식으로 반영한 것입니다.
<b>교수·학습 방법</b>	학습 목표를 성취하기 위한 교수자와 학습자 간, 학습자와 학습자 간 상호 작용이 활발하게 일어날 수 있도록 교수자의 활동 및 교수 전략, 학습자의 활동을 제시한 것입니다.
<b>평가</b>	평가는 해당 학습모듈의 학습 정도를 확인할 수 있는 평가 준거 및 평가 방법, 평가 결과의 피드백 방법을 제시한 것입니다.

학습 1	촬영 대본 확정하기
학습 2	제작 스태프 조직하기
학습 3	촬영 장비 계획하기
학습 4	촬영 소품 계획하기

## 2-1. 기술 스태프 조직

**학습 목표** • 프로그램 제작에 적합한 기술 스태프를 조직할 수 있다.

**필요 지식** /

- ① 기술 스태프의 구성
 

프로그램의 장르에 따라 구성하는 기술 스태프는 많은 차이가 있다. 같은 장르의 프로그램이라도 그 형식이나 내용, 규모에 따라서 구성되는 기술 스태프의 종류와 인원 수는 천차만별이다.

  1. 스튜디오 프로그램
 

토크쇼, 종합 구성, 예능과 같은 스튜디오 프로그램은 부조정실과 스튜디오를 사용하여 제작하기 때문에 많은 기술 스태프가 필요하다.

**학습은**

해당 NCS 능력단위요소 명칭을 사용하여 제시하였습니다. 하나의 학습은 일반교과의 '대단원'에 해당되며, 학습모듈을 구성하는 가장 큰 단위가 됩니다. 또한 하나의 직무를 수행하기 위한 가장 기본적인 단위로 사용할 수 있습니다.

**학습 내용은**

NCS 능력단위요소별 수행준거를 기준으로 제시하였습니다. 일반교과의 '중단원'에 해당합니다.

**학습 목표는**

학습 내용을 이수할 때 학습자가 갖춰야 할 행동 수준을 의미합니다. 따라서 수업시간의 과목 목표로 활용할 수 있습니다.

**필요 지식은**

해당 NCS의 지식을 토대로 학습에 대한 이해와 성과를 제고하기 위해 반드시 알아야 할 주요 지식을 제시하였습니다. 필요 지식은 수행에 꼭 필요한 핵심 내용을 위주로 제시하여 교수자의 역할이 매우 중요하며, 이후 수행 순서와 연계하여 교수·학습으로 진행할 수 있습니다.

## 수행 내용 / 기술 스태프 구성표 작성하기

### 재료·자료

- 방송프로그램 제작 기획서 및 방송 대본, 콘티(continuity), 제작 일정, 운용표
- 장비 및 시설, 제작 시설 배정 의뢰서 및 배정표, 방송 기술 스태프 데이터베이스(DB) 자료

### 기기(장비·공구)

- 컴퓨터 등

### 안전·유의 사항

- 프로그램의 내용과 제작 방법을 분석하고, 각 스태프들의 역할을 신중하게 검토한다.

### 수행 순서

1. 방송 대본이나 콘티(continuity), 큐 시트를 분석하고, 프로그램의 내용적 특성, 제작 과정에 대한 자료를 수집한다.
2. 프로그램 제작 방법을 결정한다.
  1. 스튜디오 녹화를 할 것인가, 야외 촬영을 할 것인가 검토한다.

#### 수행 tip

- 스태프의 결정은 스태프 간의 호흡을 중 요시하여 선정해야 프로그램의 질을 향상시킬 수 있다.

#### 수행 내용은

해당 학습모듈에서 제시한 내용 중 기술(skill)을 습득하기 위한 실습과제로 활용할 수 있습니다.

#### 재료·자료는

수행 내용을 수행하는데 필요한 재료 및 준비물로 실습 시 활용할 수 있습니다.

#### 기기(장비·공구)는

수행 내용에 필요한 기본적인 장비 및 도구를 제시하였습니다. 제시된 기기 외에도 수행에 필요한 다양한 도구나 장비를 활용할 수 있습니다.

#### 안전·유의사항은

수행 내용을 수행하는 데 있어 안전상 주의해야 할 점 및 유의사항을 제시하였습니다. 실습 시 유념해야 하며, NCS의 고려 사항도 추가적으로 활용할 수 있습니다.

#### 수행 순서는

실습 과제의 진행 순서로 활용할 수 있습니다.

#### 수행 tip은

수행 내용에서 실습을 용이하게 할 수 있는 아이디어를 제시하였습니다. 수행 tip은 지도상의 안전 및 유의사항 외에 전반적으로 적용되는 주안점 및 수행 과제 목적에 대한 보충설명, 추가사항 등으로 활용할 수 있습니다.

## 학습2

## 교수·학습 방법

### 교수 방법

- 방송 프로그램의 기술적 요소, 미술 구성 요소, 특수 촬영에 대해 설명한다.
- 방송 프로그램 제작에서 각 기술 스태프의 역할에 대해 설명한다.
- 방송 프로그램을 분석하고 필요한 기술 스태프를 구성할 수 있도록 지시한다.

### 학습 방법

- 방송 프로그램의 기술적 요소, 미술 구성 요소, 특수 촬영에 대해서 알아본다.
- 프로그램 제작에 필요한 기술 스태프의 역할을 이해하고, 기술 스태프 구성표를 작성한다.

#### 교수·학습 방법은

학습 목표를 성취하는 데 필요한 교수 방법과 학습 방법을 제시하였습니다.

#### 교수 방법은

해당 학습 활동에 필요한 학습 내용, 학습 내용과 관련된 자료명, 자료 형태, 수행 내용의 진행 방식 등에 대하여 제시하였습니다. 또한 학습자의 수업참여도 제고 방법 및 수업 진행상 유의사항 등도 제시하였습니다. 선수학습이 필요한 학습을 학습자가 숙지하였는지 교수가 확인하는 과정으로 활용할 수도 있습니다.

#### 학습 방법은

해당 학습 활동에 필요한 학습자의 자기 주도 학습 방법을 제시하였습니다. 또한 학습자가 숙달해야 할 실기 능력과 학습 과정에서 주의해야 할 사항 등도 제시하였습니다. 학습자가 학습을 이수하기 전 반드시 숙지해야 할 기본 지식을 학습하였는지 스스로 확인하는 과정에 활용할 수 있습니다.

## 학습2      평 가

### 평가 준거

- 평가자는 학습자가 학습 목표를 성공적으로 달성하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습 내용	학습 목표	성취수준 상 중 하
기술 스태프 조직	- 프로그램 제작에 적합한 기술 스태프를 조직할 수 있다.	
미술 스태프 조직	- 프로그램 제작에 적합한 미술 스태프를 조직할 수 있다.	
전문 스태프 조직	- 프로그램 특수 촬영을 위한 전문 스태프를 조직할 수 있다.	

### 평가 방법

- 사례 연구

학습 내용	평가 항목	성취수준 상 중 하
기술 스태프 조직	- 프로그램에서 기술적 요소의 파악 여부	
	- 기술 스태프의 역할 파악 여부	
	- 프로그램에 필요한 기술 스태프 구성표 작성 능력	

### 피드백

1. 사례 연구

- 프로그램을 선택하여 기술 스태프, 미술 스태프, 전문 스태프 구성표를 예시와 같이 작성하였는지 개인별 능력을 평가한 후, 그 결과를 모든 학습자에게 공유하도록 한다.

**평가는**  
NCS 능력단위의 평가 방법과 평가 시 고려사항을 준용하여 작성합니다. 교수자와 학습자가 평가 항목별 성취수준 확인 시 활용할 수 있습니다.

**평가 준거는**  
학습자가 학습을 어느 정도 성취하였는지 평가하기 위한 기준을 제시하고 있습니다. 학습 목표와 연계하여 단위수업 시간에 평가 항목 별 성취수준을 평가하는 데 활용할 수 있습니다.

**평가 방법은**  
NCS 능력단위의 평가 방법을 참고하였으며, 평가 준거에 따른 평가 방법을 2개 이상 제시합니다. 평가 방법의 종류는 포트폴리오, 문제해결 시나리오, 서술형 시험, 논술형 시험, 사례 연구, 평가자 체크리스트, 작업장 평가 등이 있으며, NCS 능력단위 요소 별 수행 수준을 평가하는 데 가장 적절한 방법을 선정하여 활용할 수 있습니다.

**피드백은**  
평가 후에 학습자들에게 평가 결과를 피드백하여 학습 목표를 달성하는 데 활용할 수 있습니다.

## 4. 참고 자료

### 참고 자료

- 교육부(2013). 섭외 및 제작스태프 구성(LM0803020104\_13v1). 한국직업능력개발원.

**참고 자료는**  
해당 학습מוד에 제시된 인용 자료의 출처를 제시하였습니다. 교수·학습의 과정에서 참고로 활용할 수 있습니다.

## 5. 활용 서식/부록

### 활용 서식

스튜디오 기술 스태프 구성표					
직종	이름	연락처	소속	특이사항	비고
기술감독					
소명감독					

**활용 서식은**  
평가 서식, 실습 시트 등 교수·학습 시 활용할 수 있는 다양한 서식들로 구성하였습니다. 수행에서 평가에 이르기까지 필요한 서식을 해당 모듈의 특성에 맞춰 개발하거나 기존의 양식을 활용하여 제시하였습니다.

### 부록

[디지털 텔레비전 방송프로그램 음량 등에 관한 기준]  
제정 2014. 11. 29. 미래창조과학부 고시 제2014-87호

**제1장 총칙**

**제1조(목적)** 이 고시는 방송법 제70조의2제1항에 따라 방송사업자가 디지털 텔레비전 방송프로그램 및 방송광고의 음량을 일정하게 유지하기 위해 필요한 사항을 규정함을 목적으로 한다.

**부록은**  
활용 서식 이외에 교수·학습 과정에서 참고할 수 있는 자료가 있는 경우 제시하였습니다.

## [NCS-학습모듈의 위치]

대분류	정보통신
중분류	정보기술
소분류	인공지능

세분류	능력단위	학습모듈명
인공지능플랫폼 구축	인공지능서비스 구현 요건 분석	인공지능서비스 구현 요건 분석
인공지능서비스 기획	인공지능서비스 상위 설계	인공지능서비스 상위 설계
인공지능모델링	인공지능서비스 상세 설계	인공지능서비스 상세 설계
인공지능서비스 운영 관리	인공지능서비스 애플리케이션 개발	인공지능서비스 애플리케이션 개발
인공지능서비스 구현	인공지능서비스 모델 적용	인공지능서비스 모델 적용
	인공지능서비스 인터페이스 개발	인공지능서비스 인터페이스 개발
	인공지능서비스 테스트	인공지능서비스 테스트
	인공지능서비스 이행	인공지능서비스 이행

---

# 차 례

---

학습모듈의 개요	1
----------	---

## 학습 1. 인공지능서비스 비즈니스 로직 구현하기

1-1. 인공지능서비스 개발 환경 이해	3
1-2. 비즈니스 이해 및 로직 구현	19
1-3. 비즈니스 로직 문제 해결	32
• 교수 · 학습 방법	40
• 평가	41

## 학습 2. 인공지능서비스 플랫폼 연동하기

2-1. 플랫폼 연동 환경 구성	43
2-2. 플랫폼 연동	52
2-3. 플랫폼 연동 문제 해결	59
• 교수 · 학습 방법	64
• 평가	65

## 학습 3. 인공지능서비스 모델 연동하기

3-1. 모델 연동 환경 구성	68
3-2. 모델 연동	74
3-3. 모델 연동 문제 해결	84
• 교수 · 학습 방법	89
• 평가	90

## 학습 4. 인공지능서비스 인터페이스 연동하기

4-1. 인터페이스 연동 환경 구성	94
4-2. 인터페이스 연동	103
4-3. 인터페이스 연동 문제 해결	112
• 교수 · 학습 방법	121
• 평가	122
<b>참고 자료</b>	<b>125</b>
<b>부    록</b>	<b>127</b>

# 인공지능서비스 애플리케이션 개발 학습모듈의 개요

## 학습모듈의 목표

비즈니스 로직을 구현하고 플랫폼, 모델, 인터페이스를 연동하는 응용 소프트웨어를 개발할 수 있다.

## 선수학습

화면 구현(2001020225\_19v5), 프로그래밍 언어 응용(2001020230\_19v4), 프로그래밍 언어 활용(2001020231\_19v4)

## 학습모듈의 내용체계

학습	학습 내용	NCS 능력단위 요소	
		코드번호	요소 명칭
1. 인공지능서비스 비즈니스 로직 구현하기	1-1. 인공지능서비스 개발 환경 이해	2001070504_20v1.1	인공지능서비스 비즈니스 로직 구현하기
	1-2. 비즈니스 이해 및 로직 구현		
	1-3. 비즈니스 로직 문제 해결		
2. 인공지능서비스 플랫폼 연동하기	2-1. 플랫폼 연동 환경 구성	2001070504_20v1.2	인공지능서비스 플랫폼 연동하기
	2-2. 플랫폼 연동		
	2-3. 플랫폼 연동 문제 해결		
3. 인공지능서비스 모델 연동하기	3-1. 모델 연동 환경 구성	2001070504_20v1.3	인공지능서비스 모델 연동하기
	3-2. 모델 연동		
	3-3. 모델 연동 문제 해결		
4. 인공지능서비스 인터페이스 연동하기	4-1. 인터페이스 연동 환경 구성	2001070504_20v1.4	인공지능서비스 인터페이스 연동하기
	4-2. 인터페이스 연동		
	4-3. 인터페이스 연동 문제 해결		

## 핵심 용어

인공지능, 플랫폼 연동, 모델 연동, 전이 학습, 파이선, 장고(django) 프레임워크, 인공지능서비스



## 학습 1

# 인공지능서비스 비즈니스 로직 구현하기

학습 2	인공지능서비스 플랫폼 연동하기
학습 3	인공지능서비스 모델 연동하기
학습 4	인공지능서비스 인터페이스 연동하기

## 1-1. 인공지능서비스 개발 환경 이해

### 학습 목표

- 인공지능서비스 애플리케이션 설계서에 따라 인공지능 비즈니스 로직 구현을 위한 개발 환경을 구성할 수 있다.

### 필요 지식 /

#### ① 인공지능 기반 비즈니스 환경

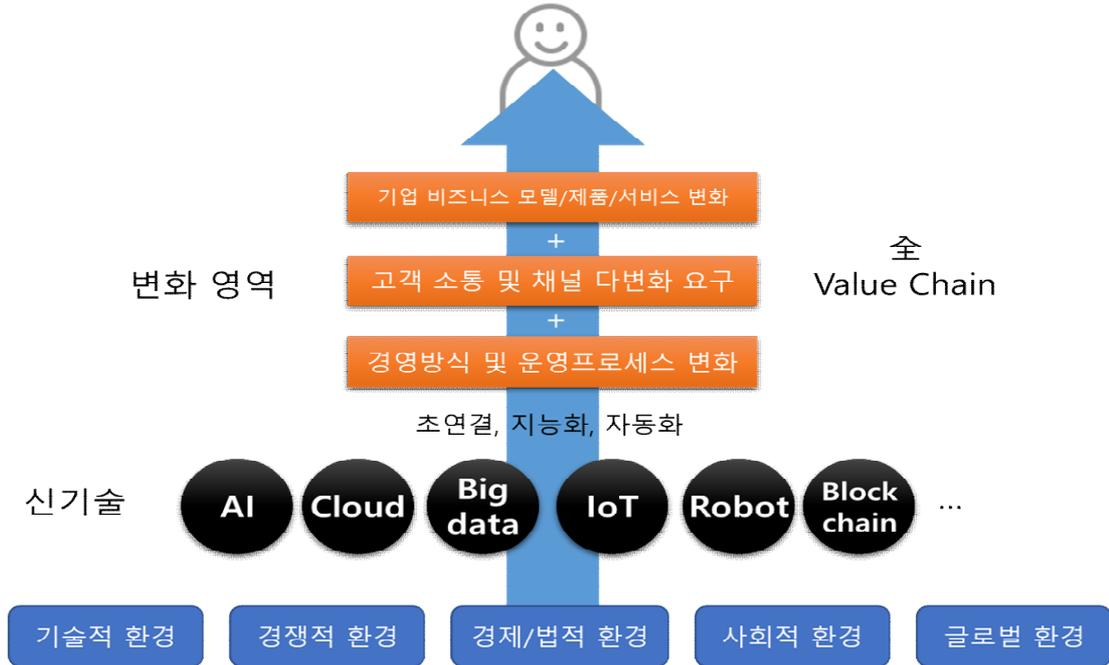
비즈니스(Business)는 조직 또는 그룹의 목적(이익, 정의 실현 등) 달성을 위해 재화 또는 서비스를 생산 및 제공하기 위한 조직적 노력과 활동을 의미한다. 비즈니스 목적 달성에는 비즈니스 환경이 크게 영향을 미치는데, 비즈니스 환경이란 '기업의 존속·성장에 영향을 미치거나 기회, 위협이 되는 기업의 외부 존재'를 의미하는 것으로 다음과 같이 비즈니스 환경에는 크게 다섯 가지 환경을 고려해 볼 수 있다.

<표 1-1> 비즈니스 환경의 이해

구분	개념	예시
기술적 환경	정보 통신 기술의 급격한 발전에 따라 전자 거래가 확대되고 기업의 운영 방식과 소비자의 구매 방식이 변화되는 환경	4차 산업 혁명, AI, 블록체인 등 도입
경쟁적 환경	고객 환불 제도, A/S 제도 등 기업 간 경쟁 심화로 고객의 기대는 계속 높아지며 고객 기대를 적절히 충족시키는 기업만이 생존	-
경제적·법적 환경	경제 시스템에 대한 정부의 역할에 따라 기업에게 리스크로 적용 - 단체를 통한 업계의 의견 적극적으로 개진 - 정부 정책에 대해 적극적으로 의견 제시	「개인 정보 보호법」에 따른 규제 강화 등
사회적 환경	인구 통계학적인 변화(다문화, 고령화 사회, 도시화 등)에 따른 적극적인 대응 중요	-
글로벌 환경	글로벌 경쟁과 자유 무역, 교육·통신 발전(인터넷을 통한 교류 확대)에 따라 글로벌 경쟁력을 확보하지 못하면 생존하기 어려움 - 경쟁 우위에 있는 제품 생산 특화, 국가 간 교역 활용 등	삼성전자, 샤오미, 애플, 테슬라 등

비즈니스 환경에서 4차 산업 혁명의 핵심 기술인 인공지능 기술은 비즈니스 혁신의 큰 동인(動因, Driver)으로 작용하고 있다. 아마존의 알렉사나 애플의 시리와 같은 인공지능서비스 시스템은 인간이 기계와 상호 작용하는 방식의 변화를 유도하고 있으며 인공지능 기반의 혁신적인 시스템은 다시 기업의 비즈니스 환경에도 크게 영향을 주고 있다.

“고객 가치 및 사용자 경험의 근본적인 혁신”



출처: 집필진 제작(2021)  
[그림 1-1] 비즈니스 환경 변화의 주요 동인

비즈니스 환경에서 인공지능서비스는 비즈니스의 목적 달성을 위해 일상 업무의 지원뿐만 아니라 데이터 분석과 개선, 데이터 흐름에서의 미세한 변화와 패턴 추적, 보고서의 요약, 제품 추천 및 맞춤형 고객 경험 제공 등 다양한 업무 분야에서 검토되거나 도입 시도되고 있다.

〈표 1-2〉 인공지능서비스 활용 분야의 사례

산업	활용 기업	사례
자동차	GM	텔레매틱스 서비스 'OnStar GO'
	Local Motors	자율 주행 전기 버스 'Olli' 탑재(12인승)
의료	IBM 왓슨 닥터	암 진단(앤더스 암센터 메모리얼슬론케터링 암센터/치료법 제안 MD)
제약	테바제약	신약 개발 맞춤형, 발병 예측
음악	알렉스 다 키드	데이터 분석을 통한 음악 작곡 및 동향 파악
IoT	시스코	기기 상태 모니터링(IoT 장비에 적용)
영화	20세기Fox	영화 예고편 편집(Morgan)
항공	에어버스	부품 교체 주기 파악, 생산 공정 관리

산업	활용 기업	사례
법률	ROSS	'로봇 변호사 로스', 판례 수집 및 분석
금융	웰스프린트, 베테먼트, 퓨처어드바이저	신용도 평가, 투자 종목 추천, 개인 투자자 금융 자산 관리
대학	조지아공대	조교로 활용, 학생 질의에 대한 답변 및 과제 점검
로봇	소프트뱅크	교육 및 서비스 로봇

출처: 교육부(2019). 『인공지능서비스 활용 기획(LM2001070206)』. 한국직업능력개발원. p. 4.

## ② 인공지능서비스 개발 환경의 구성

인공지능서비스 비즈니스 로직을 구현할 때에는 비즈니스의 목적과 설계 내용에 대한 명확한 이해가 필요하고, 개발 환경을 통해 구현되기 때문에 개발 환경의 구성을 위한 검토를 해야 한다.

### 1. 개발 환경의 이해

프로그램 개발에 필요한 모든 작업을 통합된 프로그램 안에서 처리하도록 구성된 통합 소프트웨어(개발 도구)와 이를 작동시키는 하드웨어를 '개발 환경'으로 정의한다. 일반적으로 프로그램 개발에는 컴파일러, 개발 언어용 텍스트 편집기, 디버거 등의 구성이 필요하다. 그러나 최근에는 이런 구성들을 통합(각 구성을 하나로 묶음.)하여 대화형 인터페이스를 제공하는 통합 개발 환경이 보편적으로 사용된다.

인공지능서비스 개발 환경을 구성하기 위해서는 도구의 각 구성과 특성을 파악하고, 필요 여부를 판단할 수 있어야 하며 개발 도구가 설치되는 하드웨어에 대해서도 이해할 수 있어야 한다.

### 2. 인공지능서비스 개발 언어 및 개발 도구

#### (1) 인공지능서비스 개발 언어

프로그램 개발 언어는 다양하며 인공지능서비스를 개발하는 개발 언어에 큰 제약은 없지만, 개발 언어를 선택할 때에는 양질의 머신러닝 및 딥러닝 라이브러리를 다양하게 지원하는지 검토해야 한다. 그리고 실행 성능과 다양한 도구 지원, 프로그래머의 커뮤니티 지원 여부도 고려하는 것이 로직 구현에 유리하다.

〈표 1-3〉 인공지능서비스 개발에 적합한 언어, 특징 및 개발 환경

개발 언어	주요 특징	개발 도구 및 환경
파이선 (python)	넘파이(NumPy), 판다스(Pandas), 자연어 처리(NLP)의 NLTK, 스페이스(SpaCy), 사이킷-러(Scikit-learn) 등 인공지능서비스 모델 구현을 위한 수많은 라이브러리를 오픈 소스로 제공한다. 특히 딥러닝 관련해서는 현재 대부분의 라이브러리가 파이선을 지원하고 있다.	주피터 노트북(아나콘다, Google Colab), PyCharm(JetBrains, 커뮤니티 에디션), VSCODE(MS), 아툼(깃허브재단) 등

개발 언어	주요 특징	개발 도구 및 환경
자바(java)	텐서 연산(ND4J), GPU 가속 딥러닝 스택(DL4J), 자연어 처리(CoreNLP) 등 라이브러리를 사용할 수 있다. 개발, 배포, 모니터링을 위한 기존의 모든 자바 인프라를 사용할 수 있다.	eclipse, IntelliJ IDEA(커뮤니티 에디션), Netbeans, BlueJ 등
C/C++	텐서플로, 카페(Caffe) 등 라이브러리 연동이 가능하다. (예: 파이선으로 만든 모델을 C/C++를 통해 실행 환경 튜닝 등)	Visual Studio

파이선은 인터프리터 방식의 프로그래밍 언어로 TIOBE는 개발 언어 시장 점유와 관련하여 약 20년간 조사 결과를 제공하고 있는데, 2021년에는 파이선 언어가 자바(java), C++를 제치고 2위를 차지한 것으로 발표하였다.

〈표 1-4〉 개발 언어 순위(2021년 기준)

개발 언어	순위	비고
C	1	
파이선	2	장고(django) 웹 프레임워크를 통해 웹서비스 가능
자바(java)	3	
C++	4	
C#	5	
비주얼 베이식(Visual Basic)	6	
...	...	
트랜잭트(Transact)-SQL	20	

출처: TIOBE(2021년). TIOBE Index. <https://www.tiobe.com/tiobe-index/>에서 2021. 06. 15. 검색.

파이선은 수학, 과학, 공학 분야에서 필요로 하는 많은 기능을 언어 자체와 다양한 라이브러리 형태를 통해 제공한다. 넘파이(NumPy)+사이파이(SciPy)+맷플롯립(matplotlib)+판다스(Pandas) 조합은 과학, 공학 계산용으로 자주 활용되고 있다. 넘파이(NumPy), 사이파이(SciPy), 판다스(Pandas)와 같은 라이브러리는 복잡한 수치와 큰 데이터를 다루는 연산에 알맞으며, 맷플롯립(matplotlib) 라이브러리는 데이터를 그래프로 시각화하는 데 활용되는 라이브러리이다.

### 파이선의 수치 자료형 표현 범위

데이터형	비트 수	표현 범위
int	32 비트	-2,147,483,648 ~ 2,147,483,648
long	무제한(메모리 허용 한도)	무제한
float	64 비트	유효 자리 15자리, 약 10의 ±308승 (-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807)
complex	실수부 및 허수부 각각 64 비트	유효 자리 15자리, 약 10의 ±308승

파이선은 문법이 쉽고 높은 가독성 등 처음 프로그래밍을 배울 때 추천되는 언어이며 학습용으로도 적극적으로 사용되고 있고, 범용성과 높은 생산성을 가지고 있어 인공지능 분야에 많이 사용 중이므로 수행하기에는 파이선 개발 환경을 구성해 본다.

#### (2) 인공지능서비스 개발 도구

인공지능서비스 개발 도구는 로직의 구현 및 디버깅을 쉽게 진행하도록 도움을 주는 텍스트 편집기, 디버거, 컴파일러 등으로 구성된 소프트웨어이다. 이러한 개발 도구들을 통합하여 구성한 소프트웨어를 통합 개발 환경(IDE, Integrated Development Environment)이라고 하며, 최근의 다양한 개발 도구는 다양한 구성을 포함한 통합 개발 환경으로 배포되고 있다.

〈표 1-5〉 인공지능서비스 개발 도구의 구성

개발 언어	순위	비고
소스 코드 편집기	- 구문(Syntax) 강조, 메소드 리스팅, 자동 들여쓰기 등의 기능을 포함하여 소프트웨어 코드를 작성하도록 돕는 텍스트 편집기	언어별 자동 완성 기능 지원 등
디버거	- 소스 코드에 있는 버그의 위치를 그래픽으로 표시하는 등 소스 코드의 결함 여부를 테스트하는 프로그램	로직 구현 생산성 도구
컴파일러	- 문법에 어긋나는지 확인하고 기계어로 변환하는 프로그램	
각종 연계 도구	- 자동 배포 등을 위해 외부 형상, 배포 관리 기능과 연계하도록 지원하는 프로그램 - 데이터베이스 연동을 위해 JDBC, ODBC 등 연계하도록 지원하는 프로그램	대규모 DB 연계 시 꼭 필요한 구성이 될 수 있음.

최근 애플리케이션 개발 환경은 컨테이너 같은 경량화된 가상화 환경에서 구동하는 단위로 개발하고, 다수의 마이크로서비스와 통신하는 컨테이너들을 관리할 수 있는 환경으로 진화하고 있다.

### 컴파일의 개념

원시 코드(Source Code, 고급 언어)에서 목적 코드(Object Code, 기계어)로 옮기는 과정을 컴파일(compile)이라고 하며, 소스 코드를 컴파일하는 이유는 사람이 이해하는 형태의 언어로부터 실행 가능한 기계어 코드를 생성하기 위해서이다. 소스 코드를 읽어서 결과를 나타내는 인터프리터와는 구분되나, 현대에는 JIT(Just In-Time) 등의 기술에 따라 실시간 컴파일이 수행되므로 컴파일러와 인터프리터의 구분에 사실상 큰 차이가 없다. 컴파일이 필요한 언어: C/C++, Java(바이트코드로 컴파일 후 JVM을 통해 인터프리트)

인공지능서비스 개발도 이러한 마이크로서비스, 컨테이너 및 컨테이너 통합-관리 환경과 같은 오픈 소스 프레임워크 기반 아키텍처들이 많이 활용되고 있다. 특히 인공지능서비스를 개발하는 일련의 작업 흐름, 즉 컨테이너 등 환경 자동화 관리와 GPU 및 볼륨 관리, GPU 성능 모니터링, 멀티 GPU 클러스터링, 잡 스케줄링, 멀티 클라우드 등 일련의 작업을 자동화하는 클라우드 기반의 인공지능 개발 플랫폼이 나타나고 있다.

### 마이크로서비스, 컨테이너, 프레임워크의 개념

- ① 마이크로서비스: 소프트웨어 개발을 위한 아키텍처(또는 접근 방식), 애플리케이션 서비스를 위해 독립적인 최소 구성 요소로 분할하고 표준화된 방식으로 서비스
- ② 컨테이너: 애플리케이션의 논리적인 패키징 기술로 어떤 환경에서든 실행 가능 애플리케이션의 배포 목적. 컨테이너 기반 애플리케이션을 쉽게 지속적으로 배포 가능한 접근 방식
- ③ 프레임워크: 소프트웨어의 표준 구조(아키텍처)를 구현하는 클래스와 라이브러리 모임

## 3. 인공지능서비스 개발의 하드웨어 환경

인공지능 기술의 중심은 소프트웨어 기술이지만 복잡한 인공지능을 구현하기 위해서는 하드웨어(HW)의 성능을 고려해야 한다. 인공지능은 빅데이터를 바탕으로 하여 모델 부피를 키워 성능을 높이는 시스템으로 특정 연산을 수없이 계산하는데, 계산이 많으면 많을수록 오차가 줄고 시스템의 정확도는 높아지는 특성을 갖는다.

자연어 처리(NLP: Natural Language Processing)나 머신 비전과 같은 인공지능 모델에서는 프로세서 병렬 연산 속도가 얼마나 빠른지를 기준으로 성능을 구분한다. 그래픽 처리 장치(GPU: Graphics Processing Unit)는 가장 대표적인 병렬 연산 가능 프로세서이며, 현재는 GPU 외에도 신경망 처리 장치(NPU: Neural Processing Unit)나 이미지 처리 장치(IPU: Image Processing Unit) 등 다양한 병렬 연산 장치가 개발되고 있다.

### GPGPU의 유래

GPU는 원래 이름 그대로 컴퓨터 그래픽 요소를 처리하기 위해 만든 개념이다. 하지만 GPU가 일반적인 연산에도 유용하게 쓰인다는 것이 알려지면서 범용 컴퓨팅 처리를 위한 GPGPU(General-Purpose computing on GPU) 기술이 개발됐다.

GPGPU 기술이 개발되면서 수개월 걸리던 대형 신경망 네트워크 학습 시간이 짧게는 몇 시간에서 길게는 며칠로 줄어드는 등 인공지능 성능이 급격하게 좋아지게 되었다.

〈표 1-6〉 CPU와 GPU의 특징 비교

구분	CPU	GPU
설계 사상	- 다양한 환경에서 빠른 작업 수행을 위해 복잡한 구조로 설계	- 수백 개 코어를 단순 결합한 구조로 설계
특징	- 명령어 하나로 계산 여러 개를 한꺼번에 하거나 복잡한 수식 처리에 유리 - 순차적으로 계산하는 직렬 처리 기술에 초점	- 그래픽과 같은 특화된 연산을 빠른 속도로 수행 - 연산력을 높이는 데 특화 - CPU에 의해 GPU 제어(직접 작업 처리 불가)

어떤 하드웨어 환경에서 인공지능서비스 로직을 구현할 것인지는 인공지능서비스 모델의 성능에 크게 영향을 미치므로 하드웨어 환경에 대한 고려가 중요하며 인공지능 클라우드 개발 플랫폼에서 제공하는 GPU 환경도 고려할 수 있다.

#### GPU의 성능 사례

구글은 인공 신경망 구축을 위해 16,000개의 CPU 코어를 사용했다. 반면, GPU는 병렬 처리 방식으로 대규모의 계산을 빠르게 처리하는 것뿐만 아니라, 많은 코어(Core)를 집약하였기 때문에 GPU 가속화 서버 16대만으로도 구글에 비해 6.5배 확장한 인공 신경망을 구축할 수 있었다.

## 수행 내용 / 인공지능서비스 개발 환경 구성하기

### 재료·자료

- 요건 분석서 / 비즈니스 로직 설계서 / 아키텍처 정의서
- 인공지능서비스 로직 구현을 위한 개발 도구  
(인터넷 가능한 컴퓨터 환경, 설치 파일을 다운로드받아야 하는 경우 대비)
- 운영 체제 - 윈도우 10 환경

### 기기(장비·공구)

- 컴퓨터(필요한 경우 GPU 탑재)
- 문서 열람 도구(ODF 등)

#### ODF - Open Document Format for Office Application의 Spreadsheets

개방형 문서 형식은 스프레드시트, 차트, 프레젠테이션, 데이터베이스, 워드 프로세스를 비롯한 사무용 전자 문서를 위한 파일 형식으로, XML 파일 형식을 바탕으로 하여 OASIS 컨소시엄이 표준화하였다. 최근에는 상용 Office 제품군들도 ODF 포맷을 지원하고 있다.

### 안전·유의 사항

- 설계자 또는 이해관계자들과 원활한 의사소통을 할 수 있는 역량을 갖춘다.
- 비즈니스 로직 구현을 위한 설계서를 이해할 수 있어야 한다.
- 개인 컴퓨터(PC) 환경에서 GPU 기반 개발 환경을 구성할 것인지 판단한다.
- 본 수행 내용에는 CPU 기반 개발 환경을 구성하도록 정리한다. GPU 기반 개발 환경은 부록에 별도로 구성했으니 참고한다.

### 수행 순서

#### ① (개인 컴퓨터)인공지능서비스의 개발 환경을 구성한다.

##### 1. 개발 환경의 구성 요소를 확인한다.

인공지능서비스 구현을 위한 개발 환경 구성을 위한 설치 파일 구성을 하나하나 확인하고 순서에 따라 컴퓨터에 설치한다.

〈표 1-7〉 인공지능서비스 구현 환경의 구성 순서

구성 순서	설명
GPU 모델 확인	- 내 PC 확인: (윈도키+X) > 장치 관리자 > 디스플레이 어댑터 > GPU 모델 확인
및 CUDA 지원 여부 확인 (부록 1-1에 정리)	- GPU 모델이 CUDA Toolkit을 지원하는지 여부 확인 ( <a href="https://developer.nvidia.com/cuda-gpus">https://developer.nvidia.com/cuda-gpus</a> ) - GPU가 장착되어 있지 않거나 CUDA를 지원하지 않는다면 TensorFlow, CUDA, cuDNN 등 일부 모듈은 설치하지 않고 진행
아나콘다 (Anaconda) 설치	- 아나콘다 공식 사이트에 접속하여 다운로드 ( <a href="https://www.anaconda.com/products/individual#download-section">https://www.anaconda.com/products/individual#download-section</a> )
텐서플로 (TensorFlow) 설치 사전 호환성 검토 (부록 1-1에 정리)	- TensorFlow-GPU를 설치하기 위해 CUDA, cuDNN, Visual Studio와의 호환성 검토 - <a href="https://www.tensorflow.org/install/gpu#software_requirements">https://www.tensorflow.org/install/gpu#software_requirements</a> - <a href="https://www.nvidia.com/download/index.aspx?lang=en-us">https://www.nvidia.com/download/index.aspx?lang=en-us</a> - <a href="https://developer.nvidia.com/cuda-toolkit-archive">https://developer.nvidia.com/cuda-toolkit-archive</a>
Visual Studio 설치 (부록 1-1에 정리)	- LightGBM, Surprise 등 패키지 활용을 위해 Build Tools 설치 - Visual Studio 설치 버전은 아래 URL을 통해 확인 - <a href="https://docs.nvidia.com/cuda/archive/11.3.0/cuda-installation-guide-microsoft-windows/index.html">https://docs.nvidia.com/cuda/archive/11.3.0/cuda-installation-guide-microsoft-windows/index.html</a> - <a href="https://visualstudio.microsoft.com/ko/downloads/">https://visualstudio.microsoft.com/ko/downloads/</a>
CUDA 설치 (부록 1-1에 정리)	- <a href="https://developer.nvidia.com/cuda-toolkit-archive">https://developer.nvidia.com/cuda-toolkit-archive</a>
cuDNN 설치 (부록 1-1에 정리)	- <a href="https://developer.nvidia.com/cudnn">https://developer.nvidia.com/cudnn</a>
가상 개발 환경 만들기/접속	- 프로젝트별 파이선 라이브러리 사용 여부 확인 - 파이선 사용 버전이 다를 경우 충돌 방지를 위한 가상 개발 환경 구성 - jupyter notebook, numpy, scipy, spyder, pandas, seaborn, scikit-learn, h5py, pillow, tqdm 등 설치
텐서플로 및 기타 라이브러리 설치	- tensorflow-gpu 설치(tensorflow 2버전부터는 keras 내장) - xgboost, catboost, lightgbm, pydotplus, pydot, scikit-image, patsy, statsmodels, opencv 등 설치
설치 환경 테스트	- tensorflow-gpu 버전 설치 여부 확인

**쿠다(CUDA, Compute Unified Device Architecture), cuDNN(CUDA® Deep Neural Network library)**

CUDA - GPU에서 수행하는 병렬 처리 알고리즘을 C언어를 포함하여 산업 표준의 다양한 언어를 활용하여 구현 가능하도록 지원하는 GPGPU 기술이다. CUDA는 엔비디아(nVidia)가 개발해오고 있으며 CUDA 아키텍처를 사용하려면 엔비디아 GPU와 스트림 처리 드라이버가 필요하다. CUDA는 G8X GPU로 구성된 Geforce 8 Series 이상에서 동작하며, 커널 실행을 위해 GPU의 가상 명령 집합과 병렬 연산 요소들을 직접(direct) 접근이 가능한 소프트웨어 계층이다.

cuDNN - CUDA 딥 뉴럴 네트워크의 약자이며, 라이브러리 집합이다. GPU 가속화 라이브러리 기반의 딥 뉴럴 네트워크 구성을 위한 컨볼루션(Convolution), 풀링(Pooling), 표준화(Normalization), 활성화(Activation)와 같은 DNN 요소들의 일반적인 루틴을 빠르게 이행할 수 있도록 지원하는 라이브러리이다.

- (1) GPU 모델 확인 및 CUDA 지원 여부를 확인한다(GPU가 장착되지 않은 경우 (2)번 진행).  
인공지능서비스 개발 환경 수행하기에는 CPU 기반 구현 환경을 구성하므로 GPU 기반 개발 환경 구성하기 내용은 부록의 1-1번을 참고한다.
- (2) 아나콘다를 설치한다.

아나콘다(Anaconda)는 머신러닝(딥러닝)뿐만 아니라 과학 계산을 위한 용도로도 유용하게 활용되는 파이선 기반의 범용 패키지이다. 아나콘다 대신 파이선을 독립적으로 직접 설치할 수 있지만 각 패키지들의 호환성, 의존성 관리에도 효과적인 아나콘다를 설치하여 패키지들을 관리할 수 있도록 한다.

**pip, conda: 패키지 관리 프로그램**

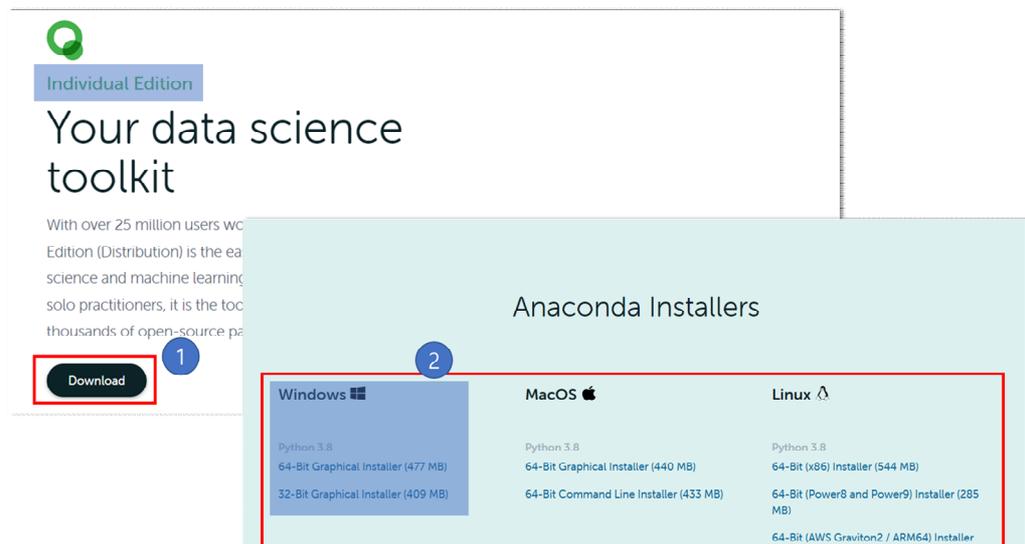
pip: 파이선으로 작성된 패키지 소프트웨어를 설치·관리하는 패키지 관리 시스템이다.  
파이선에서 필요로 하는 패키지를 pip 명령어 실행으로 설치한다.

예제) 패키지 설치 → pip install 패키지\_이름  
패키지 삭제 → pip uninstall 패키지\_이름

conda: 파이선, R, Ruby, Lua, Scala, Java, Javascript, C/C++, FORTRAN 등 많은 언어의 의존성, 패키지, 환경을 관리

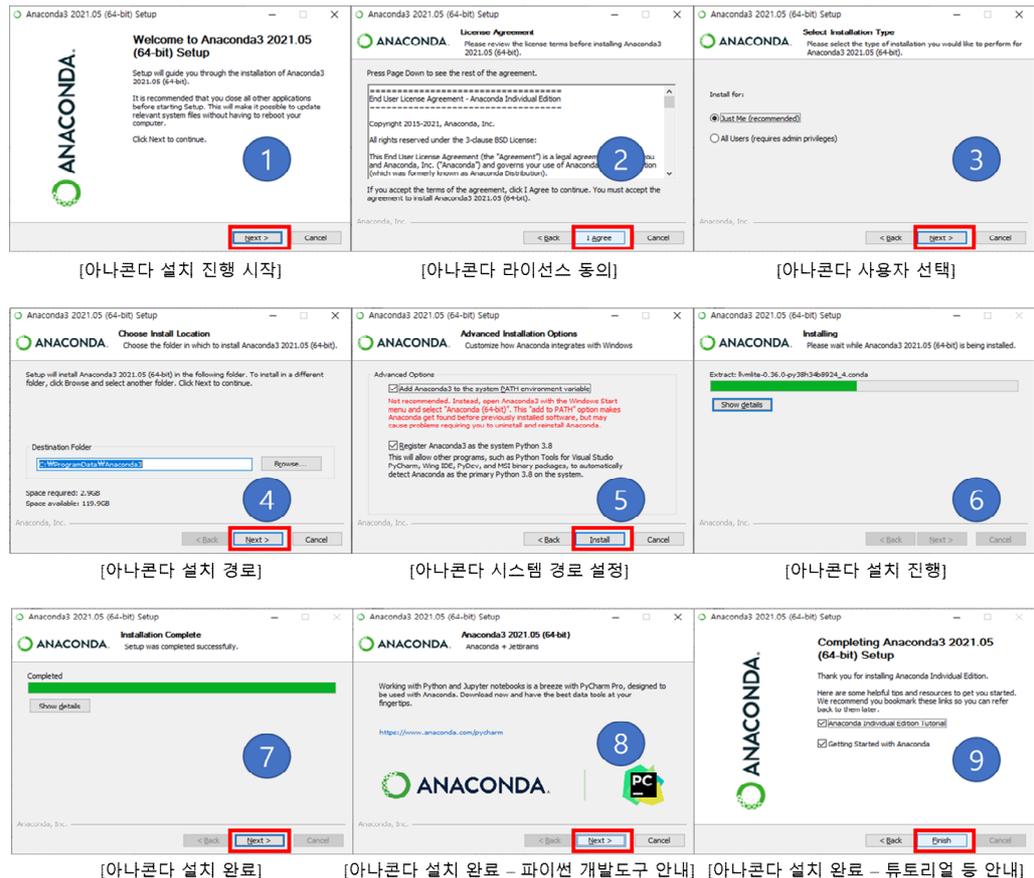
TensorFlow는 윈도우에서 아나콘다 배포판을 설치할 것을 공식적으로 권장한다.

아나콘다 공식 사이트(<https://www.anaconda.com/products/individual#download-section>)에 접속하여 설치 파일을 다운로드한다. Download 버튼(①)을 누르면 해당 사이트의 하단으로 이동하고(②) 개인 컴퓨터 환경(OS 및 32/64-Bit)에 따라 일치하는 버전을 다운로드받는다.



출처: 집필진 제작(2021)  
[그림 1-2] 아나콘다 다운로드 과정 ① ~ ②

다운로드가 완료되면 아나콘다 설치 파일을 더블 클릭한다. 더블 클릭하면 [그림 1-3]과 같은 각 단계를 진행하게 되는데 잘 따라 하면 큰 어려움 없이 설치를 완료할 수 있다.



출처: 집필진 제작(2021)  
[그림 1-3] 아나콘다 설치 과정 ① ~ ⑨(Anaconda Individual Edition)

(가) 아나콘다 라이선스 종류에 대해 참고한다.

설치 과정 ②번은 아나콘다 소프트웨어의 라이선스를 설명하고 있으며, 동의(I Agree)를 클릭해야 설치를 진행할 수 있다. 참고로 아나콘다 소프트웨어 라이선스(New BSD Licenses)는 ‘아무나 개작’할 수 있고, ‘수정한 것을 제한 없이 배포’가 가능한 오픈 소스 라이선스이다.

(나) 시스템 사용 환경을 설정할 때 고려 사항을 참고한다.

설치 과정 ③은 아나콘다 사용 환경을 의미하는 것으로 해당 개인 컴퓨터의 모든 사용자가 아나콘다 패키지를 사용할 수 있도록 하려면 “All Users”를 선택하고 다음으로 진행한다.

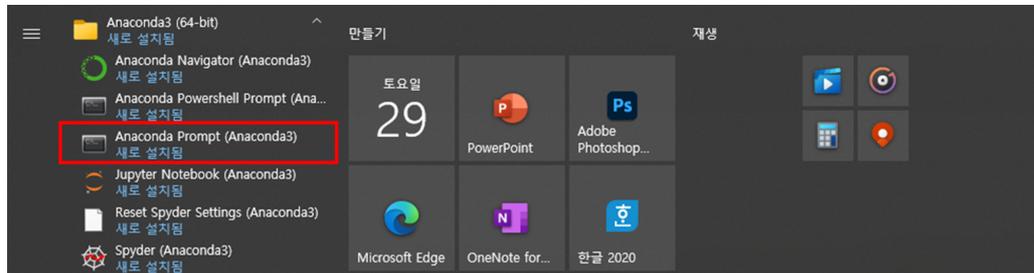
(다) 시스템 환경 변수를 체크할 때 고려 사항을 참고한다.

설치 과정 ⑤는 시스템 환경 변수에 경로를 추가할 것인지에 대한 사항으로 체크를 하고 다음으로 진행한다.

시스템 환경 변수에 경로를 추가하지 않으면 아나콘다의 다양한 실행 프로그램이 윈도 명령 프롬프트(cmd.exe)에서 실행되지 않게 된다. 아나콘다 설치가 잘되었는지 확인을 위해 [시작 버튼 > Anaconda > Anaconda Prompt]를 클릭한다.

(3) 아나콘다를 설치 완료한 후 정상 여부를 확인한다.

Anaconda Prompt를 실행하여 커서가 깜빡이는 곳에 “python”을 타이핑하고 엔터(키보드) 후 결과에 파이썬(python)의 버전이 나타난다면 설치가 정상적으로 완료되었다는 것이다. Ctrl + z 명령어를 통해 python 프롬프트(>>>))에서 빠져나온 후 conda --version을 통해 아나콘다의 버전도 확인한다.



출처: 집필진 제작(2021)

[그림 1-4] 아나콘다 설치 결과 - Anaconda Prompt ([시작 버튼 > Anaconda])

#### 아나콘다 최신 버전 업데이터 방법

Anaconda Prompt를 관리자 권한으로 실행 후 다음 명령어를 입력한다.

```
conda update -n base -c defaults conda # 아나콘다 업그레이드  
conda update --all # 파이썬 패키지 업그레이드
```

(4) 텐서플로 설치를 위해 호환성을 검토한다(GPU가 장착되지 않은 경우 (6) 이동).

다음의 절차는 GPU 환경을 위한 설치 과정으로 상세 내용은 부록의 1-1번을 참고한다.

(가) GPU(NVIDIA) 드라이버를 설치한다.

(나) CUDA Toolkit을 확인한다.

(다) Visual Studio 2019(16.x)를 설치한다.

(5) CUDA 및 cuDNN 모듈을 설치한다.

(가) CUDA Toolkit을 설치한다.

(나) cuDNN을 설치한다.

(6) 가상 개발 환경을 만들고 접속한다.

지금까지 개발 환경 구성을 위한 사전 환경 설정을 완료하였다. 텐서플로 등의 패키지 설치를 위해 가상 개발 환경을 구성한다.

#### 가상 개발 환경 구성 필요성

파이선 프로젝트별로 사용하는 라이브러리의 버전이 다를 경우 문제 해결에 어려움이 발생할 수 있다. 따라서 프로젝트별로 사용하는 라이브러리 버전 간의 충돌 방지를 위해서 가상 개발 환경을 구성하고 프로젝트 환경을 구성한다.

아나콘다 프롬프트(시작 버튼 → Anaconda → Anaconda Prompt)를 클릭한 후, 다음과 같이 가상 환경용 폴더(v\_aincs\_py38)를 생성하고, conda도 최신버전으로 업데이트한다.

#### 가상 환경 생성을 위한 기초 명령어(프로젝트 폴더 생성 등)

```
(base) C:\ncs_prj>mkdir v_aincs_py38
(base) C:\ncs_prj>cd v_aincs_py38
(base) C:\ncs_prj\v_aincs_py38>
(base) C:\ncs_prj\v_aincs_py38>conda update -n base -c defaults conda
```

계속해서 새로운 가상 환경(예제: v\_aincs\_py38)을 생성한다.

#### 가상 환경 생성 및 접속 명령어

```
===== 가상 환경 생성 명령어 =====
(base) C:\ncs_prj\v_aincs_py38>conda create -n v_aincs_py38 python=3.8
===== 가상 환경에 설치 과정 진행 =====
===== 가상 환경 접속 명령어 =====
(base) C:\ncs_prj\v_aincs_py38>conda activate v_aincs_py38
(v_aincs_py38) C:\ncs_prj\v_aincs_py38>
```

위와 같이 activate 명령어를 통해 가상 환경에 접속하면 프롬프트 앞이 (base)에서 (v\_aincs\_py38)로 변경되는 것을 확인할 수 있다. 가상 환경에 구성해야 할 텐서플로(버전 주의)와 필요한 라이브러리(부록 1-1 참고)를 설치한다.

(7) 텐서플로 및 기타 라이브러리를 설치한다.

(가) 주피터 노트북을 설치한다.

텐서플로를 설치하기 전에 주피터 노트북과 머신러닝, 딥러닝에 자주 활용되는 라이브러리를 먼저 설치한다.

### 주피터 노트북(jupyter notebook)이란?

인터랙티브 형식의 웹 베이스 애플리케이션으로 라이브 코드를 제공한다. 웹 브라우저 안에서 파이썬 코드를 입력하고 실행 버튼을 통해 결과를 바로 확인할 수 있다.

주피터 노트북 사용(웹 브라우저 인터페이스)을 통해 수학 공식의 표현과 내레이션 텍스트나 다양한 그래프를 생성한 데이터 시각화, 이미지 등을 추가하여 노트북 문서를 만들고 공통 작업을 위한 공유 환경으로 사용할 수 있다.

주피터 노트북은 파이썬뿐만 아니라 R, Go, C++, Julia, Java 등 다양한 종류의 개발 언어를 지원한다.

#### 주피터 노트북 설치 명령어(C:\ncs\_prj\v\_aincs\_py38) ← 생략

```
(v_aincs_py38) conda install -n v_aincs_py38 ipython notebook jupyter
```

(나) 머신러닝이나 딥러닝을 개발할 때 활용되는 라이브러리를 설치한다.

머신러닝, 딥러닝에 활용되는 관련 라이브러리의 설치 방법은 부록 1-1을 참고한다.

(다) 인공지능 프레임워크인 텐서플로 및 케라스를 설치한다.

텐서플로와 케라스는 인공지능 프레임워크이다. 인공지능 모델을 쉽게 개발하기 위해 프레임워크를 설치한다. 텐서플로 2 버전부터는 케라스가 내장되어 있으므로 별도의 설치가 필요 없지만, 이전 버전으로 설치한 경우는 설치하도록 한다.

#### 텐서플로 및 케라스 설치 명령어(C:\ncs\_prj\v\_aincs\_py38) ← 생략

```
(v_aincs_py38) conda install -n v_aincs_py38 tensorflow #GPU 모델이 아닌 경우
```

```
(v_aincs_py38) pip install tensorflow-gpu #GPU 모델인 경우
```

```
(v_aincs_py38) conda install -n v_aincs_py38 keras # 설치가 필요한 경우
```

(라) 기타 라이브러리를 설치한다.

머신러닝 앙상블 기법으로 사용되는 xgboost, catboost, lightgbm 라이브러리 등 관련 라이브러리의 설치 방법은 부록 1-1을 참고한다.

(8) 인공지능서비스 개발 환경의 구성이 완료되었는지 테스트한다.

지금까지 설치한 개발 환경이 잘 동작하는지 테스트한다. 아나콘다 프롬프트에서 가상 환경에 접속한 후 주피터 노트북에 접속한다.

**주피터 노트북 접속 명령어(C:\ncs\_prj\v\_aincs\_py38) ← 생략)**

(base) **activate v\_aincs\_py38**

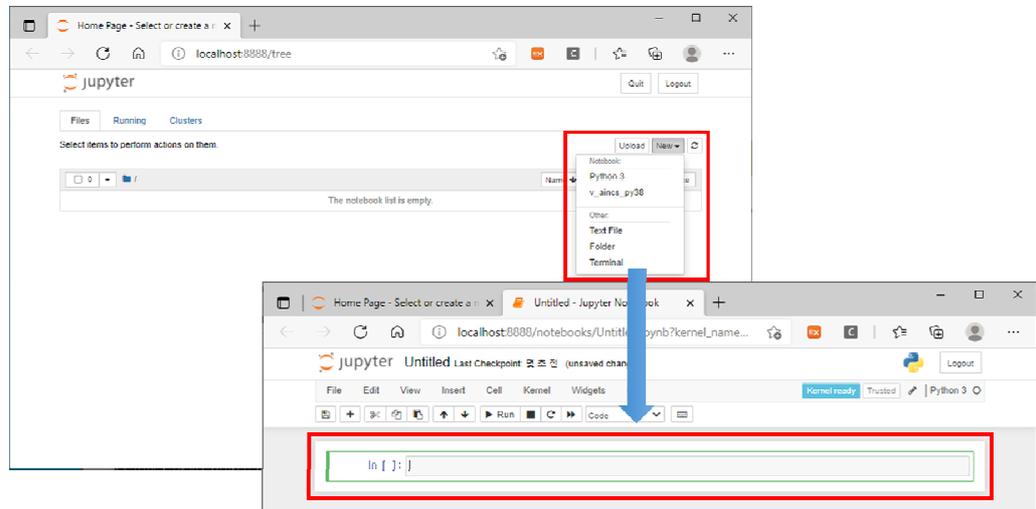
(v\_aincs\_py38) **jupyter notebook**

※ 다중 커널 실행 모드 설정

다중 커널은 아나콘다 기본 편집기인 주피터 노트북에서 기본 파이선 3 버전 설치 시 노트북 생성 파일에 파이선 3 버전 외에도 2 버전 등을 같이 사용할 수 있는 독립적인 개발 환경을 의미한다.

(base) **python -m ipykernel install --user --name v\_aincs\_py38**

주피터 노트북이 <http://localhost:8888/tree> 주소로 실행되는 것을 확인한다. 오른쪽 상단의 New 버튼을 클릭한 후 v\_aincs\_py38(또는 python3)을 선택하면 [그림 1-5]와 같이 파이선 코드를 입력할 수 있는 새로운 창이 나타난다.



출처: 집필진 제작(2021)

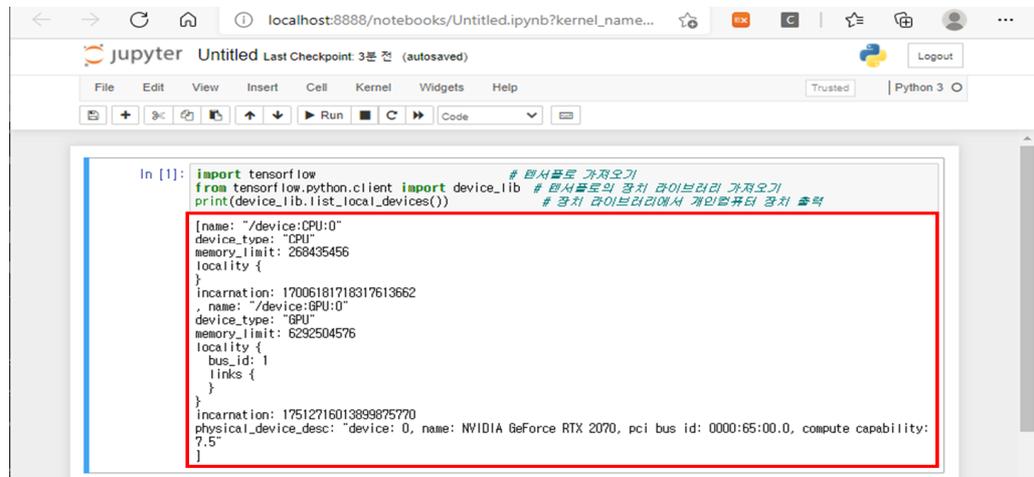
[그림 1-5] 주피터 노트북 접속 화면

주피터 노트북 새로운 창의 빈 셀에 다음의 파이선 코드를 입력한 후 “Run(Shift + Enter)”을 클릭해 본다.

**주피터 노트북 빈 셀에 파이선 코드 입력 후 실행**

```
import tensorflow                # 텐서플로 가져오기
from tensorflow.python.client import device_lib # 텐서플로의 장치 라이브러리 가져오기
print(device_lib.list_local_devices())          # 장치 라이브러리에서 개인 컴퓨터 장치 출력
```

“Run(단축키 Shift + Enter)”을 클릭하면 [그림 1-6]과 같은(GPU 기반 개발 환경을 설치했다면 GPU 정보를 확인) 정보를 확인할 수 있다. 집필진은 GPU 기반 개발 환경을 구성했기 때문에 tensorflow-gpu의 지원 정보로 compute capability는 7.5, 메모리는 약 6GB, GPU는 1개 등의 정보가 나타났다.



출처: 집필진 제작(2021)  
 [그림 1-6] 주피터 노트북을 통해 텐서플로 GPU 가용 정보 확인

### 설치 라이브러리들의 버전 확인

```
import tensorflow
from tensorflow import keras
import pandas, sklearn, scipy, numpy, matplotlib, pydotplus, pydot, h5py
print('tensorflow 버전: ' + tensorflow.__version__)
print('keras 버전: ' + keras.__version__)
print('sklearn 버전: ' + sklearn.__version__)
print('pandas 버전: ' + pandas.__version__)
print('h5py 버전: ' + h5py.__version__)
print('numpy 버전: ' + numpy.__version__)
print('scipy 버전: ' + scipy.__version__)
print('matplotlib 버전: ' + matplotlib.__version__)
```

### 수행 tip

- 가상 환경에 개발 환경을 구성할 때 'conda install' 명령어가 개인 컴퓨터의 환경, 버전 등의 이유로 오류가 나타날 경우 해당 라이브러리와 관련된 공식 문서를 검토할 필요가 있다.

# 1-2. 비즈니스 이해 및 로직 구현

## 학습 목표

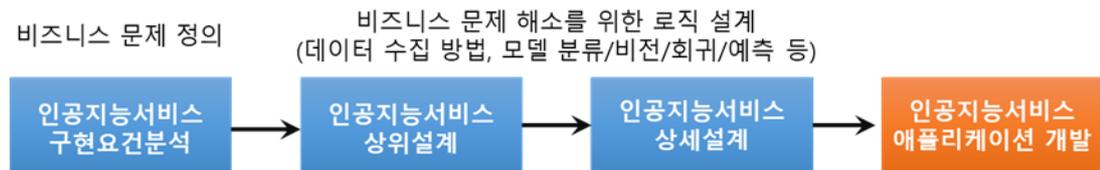
- 인공지능서비스 애플리케이션 설계에 따라 인공지능 비즈니스 로직을 구현할 수 있다.

## 필요 지식 /

### ① 비즈니스의 이해 및 설계 내용의 파악

인공지능서비스를 비즈니스에 적용하기 위해서는 먼저 비즈니스의 어떤 문제를 인공지능서비스를 통해 해결할 것인지 정의하는 것이 필요하다. 문제 정의가 되었다면 비즈니스 문제 정의를 바탕으로 하여 왜곡되지 않고 편향적이지 않으며, 다양성을 확보한 데이터를 정확하게 수집, 축적한 후 인공지능서비스를 지속적으로 학습 및 검증 과정을 통해 성능을 향상시킨다.

정보 통신 > 정보 기술 > 인공지능의 ‘인공지능서비스 구현’ 학습 모듈 중에서 ‘인공지능서비스 구현 요건 분석’을 통해 정의된 비즈니스 문제 및 요건을 파악하고, ‘인공지능서비스 상위 설계 및 상세 설계’를 통해 로직 구현을 위한 설계 내용을 파악한다.



출처: 집필진 제작(2021)

[그림 1-7] 인공지능서비스 구현 상위 학습모듈과 연계도

### ② 인공지능서비스 기반 비즈니스 로직의 구현

#### 1. 인공지능서비스 로직의 구현 절차

인공지능서비스 로직의 구현 절차는 산업계에서는 다양하게 정의되고 있으나, 소프트웨어 개발 절차(분석 → 설계 → 구현 → 테스트 → 배포)와 같이 체계화되어 있지는 않다. 다음은 다양하게 활용 중인 인공지능서비스 구현 절차를 예시하였다.

<표 1-8> 인공지능서비스 로직 구현 절차별 설명

절차명	설명
CRISP-DM	- Cross Industry Standard Process for Data Mining - 널리 사용되고 있는 공개된 표준 분석 모델 - 6단계로 구성되어 있으며 각 단계 간 피드백을 통해 완성도를 높이는 방식으로 구성 - CRISP-DM 절차

절차명	설명		
	단계	설명	
	Business Understanding (비즈니스 이해)	<ul style="list-style-type: none"> <li>- 각종 참고 자료를 현업 책임자와의 커뮤니케이션을 통해 비즈니스를 이해</li> <li>- 업무 목적, 데이터 마이닝 목표 설정, 상황 파악, 프로젝트 계획 수립 등</li> </ul>	
	Data Understanding (데이터 이해)	<ul style="list-style-type: none"> <li>- 분석을 위한 데이터 수집 및 속성 이해</li> <li>- 문제점 식별 및 숨겨진 인사이트(Insight)의 적극적인 발견</li> <li>- 초기 데이터 수집, 데이터 기술 분석(Data Descriptive Analytics), 데이터 탐색 및 데이터 품질 확인 등</li> </ul>	
	Data Preparation (데이터 준비)	<ul style="list-style-type: none"> <li>- 데이터 정제, 데이터 생성 등 분석 가능한 자료로 만드는 단계</li> <li>- 분석 용도의 데이터 세트 선택, 분석을 위한 사전 데이터 정제 및 데이터 통합, 학습/테스트/검증용 데이터 분리 등</li> <li>- 다양한 분석 모델링 기법과 최적의 알고리즘을 선택하며 관련 파라미터의 최적화</li> </ul>	
	Modeling (모델링)	<ul style="list-style-type: none"> <li>- 모델의 테스트 계획(테스트/검증 세트), 모델의 작성 및 평가 등</li> <li>- 모델링 결과는 검증 용도의 데이터를 이용하여 평가하며, 과적합(Over-Fitting) 여부에 대해 추가 확인</li> </ul>	
	Evaluation(평가)	<ul style="list-style-type: none"> <li>- 모델의 해석 가능 여부 및 분석 결과가 프로젝트의 목적에 타당인지 평가, 결과 수용 여부 판단</li> <li>- 모델링의 과정 평가, 분석 결과에 대한 평가, 모델의 적용성 평가</li> </ul>	
	Deployment (전개)		<ul style="list-style-type: none"> <li>- 모델링 및 평가 단계 수행 결과에 따라 생성된 모델을 대상(업무 등)에 적용하기 위한 계획을 수립</li> </ul>
			<ul style="list-style-type: none"> <li>- 모델의 모니터링 등 유지를 위한 계획 마련과 관련 담당자 간 모델 검토 진행 및 검토가 완료된 모형을 대상 업무에 적용</li> <li>- 모델의 전개를 위한 계획 수립, 모니터링 등 유지·보수 계획 수립, 프로젝트 리뷰 등</li> </ul>
	<ul style="list-style-type: none"> <li>- AI 모델의 생명 주기에 해당하는 전체 단계를 자동화 시스템으로 구현하는 경우에 대해 전체 10개의 기술적으로 고려해야 할 요소로 구성된 절차를 수행</li> <li>- 알고리즘, 프레임워크, 기타 기술 영역으로 구분되고, 각 영역은 상호 간에 밀접하게 상호 작용</li> </ul>		
	AI Life Cycle 개발 자동화	단계	설명
		데이터 전처리	- 데이터 수집과 검증, 전처리, 생성 등 학습 데이터 처리
Trainer		- 프레임워크(예: TensorFlow, 등)	
DL/ML Model Code		- 알고리즘을 구현하는 코드(Parameter)	
Model Evaluation & Validation		- 모델 평가 및 선택(Hyper-parameter)	
Model management		- 학습된 모델 Lifecycle 관리, 배포, 재학습 관리	
Serving		- 모델의 추론(Inference) 서비스(속도, 동시성, 자원 효율화)	
Distributed Training/ Job Orchestration		- 알고리즘의 학습 분산과 Job에 대한 관리	
Job management/	- Job 관리, 모니터링, 시각화 서비스와 UI		

절차명	설명										
	<table border="1"> <thead> <tr> <th>단계</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>Monitoring/Visualization</td> <td></td> </tr> <tr> <td>Machine Resource Management</td> <td>- 학습/추론을 위한 GPU 등 서버 리소스 할당, 관리</td> </tr> <tr> <td>Pipeline Storage</td> <td>- 파이프라인에 따른 학습·추론 데이터 관리</td> </tr> </tbody> </table>	단계	설명	Monitoring/Visualization		Machine Resource Management	- 학습/추론을 위한 GPU 등 서버 리소스 할당, 관리	Pipeline Storage	- 파이프라인에 따른 학습·추론 데이터 관리		
단계	설명										
Monitoring/Visualization											
Machine Resource Management	- 학습/추론을 위한 GPU 등 서버 리소스 할당, 관리										
Pipeline Storage	- 파이프라인에 따른 학습·추론 데이터 관리										
	- 주요 단계로 프로젝트의 계획 수립과 데이터에 대한 이해 단계, 데이터 준비(Data pipeline) 단계, 모델 개발과 모델의 배포(Model pipeline) 단계 3단계로 구분										
	<table border="1"> <thead> <tr> <th>단계</th> <th>설명</th> </tr> </thead> <tbody> <tr> <td>프로젝트 계획 수립 단계</td> <td> <ul style="list-style-type: none"> <li>- 인공지능이 변화시킬 대상 영역을 종합적이고 충분히 고려하여 빅픽처(Big-Picture)의 설계 및 우선 추진 과제 선정</li> <li>- (후보 과제 발굴) 어떠한 업무에 인공지능이 적합한가, 어떻게 인공지능을 적용할 것인가, 적용함으로써 어떤 것이 좋아지는가를 중심으로 검토한 후 후보 인공지능 과제를 도출(What → How → Why)</li> <li>- 인공지능을 통한 예측 대상 비즈니스 변수(모델 대상) 식별 방법으로 일반적인 5개의 질문으로 적절한 해답을 검토한 후 선정</li> <li>- (회귀) 계량 지표, (분류) 범주, (클러스터링) 그룹, (이상 감지) 정상 주문이 아닐 가능성, (추천) 사용자의 수행 옵션 등</li> <li>- 선택 질문의 결괏값에 대한 측정 방법으로 고객 이탈에 대한 예측 정확도(95% 이상) 등 SMART(Specific, Measurable, Achievable, Realistic, Time-bound) 기법 고려</li> </ul> </td> </tr> <tr> <td>AI 개발 Pipeline</td> <td> <ul style="list-style-type: none"> <li>- 분석 대상 데이터의 획득 전략 수립, 데이터의 정제, 전환(가공), 레이블링, 분석 등 전처리 단계</li> <li>- 전체 개발 단계 중에서 가장 많은 자원을 투입(약 67%)</li> </ul> </td> </tr> <tr> <td>Data Pipeline</td> <td>- (What) 어떤 데이터를, (How) 어떤 방법으로, (Why) 어떤 목적 달성을 위해 획득하는가? 등에 대해 13개(비즈니스 과제, 데이터 과학 과제, 데이터 엔지니어링 과제, 데이터 수집 방법, 획득 모델, 데이터 액세스, 데이터 대상, 유형, 포맷, 범위, 확보 주기, 원천, 경쟁) 세부 차원을 중심으로 검토</td> </tr> <tr> <td>ML Pipeline</td> <td> <ul style="list-style-type: none"> <li>- 전처리 데이터를 이용한 모델 개발과 테스트 등 운영 시스템에 사용 가능하도록 배포하는 단계</li> <li>- 텐서플로 등 프레임워크는 모델 분석기(사용자 등)가 모델 개발부터 배포까지 파이프라인 전 단계를 쉽게 이용하도록 자동화한 툴킷</li> <li>- 수집→준비→분석→훈련→평가→하이퍼튜닝→배포→실험→관리</li> <li>- (모델 개발) 모델 매개 변수(하이퍼파라미터)를 변화시켜 성능이 가장 좋은 모델을 찾는 과정</li> <li>- 학습용 데이터 세트와 적절한 알고리즘의 선택이 필수</li> <li>- 모델의 개발과 테스트는 모델의 성능 향상을 위해 반복하여 수행</li> <li>- 학습이 완료된 모델들의 성능 비교 후 가장 좋은 모델을 선택</li> </ul> </td> </tr> </tbody> </table>	단계	설명	프로젝트 계획 수립 단계	<ul style="list-style-type: none"> <li>- 인공지능이 변화시킬 대상 영역을 종합적이고 충분히 고려하여 빅픽처(Big-Picture)의 설계 및 우선 추진 과제 선정</li> <li>- (후보 과제 발굴) 어떠한 업무에 인공지능이 적합한가, 어떻게 인공지능을 적용할 것인가, 적용함으로써 어떤 것이 좋아지는가를 중심으로 검토한 후 후보 인공지능 과제를 도출(What → How → Why)</li> <li>- 인공지능을 통한 예측 대상 비즈니스 변수(모델 대상) 식별 방법으로 일반적인 5개의 질문으로 적절한 해답을 검토한 후 선정</li> <li>- (회귀) 계량 지표, (분류) 범주, (클러스터링) 그룹, (이상 감지) 정상 주문이 아닐 가능성, (추천) 사용자의 수행 옵션 등</li> <li>- 선택 질문의 결괏값에 대한 측정 방법으로 고객 이탈에 대한 예측 정확도(95% 이상) 등 SMART(Specific, Measurable, Achievable, Realistic, Time-bound) 기법 고려</li> </ul>	AI 개발 Pipeline	<ul style="list-style-type: none"> <li>- 분석 대상 데이터의 획득 전략 수립, 데이터의 정제, 전환(가공), 레이블링, 분석 등 전처리 단계</li> <li>- 전체 개발 단계 중에서 가장 많은 자원을 투입(약 67%)</li> </ul>	Data Pipeline	- (What) 어떤 데이터를, (How) 어떤 방법으로, (Why) 어떤 목적 달성을 위해 획득하는가? 등에 대해 13개(비즈니스 과제, 데이터 과학 과제, 데이터 엔지니어링 과제, 데이터 수집 방법, 획득 모델, 데이터 액세스, 데이터 대상, 유형, 포맷, 범위, 확보 주기, 원천, 경쟁) 세부 차원을 중심으로 검토	ML Pipeline	<ul style="list-style-type: none"> <li>- 전처리 데이터를 이용한 모델 개발과 테스트 등 운영 시스템에 사용 가능하도록 배포하는 단계</li> <li>- 텐서플로 등 프레임워크는 모델 분석기(사용자 등)가 모델 개발부터 배포까지 파이프라인 전 단계를 쉽게 이용하도록 자동화한 툴킷</li> <li>- 수집→준비→분석→훈련→평가→하이퍼튜닝→배포→실험→관리</li> <li>- (모델 개발) 모델 매개 변수(하이퍼파라미터)를 변화시켜 성능이 가장 좋은 모델을 찾는 과정</li> <li>- 학습용 데이터 세트와 적절한 알고리즘의 선택이 필수</li> <li>- 모델의 개발과 테스트는 모델의 성능 향상을 위해 반복하여 수행</li> <li>- 학습이 완료된 모델들의 성능 비교 후 가장 좋은 모델을 선택</li> </ul>
단계	설명										
프로젝트 계획 수립 단계	<ul style="list-style-type: none"> <li>- 인공지능이 변화시킬 대상 영역을 종합적이고 충분히 고려하여 빅픽처(Big-Picture)의 설계 및 우선 추진 과제 선정</li> <li>- (후보 과제 발굴) 어떠한 업무에 인공지능이 적합한가, 어떻게 인공지능을 적용할 것인가, 적용함으로써 어떤 것이 좋아지는가를 중심으로 검토한 후 후보 인공지능 과제를 도출(What → How → Why)</li> <li>- 인공지능을 통한 예측 대상 비즈니스 변수(모델 대상) 식별 방법으로 일반적인 5개의 질문으로 적절한 해답을 검토한 후 선정</li> <li>- (회귀) 계량 지표, (분류) 범주, (클러스터링) 그룹, (이상 감지) 정상 주문이 아닐 가능성, (추천) 사용자의 수행 옵션 등</li> <li>- 선택 질문의 결괏값에 대한 측정 방법으로 고객 이탈에 대한 예측 정확도(95% 이상) 등 SMART(Specific, Measurable, Achievable, Realistic, Time-bound) 기법 고려</li> </ul>										
AI 개발 Pipeline	<ul style="list-style-type: none"> <li>- 분석 대상 데이터의 획득 전략 수립, 데이터의 정제, 전환(가공), 레이블링, 분석 등 전처리 단계</li> <li>- 전체 개발 단계 중에서 가장 많은 자원을 투입(약 67%)</li> </ul>										
Data Pipeline	- (What) 어떤 데이터를, (How) 어떤 방법으로, (Why) 어떤 목적 달성을 위해 획득하는가? 등에 대해 13개(비즈니스 과제, 데이터 과학 과제, 데이터 엔지니어링 과제, 데이터 수집 방법, 획득 모델, 데이터 액세스, 데이터 대상, 유형, 포맷, 범위, 확보 주기, 원천, 경쟁) 세부 차원을 중심으로 검토										
ML Pipeline	<ul style="list-style-type: none"> <li>- 전처리 데이터를 이용한 모델 개발과 테스트 등 운영 시스템에 사용 가능하도록 배포하는 단계</li> <li>- 텐서플로 등 프레임워크는 모델 분석기(사용자 등)가 모델 개발부터 배포까지 파이프라인 전 단계를 쉽게 이용하도록 자동화한 툴킷</li> <li>- 수집→준비→분석→훈련→평가→하이퍼튜닝→배포→실험→관리</li> <li>- (모델 개발) 모델 매개 변수(하이퍼파라미터)를 변화시켜 성능이 가장 좋은 모델을 찾는 과정</li> <li>- 학습용 데이터 세트와 적절한 알고리즘의 선택이 필수</li> <li>- 모델의 개발과 테스트는 모델의 성능 향상을 위해 반복하여 수행</li> <li>- 학습이 완료된 모델들의 성능 비교 후 가장 좋은 모델을 선택</li> </ul>										

인공지능서비스 로직 구현을 위한 가장 우선적인 조건은 명확한 비즈니스 목표에 따

른 충분한 데이터의 확보이다. 이러한 우선 조건이 완료된 이후 어떤 인공지능 모델을 채택할 것인지 아니면 새로운 인공지능 모델을 개발할 것인지, 인공지능 모델을 변형할 것인지, 전이 학습을 할 것인지 결정한다.

### 전이 학습(Transfer Learning)

이미지넷과 같이 아주 큰 데이터 세트에 훈련된 모델의 가중값을 가지고 와서 비즈니스가 해결하고자 하는 과제에 맞게 재보정해서 사용하는 것을 의미한다.

전이 학습은 비교적 수량이 적은 데이터를 이용하고도 우리가 원하는 비즈니스 목표를 해결할 수 있는 딥러닝 모델의 훈련 접근 방법이다.

인공지능서비스 로직 구현은 식별된 인공지능 모델을 연계하여 비즈니스 목표에 적합한 서비스를 구현하는 것으로 설계 내용을 고려하여 공통 모듈 구현, 비즈니스 로직 구현 및 테스트 과정을 진행한다.

## 2. 인공지능서비스 유형

인공지능서비스 유형은 크게 시각 지능, 언어 지능으로 분류할 수 있다. 시각 지능 분야는 이미지, 영상 등의 데이터를 인식하고 상황 판단, 데이터 가공 및 새로운 이미지/영상을 생성하는 인공지능서비스 기술 분야이다.

언어 지능 분야는 인간이 사용하는 일상적 언어 이해, 대화를 목표로 자연어 처리(NLP, Natural Language Processing) 중심의 기술로 발전하고 있다. 각 기술 분류에 대해 상호 작용, 추론 및 실행 부분을 <표 1-9>에 정리하였다.

<표 1-9> 인공지능서비스 유형

분류	기술 영역	설명
시각 지능	상호 작용	• 이미지/영상에서 객체 식별을 목적으로 인공지능서비스 개발
	추론	• 객체 인식 기반의 대상 이미지에 대한 상황을 판단
	실행	• 상황 판단에 근거하여 영상 및 이미지를 조작하는 기술
언어 지능	상호 작용	• 음성, 텍스트, 이미지의 상호 작용으로 아날로그 신호로 구성된 인간의 목소리를 디지털 기반의 신호로 변환하여 처리
		• 음성 신호의 주기성(Cycle)과 동기성(synchronous)을 추출한 후 의미 있는 주파수 영역을 분리하는 방식 사용
	추론	• 텍스트 상호 작용 방식은 챗봇 등에서 활용하는 방식으로, 사용자가 입력한 디지털 텍스트를 통해 상호 작용 • 이미지 상호 작용은 사진 및 동영상 등을 분석하여 텍스트에 해당하는 영역을 추출하는 방식으로 상호 작용 • 형태소 분석 방법은 형태소 사전을 활용하는 '형태소 후보 분석' 방법과 규칙이나 기계 학습을 활용하는 '품사 부착(Part of Speech Tagging)' 방법 이용 • 입력 문장은 음절 단위의 분석과 띄어쓰기, 형태소 사전을 각각 참조하여 형태소별 품사를 부착하는 형태로 진행 • 구문을 분석하는 단계에는 문장을 구성하는 단어 또는 어절들 사이의 문법적 관계를 분석하는 방법을 사용 • 구문 분석 방법으로 의존 구조 분석(Dependency parsing)과 구문 구조

분류	기술 영역	설명
		분석(Phrase structure parsing) 등을 사용 <ul style="list-style-type: none"> <li>• 어순이 자유롭고 주어나 목적어 등의 생략이 있는 한글일 경우 의존 구조 분석 방법을 활용</li> <li>• 개체명 분석은 사람, 시간, 조직, 지명 등 개체에 이름을 부여하기 위한 과정</li> <li>• 의미 분석은 형태소 분석, 구조 분석, 개체명 분석을 통해 추출한 각 어절 간 관계를 기반으로 의미론적으로 옳은 문장으로 해석하기 위한 과정</li> <li>• 담화 분석은 대화의 문맥 파악을 통해 상호 참조 문제를 해결하고 의도를 파악</li> </ul>
실행		<ul style="list-style-type: none"> <li>• 학습·추론 단계에서 이해한 자연어에 대응하여 응답을 생성</li> <li>• 응답을 생성하는 것 외에 전체 대화의 의도 분석과 담화에서 주요 내용의 추출 및 요약, 비슷한 담화 구조를 갖는 대화의 생성 등 다양한 방식으로 활용</li> </ul>

### 3. 인공지능 모델 구현과 서비스 구현의 차이점

인공지능서비스 로직 구현은 전통적인 소프트웨어 개발 방식과 큰 차이가 있다. 인공지능서비스 로직 구현 과정에는 모델, 학습, 재학습 등의 과정을 통해 인공지능서비스가 비즈니스 목표를 지속적으로 달성하도록 안정화한다.

〈표 1-10〉 인공지능 모델 구현과 서비스 구현의 차이점

구분	인공지능 모델 구현	서비스 구현
콘텐츠	모델	소스 코드
개발 과정	학습/검증/테스트	디버깅
지속적인 개선	재학습	패치

인공지능서비스 로직 구현이 기존 소프트웨어 개발 방법과 많은 부분에서 다름을 이해하는 것이 중요하다.

## 수행 내용 / 인공지능서비스 비즈니스 로직 구현하기

### 재료·자료

- 요건 분석서 / 비즈니스 로직 설계서 / 아키텍처 정의서
- 인공지능서비스 로직 구현을 위한 개발 도구  
(인터넷 가능한 컴퓨터 환경, 설치 파일을 다운로드받아야 하는 경우 대비)
- 운영 체제 - 윈도 10 환경

### 기기(장비·공구)

- 컴퓨터
- 문서 열람 도구(ODF 등)

#### ODF - Open Document Format for Office Application의 Spreadsheets

개방형 문서 형식은 스프레드시트, 차트, 프레젠테이션, 데이터베이스, 워드 프로세스를 비롯한 사무용 전자 문서를 위한 파일 형식으로, XML 파일 형식을 바탕으로 하여 OASIS 컨소시엄이 표준화하였다. 최근에는 상용 Office 제품군들도 ODF 포맷을 지원하고 있다.

### 안전·유의 사항

- 설계자 또는 이해관계자들과 원활한 의사소통을 할 수 있는 역량을 갖춘다.
- 비즈니스 로직 구현을 위한 설계서를 이해할 수 있어야 한다.
- 본 수행 내용에는 GPU 기반 인공지능서비스 구현을 위한 개발 환경을 기준으로 설치하고, GPU 환경 준비가 어려운 상황을 고려하여 팀을 통해 CPU 기반 환경도 지원이 되도록 정리하였다.

### 수행 순서

#### ① 인공지능서비스 로직을 구현한다.

##### 1. 설계서를 확인한다.

정적·동적 상세 설계 내용을 바탕으로 하여 모듈화 방향을 정하고 로직을 구현한다. 모듈의 응집도 및 결합도는 NCS 학습모듈(정보 통신 > 정보 기술 > 정보 기술 개발 > 응용 SW 엔지니어링)의 서버 프로그램 구현(LM2001020211)을 참고하도록 한다.

## 2. 설계 내용에 따라 로직을 구현한다.

수행하기에서는 서비스 기능 구현을 위한 핵심 내용을 절차 위주로 설명하고, 예제의 전체 코드는 부록 1-2를 참고한다.

설계서의 주요 내용은 다음과 같은 시나리오가 있다고 가정한다.

### 설계서의 내용 가정

(시나리오) 인공지능서비스가 이미지에서 객체 식별을 목적으로 개발된 인공지능 모델을 기반으로 특정 이미지를 입력했을 때 객체를 인식하고 응답하는 서비스를 구현한다.

(기능의 흐름) 이미지 업로드 → 인공지능 모델 연동 → 업로드 이미지 결과 표출 → 개선 요청 → 재학습의 순서로 서비스는 작동한다.

수행하기에서의 예제 소스 코드는 파이선 기반으로 구동하는 장고(Django) 웹 프레임워크를 활용한다. 다음은 장고 설치 및 실행을 위한 명령어들이므로 차례로 수행한다.

### 장고(Django 웹 프레임워크) 설치 및 실행하기

인공지능서비스 비즈니스 로직 구현은 파이선 언어를 기반으로 실행되는 장고(Django) 웹 프레임워크를 이용한다. 다음 명령어를 통해 장고를 설치하고 웹 환경을 구축한다.

#### 1) 장고 설치

```
(v_aincs_py38) c:\ncsprj\ > pip install django # 장고 설치
```

```
(v_aincs_py38) c:\ncsprj\ > pip install django-restframework # 장고 rest framework 설치
```

#### 2) 장고 웹서비스 구조 생성

```
(v_aincs_py38) c:\ncsprj\ > mkdir aisvc # 웹서비스 최상위 생성
```

```
(v_aincs_py38) c:\ncsprj\ > cd aisvc # 웹서비스 최상위 생성
```

```
(v_aincs_py38) c:\ncsprj\aisvc\ > django-admin startproject cfg . # 웹서비스 디렉터리 생성/설정
```

```
(v_aincs_py38) c:\ncsprj\aisvc\ > django-admin startapp aisvcmodel # 웹서비스 디렉터리 생성/설정
```

===== 다음과 같은 디렉터리 구조 생성 =====

```
C:\NCSPRJ\AISVC
```

```
| manage.py # 사이트 관리, 개인 컴퓨터에서 웹서버 시작
```

```
└─ cfg # 웹서비스 설정 구조
```

```
| asgi.py # 웹 서버, 프레임워크, 애플리케이션을 연결해주는 표준 인터페이스
```

```
| settings.py # 웹사이트 설정
```

```
| urls.py # urlresolver가 사용하는 패턴 목록
```

```
| wsgi.py # asgi 이전 호환
```

```
└─ __init__.py
```

#### 3) cfg\settings.py 설정 변경

[ 기본 설정 ]

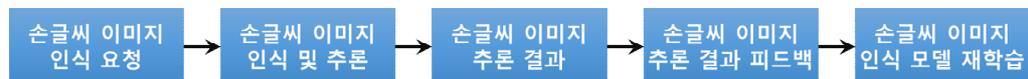
```

TIME_ZONE = 'Asia/Seoul'          # 정확한 현재시간 사용 위해
LANGUAGE_CODE = 'ko-kr'          # 웹사이트의 언어코드
STATIC_URL = '/static/'          # CSS 다룰 때 정적 파일 경로 추가
STATIC_ROOT = os.path.join(BASE_DIR, 'static') # ROOT 경로 생성
[ 데이터베이스 설정 ]
DATABASES = {
    'default': {
        'ENGINE' : 'django.db.backends.sqlite3',
        'NAME' : os.path.join(BASE_DIR, 'database.sqlite3'),
    }
}
[ 데이터베이스 생성 ]
python manage.py makemigrations
python manage.py migrate
[ 장고 관리자 생성 ]
(v_aincs_py38) c:\ncsprj\aisvc\ > python manage.py createsuperuser
사용자 이름 (leave blank to use 'cloud'): aisvcusr
이메일 주소:
Password:
Password (again):
4) 웹서버 시작
(v_aincs_py38) c:\ncsprj\aisvc\ > python manage.py runserver

```

(1) 서비스 요청 기능을 구현한다.

인공지능서비스의 비즈니스 로직은 [그림 1-8]과 같은 흐름이 있는 것으로 가정하고 로직을 구현한다.



출처: 집필진 제작(2021)  
[그림 1-8] 인공지능서비스 비즈니스 구성 예시

수행하기의 비즈니스 로직을 모두 구현하면 [그림 1-9]와 같은 1~3번의 순차적인 화면을 실행해 볼 수 있다.

## 손글씨 인식을 위한 이미지 업로드

Image: <input type="text" value="파일 선택"/> 3.jpg	<input type="button" value="Save"/>
<a href="#">손글씨 이미지 인식 결과 보기</a>	

출처: 집필진 제작(2021)  
[그림 1-9] 인공지능서비스 작동 예제 1

인식대상 이미지: 5	인공지능 다변: 답변: [3]
숫자 정정 의견:	<a href="#">0 1 2 3 4 5 6 7 8 9</a> → <a href="#">교정의견 보기</a>
인식대상 이미지: 3	인공지능 다변: 답변: [3]
숫자 정정 의견:	<a href="#">0 1 2 3 4 5 6 7 8 9</a> → <a href="#">교정의견 보기</a>

출처: 집필진 제작(2021)  
[그림 1-10] 인공지능서비스 작동 예제 2

인식 대상 이미지: 3
정정 숫자: 3 -- <a href="#">교정 적용</a>
정정 숫자: 4 -- <a href="#">교정 적용</a>

출처: 집필진 제작(2021)  
[그림 1-11] 인공지능서비스 작동 예제 3

위의 화면 및 로직 관련 소스 코드 구조는 다음과 같다. <표 1-11>의 비고는 이후 각 절차의 주요 내용을 요약한 것이므로 참고한다. 다음 절차를 단계별로 수행한다.

〈표 1-11〉 소스 코드 파일 구조

로직 구현 소스 코드 파일 구조	설명	비고
	<ul style="list-style-type: none"> <li>  db.sqlite3</li> <li>  manage.py</li> <li>  aisvcmodel</li> <li>    models.py</li> <li>    forms.py</li> <li>    urls.py</li> <li>    views.py</li> <li>    templates</li> <li>      aisvc</li> <li>        correctView.html</li> <li>        infInput.html</li> <li>        infList.html</li> <li>    settings.py</li> <li>    urls.py</li> <li>  images</li> <li>  model</li> <li>    mnist_dnn_model.h5</li> <li>    mnist_dnn_model.json</li> </ul>	<ul style="list-style-type: none"> <li>소형 DB</li> <li>장고메인파일</li> <li>App 디렉터리</li> <li>DB 모델</li> <li>화면 form구성</li> <li>url 매핑</li> <li>주요 로직구성</li> <li>화면</li> <li>화면</li> <li>화면</li> <li>cfg</li> <li>기본 설정</li> <li>공통 URL정보</li> <li>이미지 저장</li> <li>모델 정보</li> </ul>

가. 로직 구현 소스 코드 디렉터리 구조

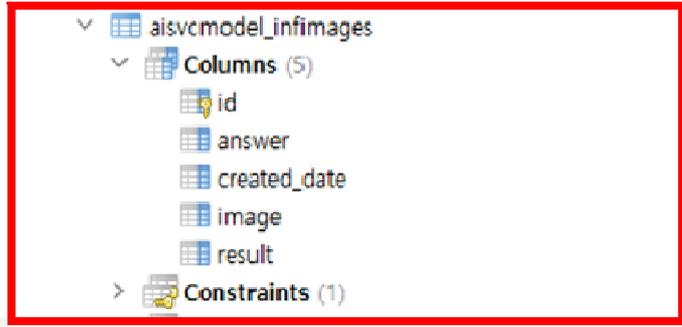
가. 출처: 집필진 제작(2021)

(가) 손 글씨 이미지 관련 정보 저장을 위한 모델(model.py)을 구현한다.

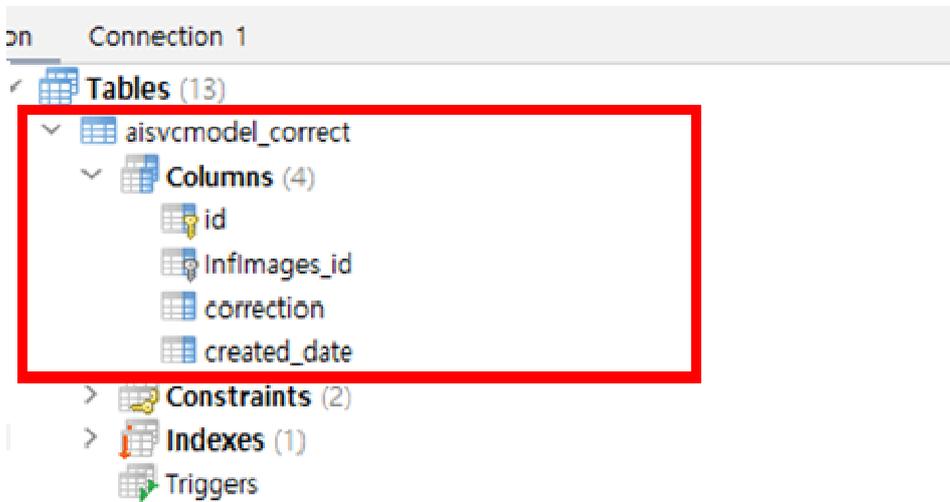
인공지능서비스의 첫 번째 이벤트로 손 글씨 이미지를 업로드하고 관련 정보를 데이터베이스 저장하는 것이다. 따라서 손 글씨 이미지 업로드 관련 정보 저장을 위한 데이터베이스 모델을 구현한다. 구현 상세 내용은 부록 1-2를 참고한다.

(나) 데이터베이스 테이블을 생성한다.

위 (가) 수행을 통한 모델 명세 후 장고의 명령어(위 2번 [데이터베이스 생성] 참고)를 실행하면 다음의 오른쪽과 같은 테이블 및 칼럼이 생성된다.



출처: 집필진 제작(2021)  
 [그림 1-12] 모델 명세 및 데이터베이스 생성 후 결과 1



출처: 집필진 제작(2021)  
 [그림 1-13] 모델 명세 및 데이터베이스 생성 후 결과 2

- (다) 이미지 파일 업로드를 위한 공통 설정(settings.py)을 변경한다. (부록 1-2 참고)
- (라) 공통 URL 정보(urls.py)를 변경한다. (부록 1-2 참고)
- (마) 인공지능서비스 애플리케이션(aisvcmodel) URL 정보(urls.py)를 변경한다. (부록 1-2 참고)
- (바) 인공지능서비스 이미지 업로드 화면 구성을 위한 form을 구현(forms.py)한다. (부록 1-2 참고)
- (사) 인공지능서비스 손 글씨 이미지 업로드 화면을 구현(inflInput.html)한다. (부록 1-2 참고)
- (아) 인공지능서비스 손 글씨 이미지 업로드 화면을 띄우기 위한 연결 정보를 구현한다. (부록 1-2 참고)

(2) 손 글씨 이미지 인식 결과를 제공하는 처리기를 구현한다.

업로드할 이미지를 선택하고 저장하게 되면 서버에서 이미지를 업로드하고 사전 학습된 모델을 이용하여 손 글씨 이미지를 판별하여 저장하는 기능을 구현한다.

(가) 손 글씨 이미지 인식 로직을 구현한다(views.py → HandwritingRecognition 함수).

손 글씨 이미지 인식을 위해 입력값을 받아 인공지능 모델에 요청하고 결괏값을 받아 처리하는 기능을 담당한다. 소스 코드에 json, h5 파일을 사용하는 데 해당 파일을 생성하는 별도의 파이선 코드(부록 1-2 참고)를 통해 오류가 없도록 구현한다.

(나) 업로드한 손 글씨 이미지의 전처리 함수를 구현한다(views.py). (부록 1-2 참고)

(다) 손 글씨 이미지 인식 결과를 저장하는 기능을 구현한다(views.py). (부록 1-2 참고)

(라) 손 글씨 이미지의 인식 결과 보기를 위한 URL 정보를 추가한다(urls.py). (부록 1-2 참고)

(마) DB 쿼리 결과를 화면에 매핑하는 코드를 구현한다. (부록 1-2 참고)

DB 쿼리를 위해 'InfImages.objects.all()'을 사용(ORM 메커니즘)하고, 조회 결과를 render라는 함수를 통해 html 문서에 표현하도록 구현한다.

#### 객체와 관계의 매핑 기술(Object Relational Mapping, ORM 기술)

객체(Object)와 관계형 데이터베이스(Relational Database)의 데이터(칼럼)를 매핑(Mapping)해 주는 기술을 의미한다. ORM 기술은 객체(클래스의 Instance) 간의 관계(메소드의 파라미터, 메소드 내에서의 객체 사용 등)를 바탕으로 하여 SQL 문장을 자동으로 생성해서 SQL 쿼리 문장이 없어도 데이터베이스의 데이터들을 다룰 수 있도록 하는 기술이다.

Java : Hibernate / PHP : Propel & Doctrine / Python : Django ORM & SQLAlchemy / C# : NHibernate & Entity Framework 등

(바) 손 글씨 이미지의 인식 결과를 보여 주는 화면(infList.html)을 구현한다. (부록 1-2 참고)

(3) 손 글씨 이미지 인식 결과에 대한 교정 의견 저장 및 재학습 기능을 구현한다.

(가) 사용자 교정 의견 데이터베이스 모델을 구현한다. (부록 1-2 참고)

사용자 교정 의견을 저장하기 위한 데이터 모델을 구성한다. models.py에 코드를 구현하고 데이터 모델을 재생성해야 한다. 생성 스크립트(python manage.py makemigrations)를 실행하고, 테이블을 생성한다(python manage.py migrate).

(나) 사용자 교정 의견을 처리하기 위해 URL 정보를 추가한다(urls.py 추가). (부록 1-2 참고)

손 글씨 이미지의 인식 결과가 원본 손 글씨 이미지와 다를 경우 사용자는 0~9까지 링크를 클릭해 의견을 전달하는 기능으로 해당 기능 제공을 위해 사용자 교정 의견 처리를 위한 url을 추가한다.

(다) 사용자의 교정 의견을 저장하는 기능을 구현한다. (부록 1-2 참고)

(4) 다양한 교정 의견 중에서 '선택 기능'과 재학습 기능을 구현한다.

(가) 다양한 교정 의견을 보기 위한 URL 정보를 추가한다(urls.py). (부록 1-2 참고)

(나) 교정 의견 테이블 쿼리 결과를 교정 의견 보기 화면에 매핑하는 코드를 구현한다. (부록 1-2 참고)

DB 쿼리를 위해 'InflImages.objects.get(pk=request.GET.get('pk'))'을 사용하고, 조회 결과를 render라는 함수를 통해 html 문서에 표현하도록 구현한다.

(다) 사용자의 다양한 교정 의견을 확인하는 화면(correctView.html)을 구현한다. (부록 1-2 참고)

(라) 사용자 교정 의견을 처리할 URL 정보를 추가한다. (부록 1-2 참고)

(마) URL 정보를 변경한다. (부록 1-2 참고)

(바) 선택된 사용자 피드백을 이용한 재학습 기능을 구현한다. (부록 1-2 참고)

#### 수행 tip

- HTTP 기반의 요청과 처리를 진행하기 때문에 기본적인 웹 페이지에 대한 개념을 숙지해야 한다.
- 장고(django) 프레임워크의 동작 방식에 대해 미리 숙지하고 실습하면 전체적인 흐름을 이해하기 쉽다.

# 1-3. 비즈니스 로직 문제 해결

**학습 목표** • 구현된 비즈니스 로직의 정상 작동 여부를 확인하고 문제 발생 시 문제를 해결할 수 있다.

## 필요 지식 /

### ① 인공지능서비스의 테스트에 대한 이해

#### 1. 소프트웨어 테스트 개념

구현된 애플리케이션이나 시스템을 사용자의 요구 사항 및 설계서를 충분히 만족하도록 개발되었는지 확인함과 더불어 기능과 비기능의 결함을 찾아내는 적극적인 개발 활동이다.

#### 2. 소프트웨어 테스트 원칙

(1) 본인이 개발한 프로그램 및 소스 코드는 본인이 직접 테스트하지 않는다.

개발자 본인이 개발한 소스 코드는 결함이 없을 것이라는 일반적인 선입견에 따라 개발자 본인이 직접 테스트할 경우 결함을 발견하지 못할 가능성이 크다.

(2) 효율적으로 결함을 제거하기 위한 법칙의 활용(파레토 법칙, 낚시의 법칙)

효율적으로 결함을 발견하고 가시화하며 결함 제거 및 예방의 순서로 결함을 정량적으로 관리할 수 있어야 한다. 이러한 효율적인 결함 발견 및 제거 방법으로 파레토 법칙 및 낚시 법칙을 적용할 수 있다.

(가) 파레토의 법칙

파레토의 법칙은 일반적으로 80 대 20 법칙으로 알려져 있으며, '전체 결과의 80%는 전체 원인의 20%에서 일어나는 현상'을 의미한다. 테스트 분야에는, 소스 코드 전체에서 발견되는 결함의 80%는 전체 기능 중에서 20% 소스 코드에 집중되어 있다.

(나) 낚시의 법칙

물고기는 특정 자리에서 잘 잡힌다는 사실을 경험적으로 알게 된다. 이와 마찬가지로 소스 코드의 결함도 특정 모듈, 라이브러리 및 기능에서 많은 결함이 발견된다는 것이 소스 코드 테스트에서의 낚시 법칙이다.

(3) 소프트웨어를 완벽하게 테스트하는 것은 불가능하다.

아무리 단순한 애플리케이션이어도 테스트를 수행하기 위한 케이스는 무수히 많기 때문에 애플리케이션을 완벽하게 테스트하는 것은 일반적으로 불가능하다.

(4) 계획 단계부터 테스트는 수행한다.

소프트웨어의 테스트 목적은 결함 발견에 있다. 하지만 소프트웨어 개발 계획 단계부터 수행할 수 있다면 결함을 예방하는 것이 가능하다.

## ② 테스트 케이스(Test Case)의 이해

### 1. 테스트 케이스 개념

기능 요구 사항의 준수 여부를 검증하기 위한 테스트의 조건과 애플리케이션의 다양한 입력값과 그에 따른 예상 출력값, 수행 결과 등 테스트를 위한 조건으로 명세화한 것이다.

### 2. 테스트 케이스 구성 요소

IEEE 829에 따르면 테스트 케이스는 식별자, 테스트 항목, 입력 및 출력 명세, 환경 설정, 특수 절차 요구 및 의존성 기술로 구성된다.

〈표 1-12〉 테스트 케이스 구성 요소(IEEE 829)

구성 항목	설명
식별자(Identifier)	항목 식별자, 일련번호
테스트 항목(Test Item)	테스트할 모듈 또는 기능
입력 명세(Input Specification)	입력값, 테스트 조건 등
출력 명세(Output Specification)	테스트 후 기대되는 출력값 결과
환경 설정(Environmental Needs)	테스트 수행 시 필요한 HW나 SW 환경
특수 절차 요구(Special Procedure Requirement)	테스트 수행 시 특수하게 요구되는 절차 사항
의존성 기술(Inter-case Dependencies)	테스트 케이스들 간 의존성 내용

## ③ 테스트 프로세스

테스트 수행과 관련된 다양한 활동들이 테스트 목적과 조건 등 의도에 따라 진행될 수 있도록 도와주는 역할을 한다.

### 1. 테스트 프로세스의 구성

계획/제어, 분석/설계, 구현/실행, 평가, 완료 5단계로 테스트 프로세스는 구성된다.



출처: 집필진 제작(2021)

[그림 1-14] 테스트 프로세스 5단계

## 2. 인공지능서비스 테스트 프로세스

### (1) 테스트 계획 및 제어

인공지능서비스 테스트의 목표와 목적 달성을 위해 필요한 활동을 계획하고, 계획에 따라 진행되는지 지속적인 제어 활동을 하는 단계이다.

### (2) 테스트 분석 및 설계

일반적인 문장으로 기술된 테스트 목적을 구체화하여 테스트 시나리오/케이스로 변환하는 단계이다.

### (3) 테스트 구현 및 실행

효과적이고 효율적인 테스트 수행을 위해 테스트 케이스를 조합하고 테스트 수행에 필요한 정보들을 포함하는 테스트 절차를 명세하는 단계이다.

### (4) 테스트 결과 평가

계획 단계에서 정의한 테스트 목표에 적합하도록 테스트가 수행되었는지를 평가하고 진행 상황을 기록하는 단계이다.

### (5) 테스트 완료

테스트를 수행할 때 명세했던 조건들을 수집하고 테스트를 수행할 때 발생한 사항 및 경험들을 추적하는 단계이다.

## ④ 인공지능서비스 로직 테스트 명세

### 1. 테스트 결과의 정리

인공지능서비스의 테스트가 완료된 후에는 테스트 케이스에 대한 설계 내용, 단계별 테스트 시나리오 및 테스트 결과 등 모두 내용이 포함된 문서를 일관되게 정리한다.

### 2. 테스트 요약

테스트 수행 계획, 테스트 소요 비용, 테스트 결과에 따라 판단 가능한 대상 소프트웨어의 품질 상태를 포함하여 테스트 요약 문서를 작성한다.

### 3. 품질의 상태 및 테스트 결과서 정리

품질의 상태는 테스트 품질 지표인 테스트 성공률, 테스트 커버리지, 발생 결함 수, 결함의 중요도 등을 포함한다.

테스트 결과서는 테스트 시 발견한 결함 내용을 중심으로 기록하고, 결함 내용, 결함 재현 순서, 결함 원인 등을 상세하게 기록한다.

### 4. 테스트 실행 절차 및 평가

단계별 테스트를 종료할 때 테스트 실행 절차를 리뷰하고 결과에 대한 평가를 수행하며, 그 결과에 따라 실행 절차를 최적화하여 다음 테스트에 적용한다.

### 재료·자료

- 제안 요청서, 제안서, 요구 사항 정의서 등의 요구 사항 관련 산출물
- 요구 사항 명세서, 유스케이스 명세서와 같은 명세 산출물
- 구현된 인공지능서비스

### 기기(장비·공구)

- 컴퓨터 또는 PC, 네트워크 장비, 프린터, 빔 프로젝트
- CASE 도구(SW 개발 지원 도구, SW 테스트 지원 도구 등)

### 안전·유의 사항

- 인공지능서비스 설계 산출물에 대한 이해가 선행되어야 한다.
- 목표 인공지능서비스의 요구 사항 범위에 대한 이해가 필요하다.

### 수행 순서

#### ① 인공지능서비스 테스트 케이스를 작성한다.

##### 1. 참조 문서를 수집한다.

인공지능서비스에 대해 상세하게 설계한 산출물과 요구 사항 정의서 및 요구 사항 명세서 등 분석서와 설계서를 수집한다.

##### 2. 단위 테스트 케이스를 작성한다.

상세하게 설계한 설계 관련 문서들과 요구 사항 정의서 및 요구 사항 명세서 등 분석서 및 설계서를 기반으로 단위 테스트 케이스를 작성한다.

##### (1) 단위 테스트 수행 방식을 결정한다.

동적 테스트와 정적 테스트 방식을 결정한다. 동적 테스트는 개발된 공통 모듈을 실행하여 테스트하는 방식이며, 정적 테스트는 자동화 도구를 사용하여 점검하는 방식이다.

##### (2) 단위 테스트 범위를 결정한다.

테스트 범위는 공통 모듈의 범위와 상위 요구 사항의 범위에 따라 판단하고, 테스트 수행 횟수를 결정하며 테스트 결과의 검증을 위해 결괏값 산출 방식을 결정한다.

(3) 단위 테스트 방식과 범위에 따른 단위 테스트 케이스를 작성한다.

단위 테스트 케이스 작성 필요 항목은 <표 1-13>을 참고하여 작성한다.

<표 1-13> 테스트 케이스 구성 요소(IEEE 829)에 따른 작성 내용

작성 항목	주요 내용
테스트 단계명, 작성자, 작성일, 승인자, 문서 버전 등의 항목을 작성한다.	단위/통합/시스템 테스트 등 계획에 따른 테스트의 단계와 작성자/승인자 및 문서 버전, 테스트 범위 및 테스트 조직 등 공통된 항목을 작성한다.
테스트 ID를 발행(발번)한다.	테스트 케이스의 유일성을 보장하기 위해 테스트 ID가 고유하도록 고유 식별자(ID)를 발행한다.
테스트 목적과 기능을 요약 작성한다.	테스트를 수행하는 목적과 테스트의 기능을 요약하여 작성한다.
테스트 입력값을 작성한다.	테스트를 수행할 때 입력해야 하는 데이터를 작성한다.
예상되는 결괏값을 작성한다.	테스트를 수행한 후 기대되는 결괏값(값, 메시지, 화면 등)들을 작성한다.
테스트를 수행할 환경을 작성한다.	테스트를 수행하는 데(제3자가 등) 문제가 없도록 논리적, 물리적 환경을 자세하게 작성한다.
테스트를 수행하기 위한 전제 조건을 작성한다.	다수의 테스트 케이스에 대해 의존성과 선행 조건 등 고려 사항을 상세하게 작성한다.
성공/실패 기준을 정의한다.	테스트 결과에 대해 성공(Pass)과 실패(Fail)를 결정하는 기준을 명확하게 작성한다.
테스트에 필요한 특수한 절차를 작성한다.	테스트를 할 때 사용자의 요구 사항이나 그 외 필요한 특수한 절차를 자세하게 작성한다.

② 작성한 단위 테스트 케이스의 내부 검토 및 이해관계자의 승인을 획득한다.

1. 단위 테스트 케이스의 적정성을 상호 검토한다.

사업 관리 책임자(PM), 설계자, 개발자 및 테스트 담당자 등 개발 업무 관련 이해관계자들이 작성한 테스트 케이스의 적정성을 상호 검토한다.

2. 요구 사항을 충족시킬 수 있는 테스트 범위인지 검토한다.

작성된 단위 테스트 케이스가 요구 사항을 충족하도록 누락되지 않았는지에 대해 테스트 범위를 검토한다.

3. 공통 모듈은 활용성이나 영향도 등을 충분히 고려하여 검토한다.

4. 작성된 단위 테스트 케이스를 사용자(고객)와 PM 등 이해관계자들에게 승인을 획득한다.

### ③ 테스트를 실시하고, 결과를 정리한다.

테스트의 결과를 확인하고 테스트 케이스에 성공과 실패 등 테스트 결과 및 관련 항목을 기록한다.

#### 1. 장고에서 단위 테스트 케이스를 구현한다.

장고 웹 프레임워크는 단위 테스트를 수행할 수 있는 도구를 지원한다. 이른바 테스트 주도 개발(Test Driven Developer, TDD)이라는 개념인데, 애플리케이션을 개발하면서 단위 모듈에 대한 테스트 코드도 작성하면서 결함을 조치해 나가는 개념이다.

장고 웹 프레임워크에는 테스트 커버리지를 점검하는 도구도 지원하고 있다. 해당 도구를 사용하면 테스트 코드를 얼마만큼 작성했는지를 시각적으로 확인할 수 있다.

#### 테스트 커버리지 확인 명령어

우선 pip 명령어를 통해 커버리지 도구를 설치한다.

```
(v_aincs_py38) C:\ncsprj\aisvc> pip install coverage
```

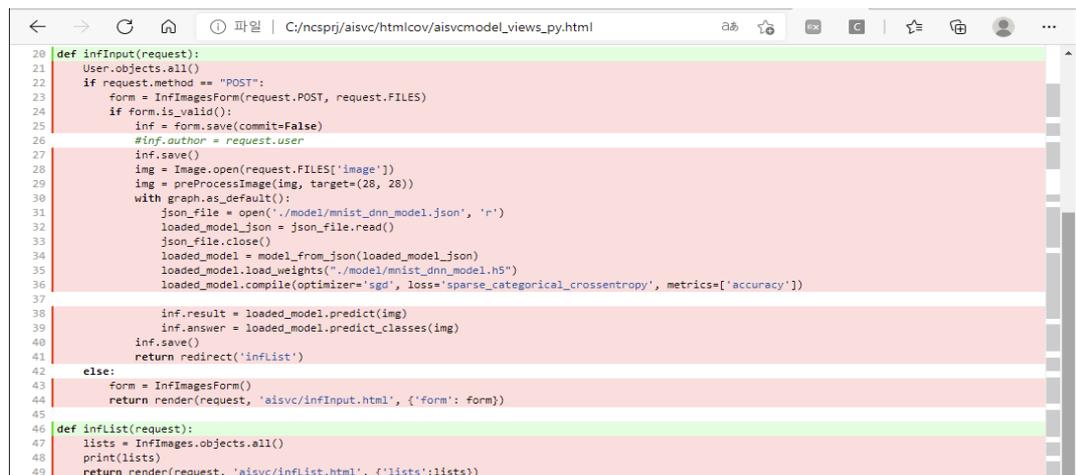
장고 프레임워크의 테스트 커버리지를 확인하기 위한 명령어를 수행한다.

```
(v_aincs_py38) C:\ncsprj\aisvc> coverage run manage.py test
```

장고 프레임워크의 테스트 커버리지를 html로 확인하기 위한 명령어를 수행한다.

```
(v_aincs_py38) C:\ncsprj\aisvc> coverage html
```

위의 명령어를 순서대로 수행하면 다음과 같이 개발 프로그램의 커버리지를 쉽게 확인할 수 있다. 문장 대비 테스트 코드가 있는지 여부를 도구가 검토하여 커버리지 상태를 확인할 수 있도록 지원한다.



```
20 def infInput(request):
21     User.objects.all()
22     if request.method == "POST":
23         form = InfImagesForm(request.POST, request.FILES)
24         if form.is_valid():
25             inf = form.save(commit=False)
26             #inf.author = request.user
27             inf.save()
28             img = Image.open(request.FILES['image'])
29             img = preProcessImage(img, target=(28, 28))
30             with graph.as_default():
31                 json_file = open('./model/mnist_dnn_model.json', 'r')
32                 loaded_model_json = json_file.read()
33                 json_file.close()
34                 loaded_model = model_from_json(loaded_model_json)
35                 loaded_model.load_weights("./model/mnist_dnn_model.h5")
36                 loaded_model.compile(optimizer='sgd', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
37
38                 inf.result = loaded_model.predict(img)
39                 inf.answer = loaded_model.predict_classes(img)
40                 inf.save()
41                 return redirect('infList')
42     else:
43         form = InfImagesForm()
44         return render(request, 'aisvc/infInput.html', {'form': form})
45
46 def infList(request):
47     lists = InfImages.objects.all()
48     print(lists)
49     return render(request, 'aisvc/infList.html', {'lists':lists})
```

출처: 집필진 제작(2021)

[그림 1-15] 테스트 커버리지 확인 결과 - 1

## Coverage report: 59%

Module	statements	missing ↓	excluded	coverage
aisvcmodel\views.py	70	52	0	26%
aisvcmodel\models.py	15	2	0	87%

출처: 집필진 제작(2021)

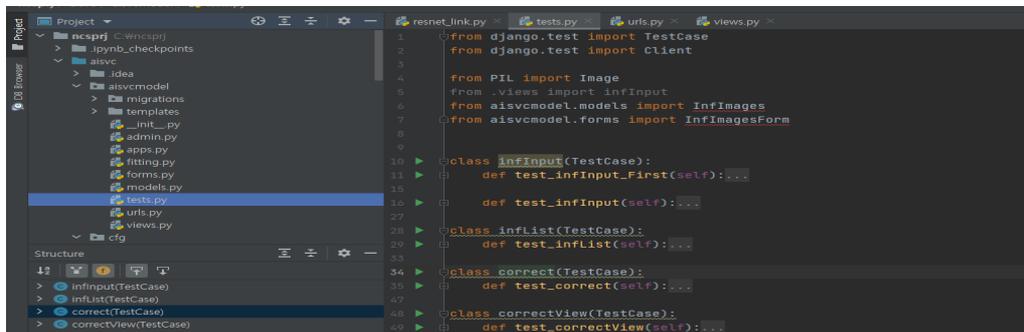
[그림 1-16] 테스트 커버리지 확인 결과 - 2

그러면 구현된 단위 모듈의 테스트를 위해 다음과 같이 'aisvcmodel/tests.py(django-admin startapp aisvcmodel 명령어를 통해 자동으로 생성된 템플릿 파일)'에 테스트 코드를 구현한다. 집필진은 예시로 5개의 테스트 코드를 구현하였고, 전체 내용은 부록 1-3을 참고한다.

### 단위 테스트를 위한 테스트 코드(aisvcmodel/tests.py) - 2개 사례

```
class InInput(TestCase):  
    def test_inInput_First(self): # 손 글씨 이미지 업로드 테스트  
        c = Client() # 브라우저 대신 요청을 보내고 받는 처리를 해 주는 객체  
        res = c.get('/aisvc/inInput') # 해당 페이지가 존재하는지 테스트하는 코드  
        self.assertEqual(res.status_code, 200) # 응답 코드가 200번인지 확인하는 코드  
  
    def test_inInput(self): # 손 글씨 이미지 업로드 테스트  
        img = Image.open("C:\\ncsprj\\test\\8번.jpg")  
        file_dict = {'image':img}  
        form = InInputForm(data={'files':file_dict})  
        self.assertTrue(form.is_valid()) # form 데이터가 유효한지 확인하는 코드
```

다음과 같이 5개의 테스트 코드를 구현한 사례를 예시하였다.



The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes folders like 'aisvcmodel' and 'tests.py'. The code editor shows the following Python code:

```
1 from django.test import TestCase  
2 from django.test import Client  
3  
4 from PIL import Image  
5 from .views import inInput  
6 from aisvcmodel.models import InImages  
7 from aisvcmodel.forms import InImagesForm  
8  
9  
10 class InInput(TestCase):  
11     def test_inInput_First(self):  
12  
13     def test_inInput(self):  
14  
15  
16  
17  
18 class InList(TestCase):  
19     def test_inList(self):  
20  
21  
22  
23  
24 class Correct(TestCase):  
25     def test_correct(self):  
26  
27  
28  
29  
30 class CorrectView(TestCase):  
31     def test_correctView(self):  
32
```

출처: 집필진 제작(2021)

[그림 1-17] 테스트 코드 구현 사례

## 2. 장고에서 구현한 단위 테스트 케이스를 실행한다.

장고 웹 프레임워크에서 웹을 실행하지 않고도 개발된 코드들을 테스트할 수 있는 명령어는 다음과 같다. 위의 작성된 테스트 코드에 대해 테스트를 수행하도록 명령을 수행해 본다. 5개의 테스트를 수행했다는 메시지가 나타났고, 오류는 없는 것으로 확인된다.

테스트를 하는 과정에서 `self.assertTrue` 등과 같은 테스트 코드를 활용하여 애플리케이션 결과의 적정성을 검증한다. 파이선 단위 테스트의 `assertion` 종류에 대해 부록 1-3에 정리하였으니 참고한다.

### 개발된 코드에 대해 테스트를 수행하는 명령어

```
(v_aincs_py38) C:\ncsprj\aisvc> python manage.py test aisvcmodel
```

```
-----  
Ran 5 tests in 0.050s
```

```
OK
```

## 3. 단위 테스트 케이스 실행 결과에 따라 결함 조치 및 재실행한다.

다시 테스트 커버리지를 확인하고 결과에 따라 미실행 테스트의 경우 테스트 코드 구현 및 재테스트 등 반복적인 개발을 진행한다. 다음 그림의 녹색 부분은 테스트 코드를 통해 테스트를 진행했다는 의미이다.

```
45  
46 def infList(request):  
47     lists = InfImages.objects.all()  
48     print(lists)  
49     return render(request, 'aisvc/infList.html', {'lists':lists})  
50  
51 def correct(request):  
52     corr = Correct()  
53     corr.InfImages = InfImages.objects.get(pk=request.GET.get('pk'))  
54     corr.correction = request.GET.get('Correction')  
55     #corr.author = request.user  
56     corr.save()  
57     return redirect('infList')  
58  
59 def correctView(request):  
60     inf = InfImages.objects.get(pk=request.GET.get('pk'))  
61     lists = Correct.objects.filter(InfImages=inf)  
62     return render(request, 'aisvc/correctView.html', {'lists': lists})  
63
```

출처: 집필진 제작(2021)

[그림 1-18] 테스트 커버리지 확인 결과

### 수행 tip

- 테스트 주도 개발의 개념과 소스 코드를 통해 단위 테스트 케이스를 생성하는 개념을 숙지하여 실습한다.
- 문제 발견 활동은 결함이 있을 것을 가정하고 진행하는 적극적인 활동이므로 사소한 것이라도 문제를 파악하고 결과를 정리한다.

### 교수 방법

- 응용 소프트웨어 개발에 필요한 하드웨어 및 소프트웨어 제품 유형에 대해 설명하고, 각 제품의 용도에 대해 제시한 후 수업을 진행한다.
- 여러 개발 도구들에 대한 이해 및 이전 사용 경험 여부에 대해 파악하여 개발 도구를 선정하여 수업을 진행한다.
- 여러 개발 도구의 매뉴얼 및 홈페이지 등을 방문하여 다양한 자료를 제시한 후 설명한다.
- 다양한 개발 도구의 설치 과정에 대해 단계적 실습이 이루어질 수 있도록 지도한다.
- 설계 내용을 파악하여 이를 개발 언어를 활용하여 작성할 수 있는 절차를 구체적으로 제시한 후 내용에 대해 설명한다.
- 구현된 로직을 테스트할 수 있는 테스트 케이스 작성 기법에 대해 특성 및 차이점을 비교·분석하여 제시하고 설명한다.

### 학습 방법

- 응용 소프트웨어 개발에 필요한 하드웨어 및 소프트웨어 제품 유형에 대해 파악하고, 각 제품의 용도에 대해 학습한다.
- 여러 개발 도구들에 대한 이해 및 이전 사용 경험을 바탕으로 하여 선정된 개발 도구로 실습한다.
- 여러 개발 도구의 매뉴얼 및 홈페이지에서 다양한 자료를 다운받아 학습한다.
- 다양한 개발 도구들의 설치 과정에 대해 단계적으로 실습하여 과정에 대해 학습한다.
- 설계서를 파악하여 개발 언어로 작성할 수 있는 절차에 대해 학습한다.
- 공통 모듈의 중요성을 인지하여 효율적 코드로 작성될 수 있도록 고급 기법 습득의 필요성에 대해 인식하고 해당 내용을 학습한다.

## 학습 1 평가

### 평가 준거

- 평가자는 학습자가 학습 목표를 성공적으로 달성하였는지에 대해 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습 내용	학습 목표	성취수준		
		상	중	하
인공지능서비스 개발 환경 이해	- 인공지능서비스 애플리케이션 설계서에 따라 인공지능 비즈니스 로직 구현을 위한 개발 환경을 구성할 수 있다.			
비즈니스 이해 및 로직 구현	- 인공지능서비스 애플리케이션 설계에 따라 인공지능 비즈니스 로직을 구현할 수 있다.			
비즈니스 로직 문제 해결	- 구현된 비즈니스 로직의 정상 작동 여부를 확인하고 문제 발생 시 문제를 해결할 수 있다.			

### 평가 방법

- 포트폴리오

학습 내용	평가 항목	성취수준		
		상	중	하
인공지능서비스 개발 환경 이해	- 인공지능서비스 개발 환경에 대한 이해 능력			
	- 인공지능서비스 개발 환경(GPU, CPU 등)을 구성하는 능력			
	- 파이선 지원하는 각종 라이브러리를 설치할 수 있는 능력			
비즈니스 이해 및 로직 구현	- 구현 언어에 대한 이해 능력			
	- 인공지능서비스 유형에 대한 이해 능력			
	- 설계서를 파악하고 모듈을 구성하는 능력			
비즈니스 로직 문제 해결	- 구현된 로직을 직접 실행하는 능력			
	- 로직으로 구성된 모듈의 성능 측정 능력			
비즈니스 로직 문제 해결	- 비즈니스 목표와 측정 성능 비교를 통해 재학습 여부를 판단할 수 있는 능력			

• 문제 해결 시나리오

학습 내용	평가 항목	성취수준		
		상	중	하
인공지능서비스 개발 환경 이해	- 인공지능서비스 개발 환경을 구성할 때 발생하는 다양한 문제를 해결할 수 있는 능력			
	- GPU, 가상 환경 등 새로운 환경을 구성할 수 있는 능력			
비즈니스 이해 및 로직 구현	- 팀별로 단순한 비즈니스 시나리오를 제시하는 능력			
	- 비즈니스 시나리오를 로직으로 구현하는 능력			
비즈니스 로직 문제 해결	- 모델의 성능 저하 원인에 따라 모델의 성능을 높이기 위하여 재학습(Retraining) 기능과 모델 재구성(Remodeling) 기능을 구현하는 능력			
	- 인공지능 모델 검증을 위하여 모델별 모형 평가를 위한 관리 기준을 정의한 내용의 적정성을 평가			

피드백

<p>1. 포트폴리오</p> <ul style="list-style-type: none"> <li>- 실습 과정을 포트폴리오에 따라 평가한 후에 개선해야 할 사항 등을 정리한 후 학습자에게 설명한다.</li> <li>- 인공지능서비스 환경 구성 방법에 대한 평가가 일정 점수 이하인 학습자에게는 추가 학습이 필요한 부분에 대해서 설명한다.</li> <li>- 인공지능서비스 로직 구현 결과에 대한 평가가 일정 점수 이하인 학습자에게는 추가 학습이 필요한 부분에 대해서 설명한다.</li> <li>- 설계서를 파악하여 작성된 로직이 설계서대로 개발되었는지 확인하여 비정상적으로 동작할 때 그 문제점을 정리하고 개선 사항을 제시한다.</li> <li>- 설계서를 파악하여 작성된 인공지능서비스 로직 구현 결과가 '상'인 경우 모범 사례로 공유한다.</li> <li>- 학습 능력이 우수한 학습자는 타 비즈니스 환경에 대한 로직 구현을 숙지하도록 심화 콘텐츠의 제공과 이에 대한 내용을 학습하도록 지도한다.</li> </ul> <p>2. 문제 해결 시나리오</p> <ul style="list-style-type: none"> <li>- 제출한 로직 구현 결과를 평가한 후에 주요 사항에 대하여 의견을 제시한다.</li> <li>- 로직 설계서에 정의된 기능이 정상적으로 동작하는지를 보고 비정상 동작의 원인과 조치 방안에 대해 설명한다.</li> <li>- 설계서를 파악하여 인공지능서비스 로직 구현 결과가 '상'인 경우 모범 사례로 공유한다.</li> <li>- 설계서를 파악하여 인공지능서비스 로직 구현 결과가 '중'인 경우 부족한 부분을 보완할 수 있도록 설명한다.</li> <li>- 설계서를 파악하여 인공지능서비스 로직 구현 결과가 '하'인 경우 모범 사례를 참조하여 다시 구현할 수 있도록 유도한다.</li> </ul>
--

학습 1	인공지능서비스 비즈니스 로직 구현하기
<b>학습 2</b>	<b>인공지능서비스 플랫폼 연동하기</b>
학습 3	인공지능서비스 모델 연동하기
학습 4	인공지능서비스 인터페이스 연동하기

## 2-1. 플랫폼 연동 환경 구성

**학습 목표** • 인공지능서비스 설계에 따라 인공지능서비스 플랫폼 연동 환경을 구성할 수 있다.

### 필요 지식 /

#### ① 인공지능서비스 플랫폼 이해

플랫폼(platform)이란 지면(plat)과 형태(form)의 합성어로 경계가 없던 땅이 구획되면서 용도에 따라 다양한 형태로 활용될 수 있는 공간을 의미한다. 플랫폼의 대표적인 예로는 승강장(버스, 지하철, 기차 등)을 들 수 있다. 이러한 플랫폼은 사람이 이동 수단을 만나기 위한 접점, 매개 지점의 역할을 한다고 볼 수 있다. 따라서 플랫폼이란 어떤 목적 달성을 위한 매개 수단으로 정리할 수 있다.

한편, 인공지능서비스란 인간의 추론, 지각, 자연 언어 이해 능력 등을 컴퓨터 프로그램으로 실현한 기술을 의미한다. 인공지능서비스 플랫폼이란 인공지능서비스의 목적 달성을 위한 매개 수단(구조 및 기반 관련 서비스 등)으로 정의할 수 있다. 인공지능서비스는 인간의 시각(비전), 의사소통, 특정 분야 학습 및 활용 분야에 활발하게 연구 및 개발이 이루어지고 있다.

〈표 2-1〉 인공지능 서빙 플랫폼 사례

구분	컴퓨터 비전(시각)	언어 처리(의사소통)	기계 학습(전문가 패턴)
네이버 CLOVA	CLOVA Vision	CLOVA Chatbot	
ETRI	-	GENIE Dialog	-
아마존 AWS	Amazon Rekognition	Amazon Lex	Amazon SageMaker
Google GCP	Vision API	Dialogflow	Google Cloud Machine Learning Engine

구분	컴퓨터 비전(시각)	언어 처리(의사소통)	기계 학습(전문가 패턴)
IBM Watson	Watson Visual Recognition	Watson Assistant	Watson Machine Learning
MS	Computer Vision API	Language Understanding	Azure Machine Learning Studio

## ② 인공지능서비스 플랫폼 연동 환경

인공지능서비스 플랫폼 연동 환경에 대해 쉬운 예로 ‘지하철 플랫폼 이용’을 위해서는 ‘개찰구’라는 ‘지하철 공사’가 설치해 놓은 구조물을 ‘정해진 규칙’에 따라 통과해야 한다. 이와 유사한 개념으로 인공지능서비스 플랫폼의 이용 환경을 이해할 수 있다.

‘특정 인공지능서비스 플랫폼 기관(기업)’이 구성해 놓은 ‘이용 환경(구성, 프로토콜, 인터페이스)’을 ‘정해진 규칙’에 따라 적절하게 구성 및 구현해야 인공지능서비스 플랫폼을 활용할 수 있게 된다.

## 수행 내용 / 인공지능서비스 플랫폼 연동 환경 구성하기

### 재료·자료

- 요건 분석서(요구 사항 정의서, 요구 사항 명세서) / 아키텍처 정의서
- 인공지능서비스 플랫폼 연동을 위한 인터넷 환경

### 기기(장비·공구)

- 문서 열람 도구(ODF 등)
- 인터넷 브라우저(Chrome 등)

### 안전·유의 사항

- 인공지능 플랫폼의 개념을 이해할 수 있어야 한다.
- 설계자 또는 이해관계자들과 원활한 의사소통을 할 수 있는 역량을 갖춘다.
- 인공지능서비스 플랫폼 연동을 위한 요구 환경을 갖출 수 있어야 한다.  
(플랫폼 연동을 위한 가입 정보에 따라 비용이 발생할 수 있으므로 주의한다.)

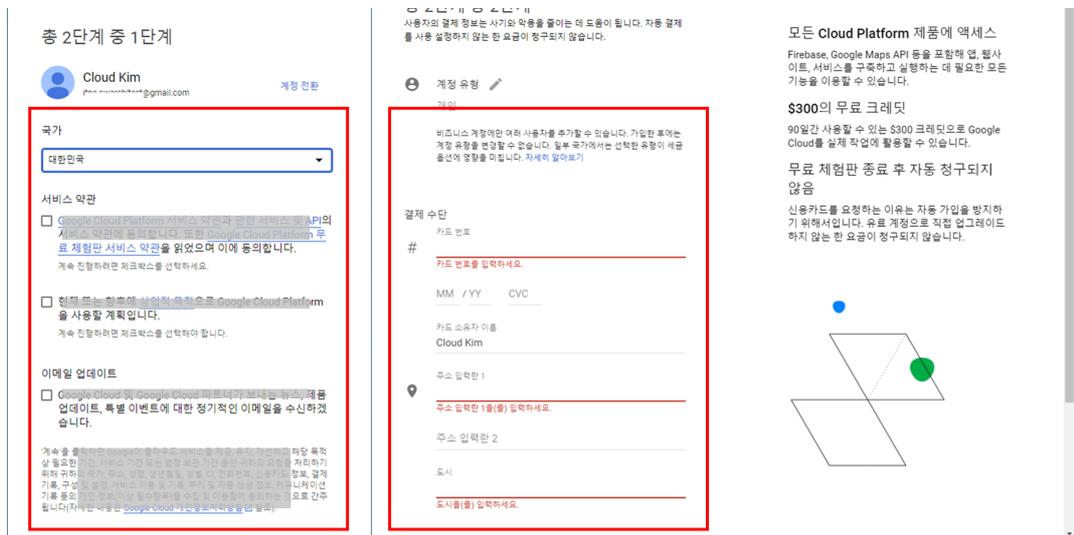
### 수행 순서

① 인공지능서비스 플랫폼 연동을 위해 환경을 구성한다.

인공지능 모델 서빙을 위한 플랫폼 기관들은 ‘필요 지식’에서 살펴보았듯이 다양하게 존재하지만 ‘수행하기’에는 ‘구글 클라우드 플랫폼(Google Cloud Platform, GCP)’을 플랫폼 연동 환경으로 구성해 본다. ‘수행하기’에서 제시한 GCP 외에 AWS, IBM Watson 등 다양한 플랫폼이 존재하므로 해당 플랫폼의 가입하기 또는 안내 페이지를 참고한다.

1. 플랫폼에 가입한다(<https://cloud.google.com/gcp/>).

구글 플랫폼 사용을 위해 해당 인터넷 페이지에 안내하는 개인 정보 및 결제 정보를 입력하고 가입을 한다.



출처: 집필진 제작(2021)  
[그림 2-1] 플랫폼 연동을 위한 플랫폼 제공 기관 가입하기

**결제 정보와 관련 ...**

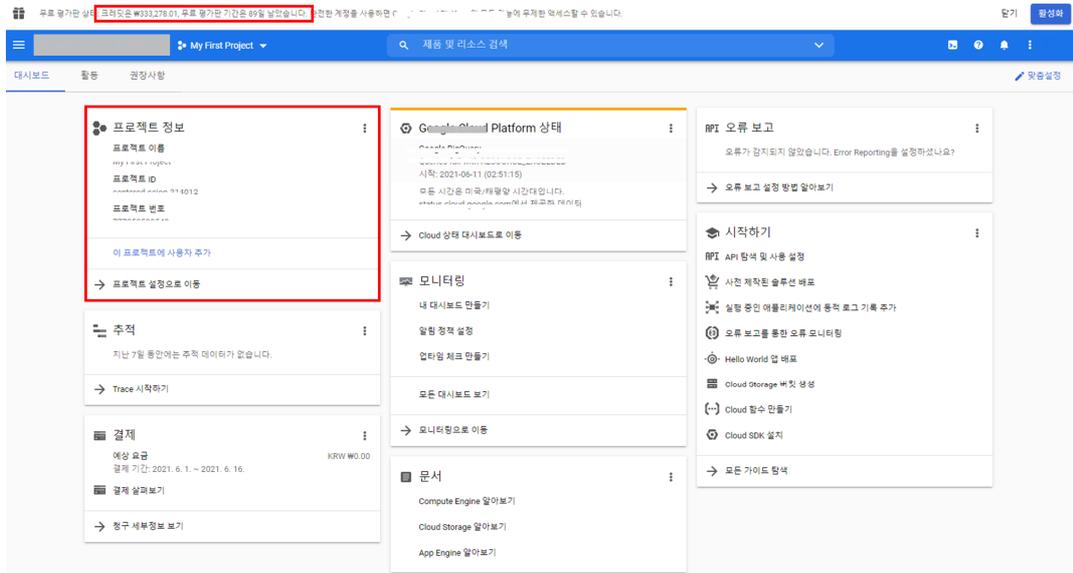
플랫폼 업계는 다양한 플랫폼 환경을 제공하는 것이 비즈니스 모델이다. 따라서 플랫폼 환경을 이용하기 위해서는 비용 지불이 필요한 것이다.

GCP의 경우 2021년 현재 다음과 같은 안내를 통해 프로모션을 진행 중이다.

**“무료 평가판 상태: 크레딧은 ₩335,369.00, 무료 평가판 기간은 91일 남았습니다. 완전한 계정을 사용하면 Google Cloud Platform의 모든 기능에 무제한 액세스할 수 있습니다.”**

인공지능서비스 구현을 위한 GCP 플랫폼 연동은 학습 목적으로 진행하므로 테스트한 후 결제 정보를 삭제하여 혹시 있을 불필요한 비용 지불이 되지 않도록 주의한다.

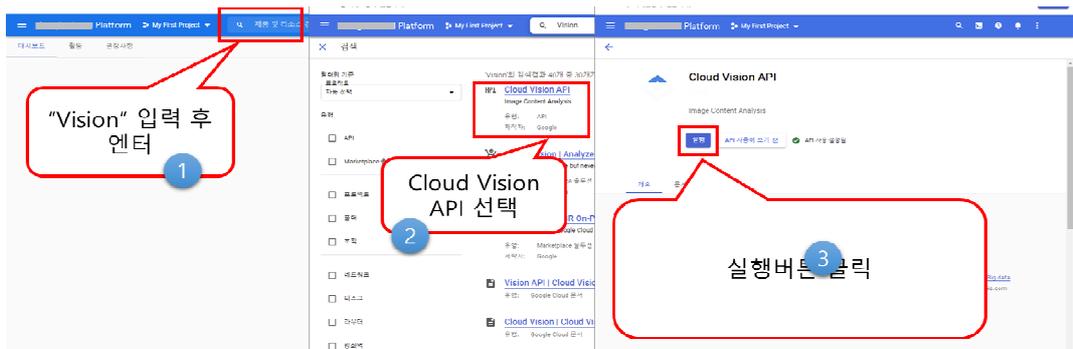
구글 플랫폼에 가입을 완료하면 [그림 2-2]와 같은 화면을 볼 수 있다.



출처: 집필진 제작(2021)  
[그림 2-2] 플랫폼 제공 기관 가입 후 대시 보드

## 2. 플랫폼의 인공지능은 ‘Cloud Vision API’를 연동한다.

GCP의 대시 보드 상단 검색창에 “Cloud Vision API” 또는 “Vision”을 입력한 후 키보드의 “엔터”키를 클릭한다(①). 검색 결과 중에서 “Cloud Vision API”를 클릭(②)한 후 다음 이동 화면의 “실행 or 사용 or 관리” 버튼을 클릭한다(③).

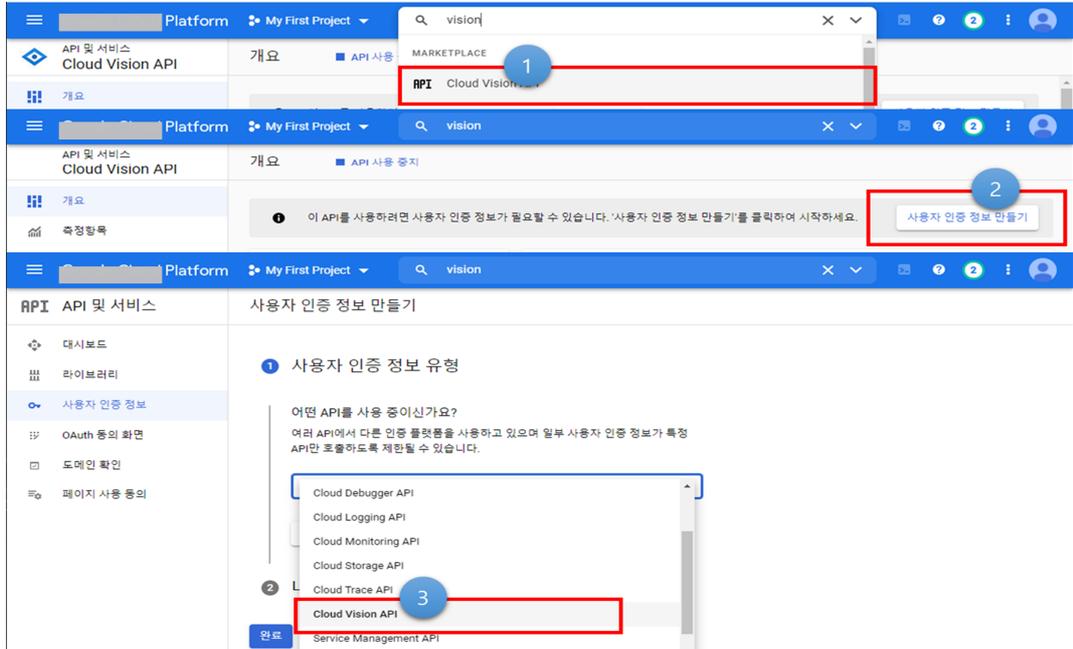


출처: 집필진 제작(2021)  
[그림 2-3] 플랫폼 연동을 위한 인공지능서비스 선택

위의 과정을 모두 진행하면 “Cloud Vision API”의 관리 화면으로 이동하게 된다. 이로써 GCP의 “Cloud Vision API”를 사용할 수 있는 플랫폼 환경이 마련되었다.

## 3. 플랫폼의 인공지능 연동을 위해 플랫폼의 “사용자 인증 정보”를 생성한다.

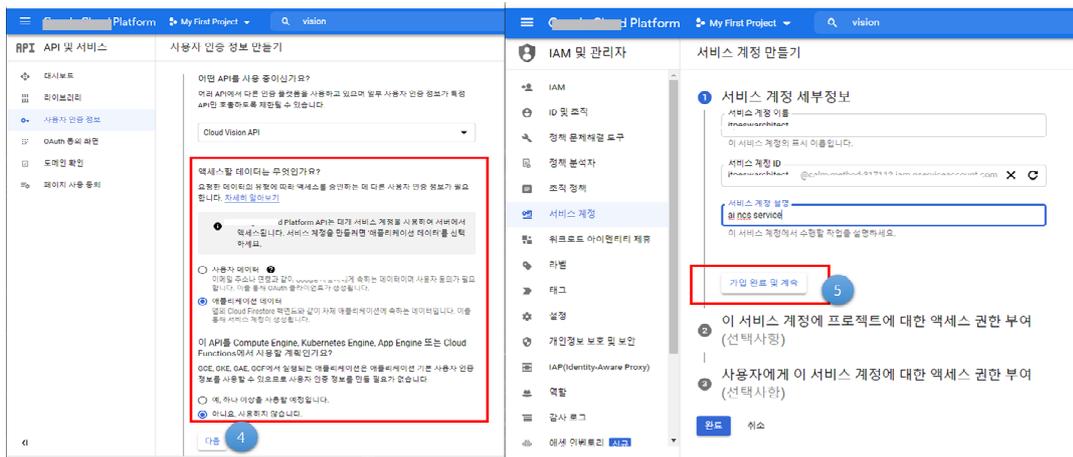
GCP 플랫폼 연동을 하려면 “사용자 인증 정보”를 생성해야 한다. GCP에서 제공하는 “사용자 인증 정보 만들기(②)”를 클릭하고, 연이어 나오는 새로운 화면에 “사용자 인증 정보 유형” 부분에 “수행하기”에는 Vision 플랫폼을 사용할 것이므로 “Cloud Vision API”를 선택하고 “완료” 버튼을 클릭한다.



출처: 집필진 제작(2021)  
 [그림 2-4] 플랫폼 연동을 위한 사용자 인증 정보 만들기 - 1

이후 [그림 2-5]와 같이 “애플리케이션 데이터” 선택, “아니요, 사용하지 않습니다” 선택 후 “다음(④)” 버튼을 클릭한다(관련 내용은 항목을 선택하면서 자세하게 안내하고 있으므로 구체적으로 확인한다).

“Cloud Vision API”를 사용할 서비스 계정 이름을 생성한다. 서비스 계정 이름에 적당한 명칭을 입력한 후 “가입 완료 및 계속(⑤)” 버튼을 클릭한다.

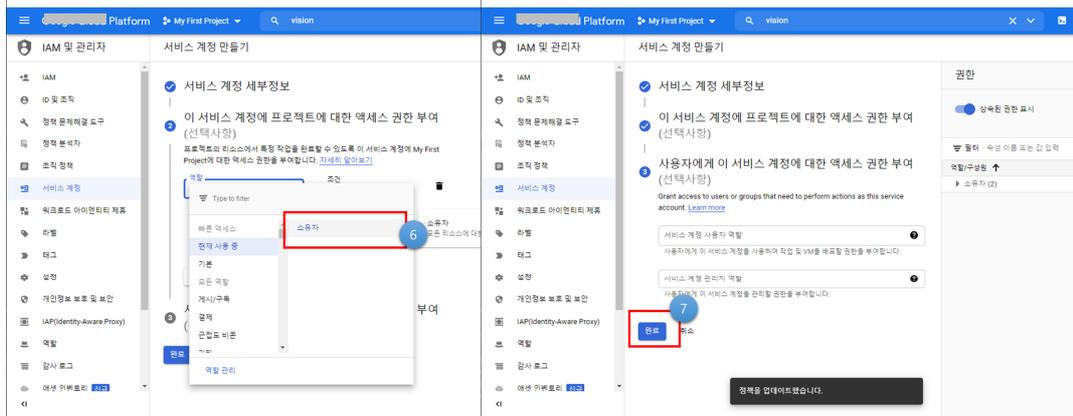


출처: 집필진 제작(2021)  
 [그림 2-5] 플랫폼 연동을 위한 사용자 인증 정보 만들기 - 2

서비스 계정에 대한 접근 권한을 부여하는데 “수행하기”에서는 “소유자” 권한을 주었다. 다른 권한이 어떤 것이 있는지 살펴보고 다양한 접근 권한을 줄 수 있으므로 참

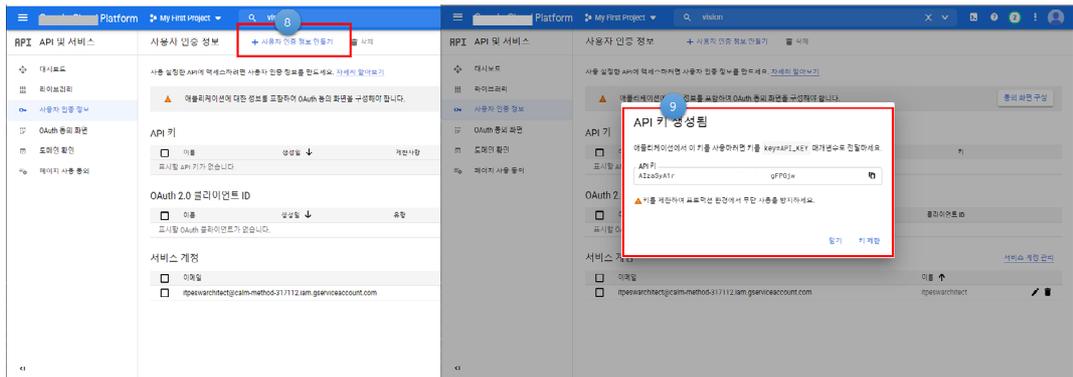
고한다. 접근 권한을 선택한 후 “완료(6)” 버튼을 클릭한다.

마지막으로 "사용자에게 이 서비스 계정에 대한 액세스 권한 부여"는 옵션(입력하지 않아도 되는 값) 사항이므로 입력은 생략하고 "완료" 버튼을 클릭한다.



출처: 집필진 제작(2021)  
[그림 2-6] 플랫폼 연동을 위한 사용자 인증 정보 만들기 - 3

⑦번까지 완료한 후 사용자 인증 정보의 API Key 발급을 위해 "사용자 인증 정보 만들기"를 클릭한다. “사용자 인증 정보 만들기” 메뉴 중에서 API Key 만들기를 클릭하면 생성된 키의 정보를 보여 주는 새로운 팝업창이 뜬다. 확인한 후 "닫기" 버튼을 클릭한다.

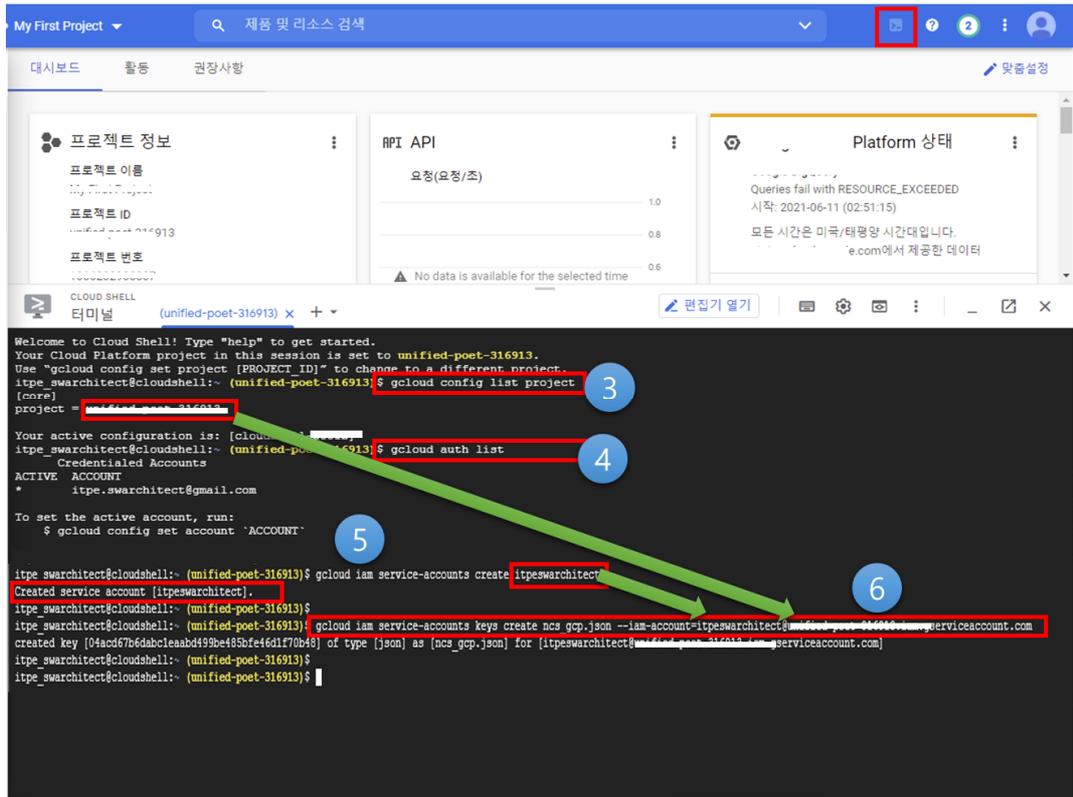


출처: 집필진 제작(2021)  
[그림 2-7] 플랫폼 연동을 위한 사용자 인증 정보 만들기 - 4

"Google Vision API" 신청과 사용자 인증 정보까지 완료했다. 인공지능서비스 플랫폼 연동을 위해 마지막 단계로 연동을 위한 인증 정보를 json 형태로 생성하고 플랫폼 연동 개발 환경에 구성해 줘야 한다.

4. 플랫폼의 인공지능 연동용 “사용자 인증 정보” json 파일을 생성한다.

GCP 대시 보드 화면 오른쪽 상단에 shell 아이콘을 클릭하면 [그림 2-8]과 같이 브라우저 하단에 명령 인터페이스(Command Line Interface, CLI) 창이 추가된다. 해당 창에서 명령 줄을 사용하여 사용자 인증 정보 연동 파일(json)을 생성한다.



출처: 집필진 제작(2021)

[그림 2-8] 플랫폼 연동용 “사용자 인증 정보” 파일 생성(③, ④, ⑤, ⑥)

**플랫폼 연동 용 “사용자 인증 정보” 파일 생성 명령어**

사용자 인증 목록을 확인한다.

```
$ gcloud auth list
```

프로젝트 명칭을 확인한다.

```
$ gcloud config list project
```

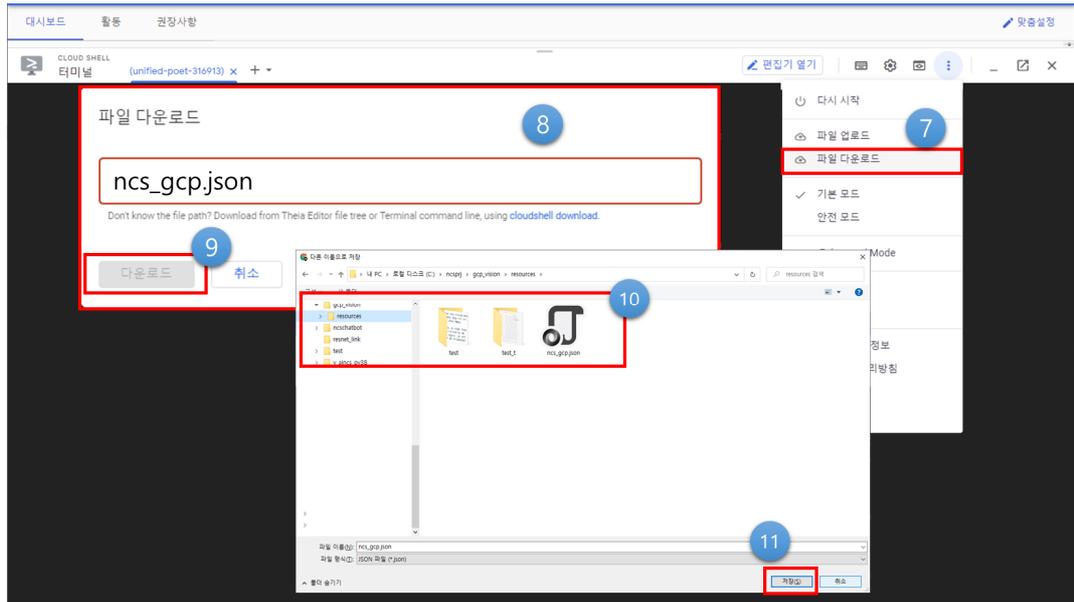
서비스 계정(IAM, Identity and Access Management)을 생성한다.

```
$ gcloud iam service-accounts create itpeswarchitect
```

“서비스 계정(ServiceAccount)”과 “프로젝트 명칭(ProjectName)” 이용자를 위한 FileName.json을 생성한다.

```
$ gcloud iam service-accounts keys create FileName.json --iam-account=ServiceAccount@PrjectName.iam.gserviceaccount.com
```

위와 같은 명령 줄을 수행한 후 명령 창 오른쪽 상단의 "더 보기..." 메뉴를 클릭한 후 json 파일 생성이 완료되었다면, [그림 2-9]와 같이 콘솔(CLI) 창 오른쪽 상단의 더 보기(...) 메뉴에서 “파일 다운로드(7)” 메뉴를 클릭하고, 팝업창(8)에 생성되어 있는 json 파일명을 입력한 후 다운로드(9) 버튼을 클릭하면 json 파일을 다운로드 할 수 있다(10, 11).

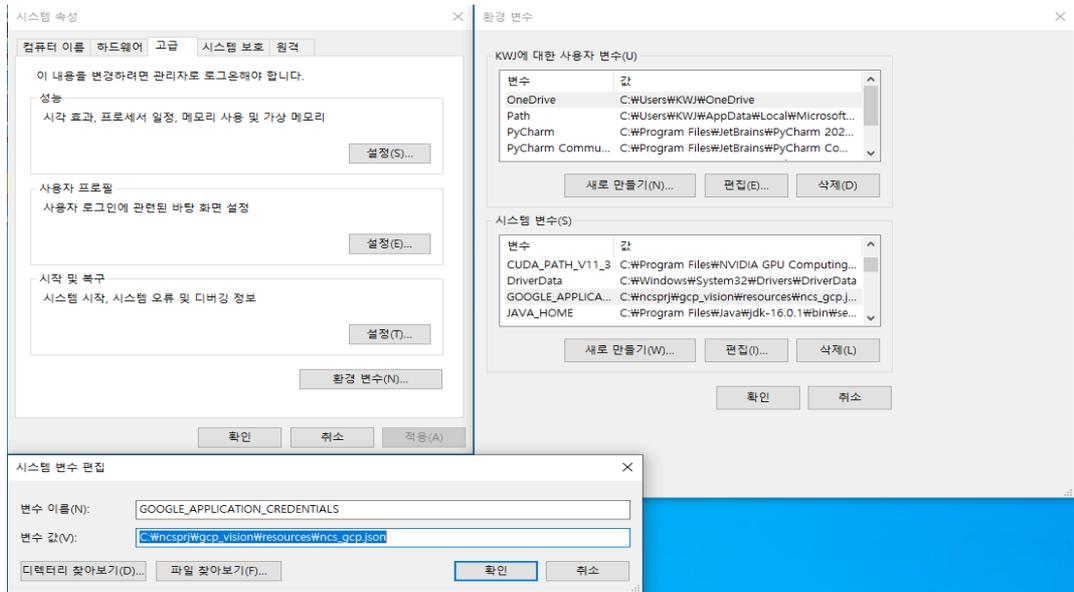


출처: 집필진 제작(2021)

[그림 2-9] 플랫폼 연동 용 “사용자 인증 정보” 파일 생성(7, 8, 9, 10, 11)

##### 5. 플랫폼의 인공지능 연동을 위해 json 구성 정보 설정과 GCP 모듈을 설치한다.

파이참(개발 도구)에서 GCP를 연동하려면 위에서 내려받은 json 파일을 환경 변수로 등록해야 한다. [그림 2-10]과 같이 시스템 환경 변수로 다운로드받은 json 파일의 경로를 등록한다.



출처: 집필진 제작(2021)  
 [그림 2-10] 플랫폼 연동 json 파일의 로컬 컴퓨터 환경 설정

### 시스템 변수 등록 절차

1. 시스템 변수 위치에 "새로 만들기" 클릭
  2. 팝업창의 변수 이름 및 변수값 입력 후 확인 버튼 클릭
- 변수 이름: GOOGLE\_APPLICATION\_CREDENTIALS  
 변수값: 전체 경로 + json 파일명

지금까지의 모든 절차를 빠짐없이 수행했다면 환경 구성이 완료된 것이다. 다음 학습모듈에서 인공지능서비스 플랫폼 연동 구현을 수행한다.

### 수행 tip

- GCP 결제 정보 미입력 시 플랫폼 연동에 제약이 있다.
- 플랫폼 환경 구성에 시행착오가 있을 경우 처음부터 재 시작해도 무관하다.
- 결제 정보 삭제는 다음 학습 모듈의 수행하기 하단에 설명한다.

## 2-2. 플랫폼 연동

### 학습 목표

- 인공지능서비스 플랫폼에서 제공하는 가이드에 따라 애플리케이션 서비스에 플랫폼을 연동할 수 있다.

### 필요 지식 /

#### ① 인공지능서비스 플랫폼의 기술 구조 이해

인공지능 플랫폼의 기반 기술은 “서비스 특화 인공지능 알고리즘”, “빅데이터”, “고성능 컴퓨팅 활용”이다.

##### 1. 서비스 특화 인공지능 알고리즘의 진화

딥러닝과 같은 기계 학습 알고리즘의 진화는 인공지능서비스의 활용 기반 가능성을 확대시키고 있다. 인공지능서비스를 가능하게 하는 여러 가지 기술들을 한 곳에 통합한 뒤 이러한 기술을 다양하게 조합하여 각 서비스 영역에 적용할 수 있도록 구성된 것이 AI 플랫폼이다.

#### 인공지능 기술 요소

Speech To Text: 사람의 음성을 인식해 텍스트로 변환하는 기술

Text To Speech: 텍스트를 다시 음성으로 변환하는 기술

Natural Language Understanding: 언어를 이해하는 기술

챗봇: 질의응답과 대화를 처리하는 기술

이러한 기술 요소가 인공지능 엔진이다. 이 인공지능 엔진들을 토대로 어떻게 모듈을 구성하느냐에 따라 챗봇, 인공지능 비서, 인공지능 스피커 등의 서비스를 구현할 수 있다.

##### 2. 인공지능 플랫폼 활용 환경에 대한 이해

IoT(Internet of Thing) 기술과 스마트폰의 대중적 보급은 데이터양의 급격한 증가, 이른바 빅데이터 환경을 가속화하고 있다. 인공지능 기술의 핵심인 학습은 대량의 데이터 환경이 필수적이며 고품질의 데이터에 따라 알고리즘의 성능이 결정된다.

글로벌 정보 기술 기업들이 인공지능 기술을 선도하는 이유는 수집 데이터의 양과 질에 있어 막대한 자본과 자원을 투입해 기반을 만들기 때문이며, 이러한 데이터는 인공지능을 발전시키는 핵심 동력으로 작용하고 있다.

### 3. 컴퓨팅 성능 향상

그래픽 처리 장치인 GPU는 기존 CPU에 비해 고성능 병렬 처리 기술을 이용해서 복잡한 행렬식이나 벡터 계산이 필요한 기계 학습 연산 수행에서 월등한 성능을 보인다. GPU를 활용한 병렬 처리 등 컴퓨팅 성능 향상이 인공지능 기술의 발전과 지속적인 선순환 구조를 이끌었다.

## ② 인공지능서비스 플랫폼 연동 기술

### 1. 애플리케이션 프로그래밍 인터페이스 기술(Application Programming Interface, API)

컴퓨터 프로그래밍에서 API는 소프트웨어 구축을 위한 서브루틴 정의, 통신 프로토콜 및 도구의 집합이다. 일반적으로 다양한 구성 요소 간의 명확하게 정의된 통신 방법의 구성으로 좋은 API는 프로그래머가 구성하는 모든 빌딩 블록을 제공하여 컴퓨터 프로그램을 쉽게 개발할 수 있도록 한다.

#### 빌딩 블록

빌딩 블록(Building Block)이란 조직의 비즈니스 요구를 만족할 수 있는 기능들의 논리적 집합이다. 즉 각 비즈니스의 요구는 다양한 기능에 의하여 지원되며 기능은 그 목적이나 특성의 유사성에 의하여 하나의 집합으로 정의될 수 있다는 것이다.

정보 기술 아키텍처를 수립하는 과정에서 다양한 기법이 사용되지만 비즈니스나 기술적 문제를 유사한 집합으로 묶고 그 집합에 상징적 의미와 이름을 부여하여 블록화하는 기법이 아키텍팅 과정에서 가장 중요한 기법 중의 하나이다.

### 2. gRPC

오픈 소스 RPC 프레임워크이다. REST와 비교했을 때의 차이점은 HTTP/2를 사용한다는 것과 프로토콜 버퍼로 데이터를 전달하는 것으로 프로토타입(Proto File)만 배포하면 환경과 프로그램 언어에 구애받지 않고 서로 간의 데이터 통신이 가능하다. HTTP/2, ProtoBuf, Proto File로 구성된다.

### 3. GraphQL

Graph + Query Language의 약자로 서비스에 데이터를 질의하기 위해서 만들어졌다. REST의 한계인 Over-Fetching, Under-Fetching을 극복하기 위한 아키텍처로 Schema, Operation, Resolver, Introspection으로 구성된다.

## 수행 내용 / 인공지능서비스 플랫폼 연동하기

---

### 재료·자료

- 요건 분석서(요구 사항 정의서, 요구 사항 명세서) / 아키텍처 정의서
- 인공지능서비스 플랫폼 연동을 위한 인터넷 환경

### 기기(장비·공구)

- 파이선 개발 도구
- 문서 열람 도구(ODF 등)
- 인터넷 브라우저(Chrome 등)

### 안전·유의 사항

- 설계자 또는 이해관계자들과 원활한 의사소통을 할 수 있는 역량을 갖춘다.
- 인공지능서비스 플랫폼 연동을 위한 요구 환경을 갖추 수 있어야 한다.  
(플랫폼 연동을 위한 가입 정보에 따라 비용이 발생할 수 있으므로 주의한다.)
- 인공지능서비스 플랫폼 연동 수행하기가 완료된 경우 “수행 tip”에 제시된 대로 결제 수단을 삭제하여 비용이 발생되지 않도록 조치해야 한다.

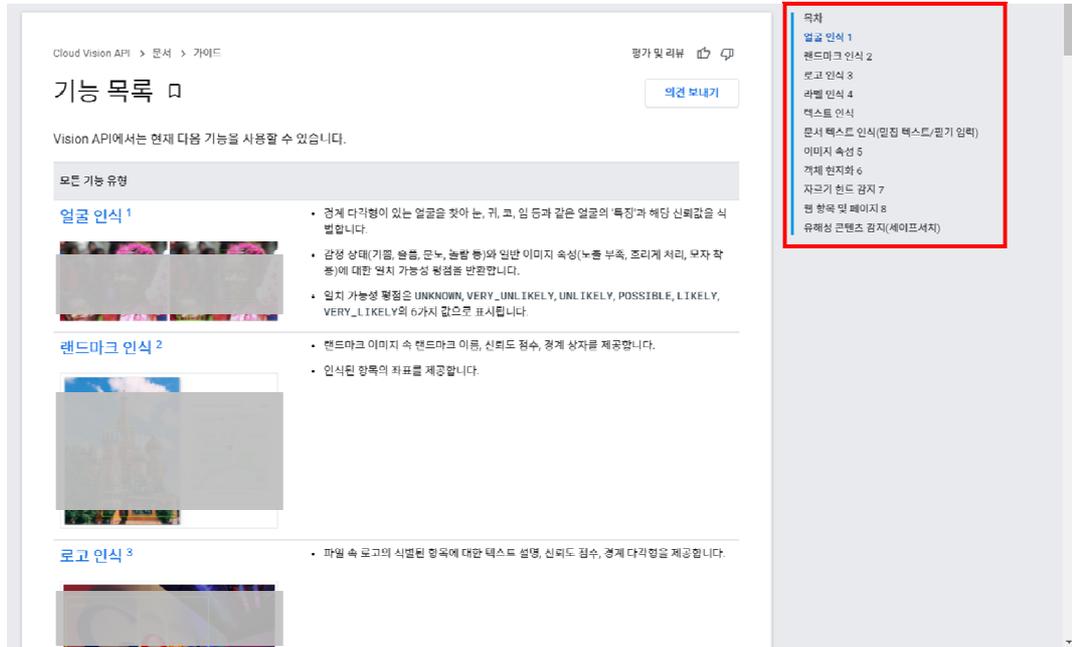
### 수행 순서

#### ① 인공지능서비스 플랫폼 연동을 위해 매뉴얼을 조사한다.

인공지능서비스 플랫폼 연동하기에서는 GCP의 “Vision API”를 사례로 연동 구현을 한다. 따라서 Vision API를 연동하기 위한 플랫폼 기업의 명세를 이해하고 구현할 수 있어야 한다.

#### 1. GCP “Vision API”가 제공하는 기능 목록을 확인한다.

“Vision API”는 얼굴 인식, 랜드마크 인식, 로고 인식, 라벨 인식, 텍스트 감지, 문서 텍스트 인식(밀집/펼기), 이미지 속성, 객체 현지화, 자르기 힌트 감지, 웹 항목 및 페이지 등을 인식하고 정보를 제공하는 인공지능서비스 플랫폼이다.



출처: 집필진 제작(2021)  
[그림 2-11] Vision API - AI 플랫폼 기능 목록

## 2. 인공지능서비스 플랫폼 연동 관련 설계 내용과 플랫폼 연동 구현 방법을 확인한다.

분석, 설계 과정을 통해 인공지능서비스 플랫폼 연동 구현 요건을 상세하게 파악하고, 플랫폼의 인공지능 기능 목록과 비교하여 타깃 기능의 명세와 설계 내용을 구체적으로 이해한다.

### (1) 설계 내용을 확인한다.

인공지능서비스 플랫폼 연동 요건과 요건에 따라 설계한 구체적인 내용을 파악하고 설계 내용에 맞게 구현할 준비를 한다. 언어는 파이썬을 사용하는 것으로 가정하고, 구체적인 요건은 손 글씨를 인식하여 사용자에게 인식한 내용을 서비스 하는 것으로 가정한다.

### (2) 플랫폼 연동 구현 방법을 확인한다.

설계 내용에 따라 가정한 “손 글씨 인식”과 관련하여 “Vision API” 플랫폼에 “문서 텍스트 인식”이라는 기능이 있는 것을 목록에서 확인했다. 해당 기능의 구체적인 명세와 구현 방법을 확인한다.

“손 글씨 인식”과 관련 GCP의 제공 명칭은 “이미지의 필기 입력 감지”이다. 해당 안내 페이지는 “<https://cloud.google.com/vision/docs/handwriting>”이며, 해당 페이지에 구체적으로 정리하고 있다. 해당 페이지의 안내에 따라 구현 예시 코드를 활용하여 “손 글씨 인식” 예제를 구현한다.

### 3. 인공지능서비스 플랫폼을 연동한다.

#### (1) 라이브러리를 설치한다.

파이선으로 “Vision API” 연동 구현을 위해 라이브러리를 설치한다.

#### **파이선(python) 라이브러리 설치**

```
pip install --upgrade google-cloud-vision
```

```
WARNING: Ignoring invalid distribution -pype1 (c:\~~~\lib\site-packages)
WARNING: Ignoring invalid distribution -pype1 (c:\~~~\lib\site-packages)
Requirement already satisfied: google-cloud-vision in c:\~~~\lib\site-packages (2.3.1)
Collecting google-cloud-vision
  Downloading google_cloud_vision-2.3.2-py2.py3-none-any.whl (367 kB)
    | 367 kB 1.3 MB/s
Installing collected packages: google-cloud-vision
  Attempting uninstall: google-cloud-vision
    WARNING: Ignoring invalid distribution -pype1 (c:\~~~\lib\site-packages)
    Found existing installation: google-cloud-vision 2.3.1
    Uninstalling google-cloud-vision-2.3.1:
      Successfully uninstalled google-cloud-vision-2.3.1
  Successfully installed google-cloud-vision-2.3.2
  WARNING: Ignoring invalid distribution -pype1 (c:\~~~\lib\site-packages)
  WARNING: Ignoring invalid distribution -pype1 (c:\~~~\lib\site-packages)

(v_aincs_py38) C:\ncsprj\gcp_vision>
```

#### (2) 플랫폼 연동 코드를 구현한다.

플랫폼 연동 코드 구현을 위한 디렉터리 구조는 다음과 같다. gcp\_vision 프로젝트 하단에 파이선 파일이 위치한다.

#### **플랫폼 연동 코드 디렉터리 설명**

(v_aincs_py38) C:\ncsprj\gcp_vision>tree	- CLI에서 디렉터리 구조 표출 명령어
gcp_vision	- 플랫폼 연동 코드 프로젝트 Root 디렉터리
├ res	- 자원 상위 디렉터리
│ └ imgDir	- 손 글씨 인식을 위한 대상 이미지 디렉터리
│ └ imgRecDir	- 손 글씨 인식 결과 파일 저장 디렉터리
gcp_link.py	- 플랫폼 연동 구현 코드 파일

플랫폼 연동 코드는 GCP 라이브러리를 통해 gRPC 기반 연동을 한다. 실제 연동을 수행하는 코드는 client.text\_detection 부분으로 손 글씨 이미지에 대해 텍스트를 인식하는 소스 코드이다.

#### 플랫폼 연동 코드

```
import io, os
# GCP 연동을 위해 코드 내에 라이브러리를 포함한다.
from google.cloud import vision
# 플랫폼 연동 구현을 위해 클라이언트를 생성한다.
client = vision.ImageAnnotatorClient()
# 손 글씨 인식을 위해 생성해 놓은 이미지를 경로에 추가한다.
file_name = os.path.abspath('res/imgDir/232140210.png')
# 이미지를 읽어서 메모리에 올린다.
with io.open(file_name, 'rb') as image_file:
    contents = image_file.read()
# GCP "Vision API" 인식 대상 클라이언트 이미지 할당
images = vision.Image(content=contents)
# 텍스트 인식 수행 요청(플랫폼 연동 실행 부분)
# 각 글자 단위 인식 및 정확도, 감지 언어 코드 등 특성들을 감지하고 response 구조로 보관
response = client.text_detection(image=images)
# 플랫폼의 응답 구조 중 텍스트 어노테이션 부분 활용
texts = response.text_annotations
# 인식된 텍스트를 출력
print(response.text_annotations[0].description)
```

플랫폼 연동 구현이 잘 되었는지 확인해 본다. 위 코드의 "232140210.png" 이미지는 <표 2-2>와 같이 집필진 본인이 손으로 쓴 글씨이고 플랫폼 연동을 통해 오른쪽과 같이 글씨 인식을 하고 응답을 준 결과를 나타내고 있다.

<표 2-2> GCP "Vision API" 플랫폼 연동 결과 사례

손 글씨 이미지	손 글씨 이미지 인식 결과
<p>&lt;스승의 질문과 제자의 대답&gt;</p> <p>"사람의 입은 하나 인데 귀는 둘이다. 왜 그럴겠는가?"</p> <p>"이야기하는 것보다 더 많이 잘 들어야 한다는 뜻입니다"</p> <p>평범한 것을 매일 평범한 기분으로 행하는 것이 비법입니다.</p>	<p>&lt;스승의 질문과 제자의 대답&gt;</p> <p>"사람의 입은 하나인데 귀는 둘이다. 왜 그럴겠는가?"</p> <p>"이야기하는 것보다 더 많이 잘 들어야 한다는 뜻입니다."</p> <p>평범한 것을 매일 평범한 기분으로 행하는 것이 비법입니다.</p>

위 플랫폼 연동 코드 하단에 `client.text_detection`을 통해 인공지능서비스를 호출하고 `response`로 응답 결과를 받는다. `response`를 통해 결과의 신뢰도를 파악할 수 있다. 다음과 같이 소스 코드를 구현하고 결과를 확인한다.

#### 플랫폼 연동 응답(response) 결과 해석

〈응답 결과 해석을 위한 코딩〉

```
para_txt = ""
space = ""
for block in response.full_text_annotation.pages[0].blocks:
    print(block.property)
    for para in block.paragraphs:
        for wd in para.words:
            for sy in wd.symbols:
                space = " " if sy.property.detected_break.type_ == 1 else ""
                para_txt = para_txt + sy.text + space
    print(para_txt)
para_txt = ""
```

〈응답 결과 해석을 위한 코딩〉

detected\_languages {language\_code: "ko" confidence: 1.0}

〈스승의 질문과 제자의 대답〉

detected\_languages {language\_code: "ko" confidence: 1.0}

'사람의 입은 하나인데 귀는 둘이다. 왜 그럴겠는가?

detected\_languages {language\_code: "ko" confidence: 0.83}

이야기하는 것보다 더 많이 잘 들어야 한다는 뜻입니다."

detected\_languages {language\_code: "ko" confidence: 1.0}

평범한 것을 매일 평범한 기분으로 행하는 것이 비범입니다.

#### 수행 tip

- GCP의 경우 결제 수단을 등록해야 관련 플랫폼 연동을 테스트해 볼 수 있었다. 따라서 위의 테스트가 완료된 경우 "Vision API"의 사용을 중지하고, 왼쪽 상단 메뉴에서 "결제" 메뉴를 클릭, "결제 계정 관리(오른쪽 상단)"에서 "결제 계정 폐쇄"를 클릭한다.
- 마지막으로 "결제 수단" 메뉴에서 "삭제"를 클릭하면 결제가 일어나지 않는다.

## 2-3. 플랫폼 연동 문제 해결

### 학습 목표

- 연동된 인공지능서비스 플랫폼 기능의 정상 작동 여부를 확인하고 문제 발생 시 문제를 해결할 수 있다.

### 필요 지식 /

#### ① 인공지능서비스 플랫폼 연동 문제 해결

인공지능서비스 플랫폼 연동과 관련된 문제는 대부분은 플랫폼 연동 방식에 대한 이해 부족 또는 잘못된 코딩으로 인해 발생하는 문제이다. 오류의 종류와 문제를 해결하기 위해 일반적으로 사용하는 디버깅 방법에 대해 학습한다.

##### 1. 오류의 종류

###### (1) 구문 오류(컴파일 오류)

소스 프로그램 작성 중에 발생하는 오류로 문법(Syntax) 오류를 의미한다. 통상 구문 오류가 가장 일반적인 형태의 오류이며 발생 즉시 코딩 환경에서 쉽게 수정할 수 있다. 프로그램을 점진적으로 개발한다면 오류가 어디에 있는지에 대한 감이 있게 마련이며, 개발자가 컴파일 직전에 추가한 마지막 라인을 집중하게 된다.

###### (2) 런 타임 오류

실행 프로그램의 실행 중에 프로그램이 진행되지 않거나 중단되는 경우의 오류를 문법 오류와 구분하여 실행 오류라고 한다. 프로그래머가 프로그래밍할 때 문제 분석을 철저히 하여 사전에 런 타임 오류를 방지하도록 한다. 대부분의 런 타임 오류는 잘못된 코드를 재작성한 후 컴파일하는 과정을 통해 수정할 수 있다.

###### (3) 논리 오류

실행 프로그램이 의도한 대로 수행되지 않을 때 발생하는 오류를 의미한다. 실행 프로그램을 검사하고 결과를 분석해야만 실행 프로그램이 정확하게 수행되는지 확인이 가능하다. 논리 오류는 오류 발생 원인을 파악하는 데 많은 시간이 필요하기 때문에 해결하기 어려운 형태의 오류이다.

##### 2. 디버깅

###### (1) 디버깅의 개념

디버깅(debugging)은 프로그램 작성과 실행 과정에서 발생된 오류를 수정해 나가는 일련의 과정을 의미한다. 디버깅의 기본적인 원리는 프로그램 소스를 눈으로 따라가며 수행 과정을 상상하여 논리적 오류를 찾아내는 것이다.

이것은 프로그래머의 많은 시간과 노력을 들이게 하며, 특히 본인이 작성한 프로그램의 경우는 절대 오류가 발생하지 않을 것이라는 선입견으로 인해 오류를 찾아내기 힘든 경우가 많다.

## (2) 컴퓨터를 이용한 디버깅 접근 방법

### (가) 프로그램 로직 중간중간 실행 상태 출력 방법

디버깅에 컴퓨터의 도움을 받는 가장 일반적인 방법은 프로그램의 중간중간에 실행 상태를 출력하도록 하는 것이다. 이렇게 함으로써 변수에 잘못된 값이 포함되는지 여부를 프로그래머가 직접 계산하는 것보다 좀 더 쉽게 파악할 수 있다.

### (나) 디버거 등 전문 SW 활용한 디버깅 방법

디버깅 과정에 컴퓨터의 도움을 받는 가장 강력한 방법은 디버거(debugger)와 같은 SW를 사용하는 것이다. 디버거는 주로 원하는 코드에 중단점(break point)을 지정하고 프로그램 실행을 정지하여 메모리에 저장된 값들을 살펴보고 실행 코드를 단계적으로 실행하거나 재개하는 등의 동작을 한다. 디버거 기능이 좋은 소프트웨어들은 메모리 누수 감지, 메모리 충돌 감지, 다중 스레드 관리 등의 기능도 지원한다.

## 수행 내용 / 인공지능서비스 플랫폼 연동 문제 해결하기

---

### 재료·자료

- 요건 분석서 / 비즈니스 로직 설계서 / 아키텍처 정의서
- 인공지능서비스 인터페이스 연동 문제 해결을 위한 개발 도구(파이참 커뮤니티 버전)
- 운영 체제 - 윈도 10 환경

### 기기(장비·공구)

- 컴퓨터
- 프로그램 개발 도구, 디버깅 도구
- 문서 열람 도구(ODF 등)

## 안전 · 유의 사항

- 플랫폼에서 제공하는 연동 매뉴얼을 면밀하게 분석하고 이해해야 한다.

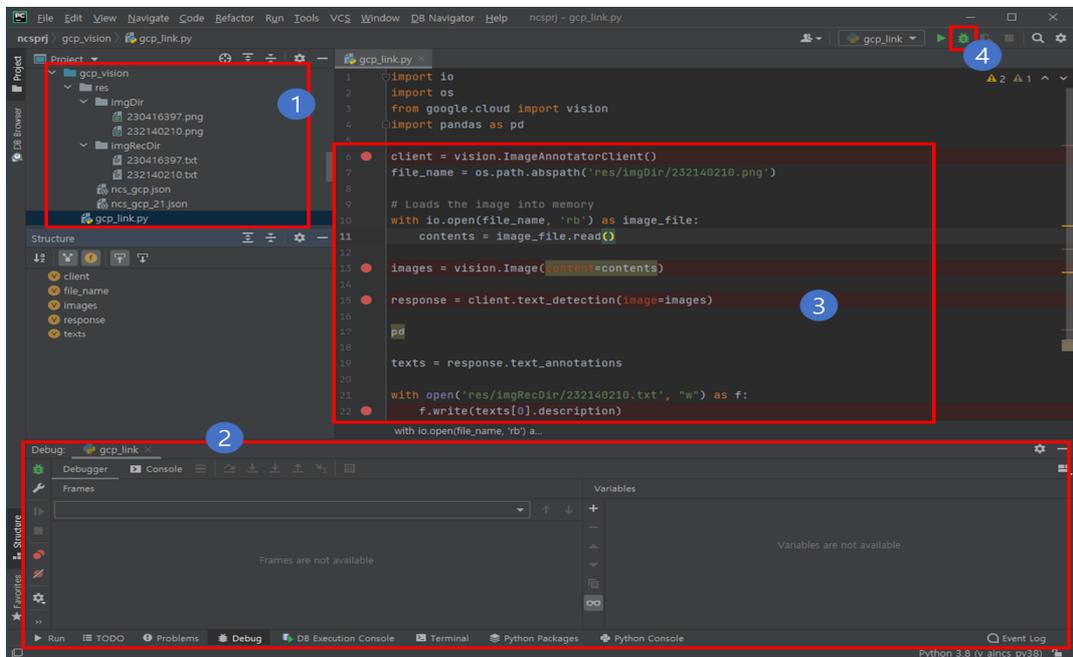
## 수행 순서

### ① 인공지능서비스 플랫폼 연동 코드에 대해 디버깅을 수행한다.

인공지능서비스 플랫폼 연동에서 GCP “Vision API” 예제 코드를 활용하여 집필진이 손 글씨로 작성한 이미지를 인식하는 예제를 수행했다. 해당 코드를 디버깅 예제로 사용하고 다양한 오류를 디버깅을 통해 해결하는 예제를 수행한다.

#### 1. 디버깅을 수행한다.

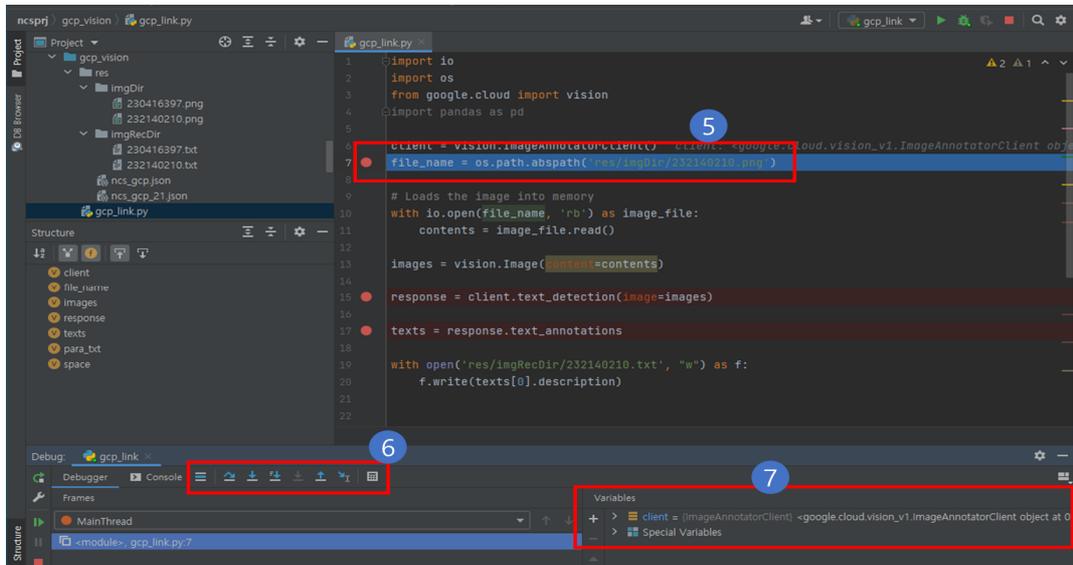
디버깅은 인공지능서비스 플랫폼 연동 예시 코드를 디버깅 대상으로 가정하고 다음과 같이 파이선 프로그램을 디버깅 대상 사례로 예시한다.



출처: 집필진 제작(2021)

[그림 2-12] 파이참(Community 버전) 개발 도구에서 소스 코드 디버깅 - 1

예제 파이선 파일(①)과 파이선 코드의 디버깅이 필요한 곳에 중단점(breakpoint, 그림의 작은 붉은 동그라미 점으로 해당 영역을 클릭하면 표시된다.) 표시(③)를 한다. 이후 ④(Shift+F9)의 디버깅 버튼을 클릭하면 [그림 2-13]의 ⑤와 같이 중단점에서 코드 실행이 멈추게 된다.



출처: 짐필진 제작(2021)  
 [그림 2-13] 파이썬(Community 버전) 개발 도구에서 소스 코드 디버깅 - 2

특히 ②의 Frames창과 Variables창은 개발 도구의 Debug Tool Window의 Add-In창으로 Frames창은 현재 애플리케이션에 사용되고 있는 스택 리스트를 보여 주며, Variables창은 변수 및 값을 보여 준다.

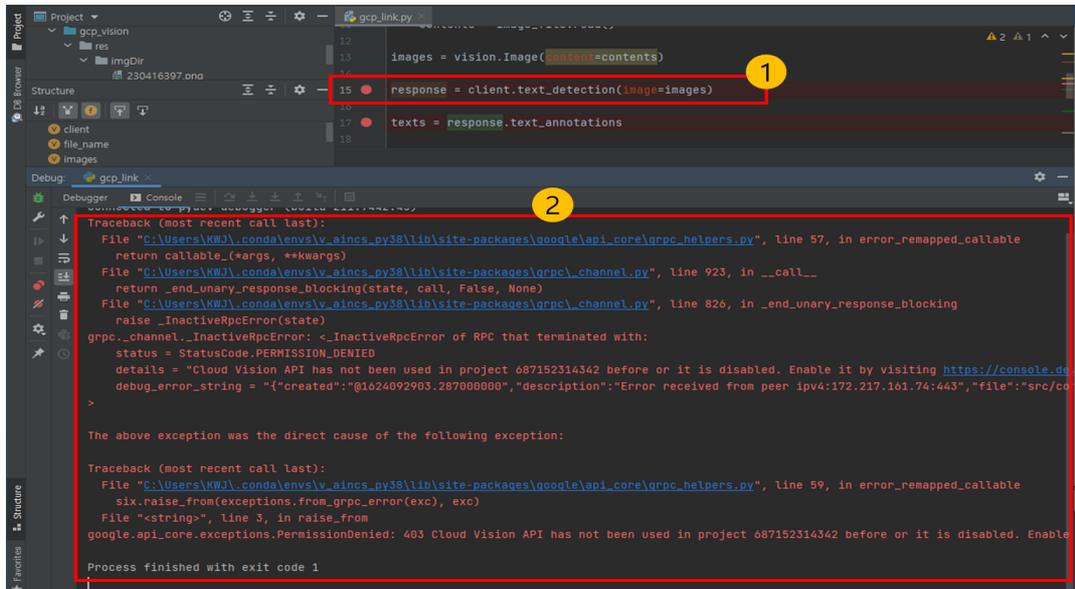
멈추게 된 상황에서 ⑦의 변수 목록에 현재 실행된 첫 번째 중단점 코드 바로 위의 client가 있는 것을 볼 수 있다. 이렇듯 실행된 코드에서 변수는 코드가 실행되면서 목록으로 등록되고 해당 변수값들을 살펴볼 수 있게 된다.

중단점에 멈춰 있는 코드 부분의 재실행 등 디버깅 활동은 ⑥의 디버깅 기능을 통해 추적해 볼 수 있다. <표 2-3>은 [그림 2-13]의 ⑥에 해당하는 디버깅 실행 기능 유형을 정리한 것으로 숙지하면 소스 코드를 개발할 때 디버깅을 효율적으로 진행할 수 있다.

<표 2-3> 디버깅 실행 유형

디버깅 실행 기능	내용	단축키
Step Over	한 줄 실행, 함수가 있어도 실행한 후 다음 줄 실행	F8
Step Into	함수 내부로 들어가 실행	F7
Step Into My Code	프로그래머가 작성한 함수로 들어가 실행	alt+shift+F7
Step Out	함수를 끝까지 실행하고 호출시킨 라인으로 이동	shift+F8
Run to Cursor	커서 위치까지 코드 실행	alt+F9
Evaluate Expression...	특정 변수값 확인	alt+F8

다음은 코드 실행 중에 ① 코드 실행 중 발생한 오류이다. ②의 영역은 오류의 메시지를 상세하게 보여 주는 것으로 해당 영역의 내용을 상세히 살펴보면 오류를 해결할 수 있어야 한다.



출처: 집필진 제작(2021)  
 [그림 2-14] 파이썬(Community 버전) 개발 도구에서 소스 코드 디버깅 오류 사례

위 Console(②)에 나열된 메시지의 제일 끝 하단 내용을 보면 “Vision API”의 사용이 거부되었고, 해당 프로젝트 아이디의 링크로 이동해서 활성화한 후 재시도하라는 메시지이다.

해당 오류는 집필진이 예제를 위해 발생시킨 오류이다. 코드를 실행하기 전, 집필진 본인은 결제 정보 삭제를 위해 API 중지, 결제 계정 닫기, 결제 정보 삭제 등 일련의 과정을 수행한 이후 해당 소스 코드를 실행했다. 따라서 해당 오류 메시지는 정확한 상황을 안내한 것으로 볼 수 있다.

**디버깅 과정에서 오류 해석 방법 사례**

google.api\_core.exceptions.PermissionDenied: 403 Cloud Vision API has not been used in project 68715000342 before or it is disabled. Enable it by visiting <https://~?project=68715000342> then retry. If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry.

**수행 tip**

- 문제 해결을 위해 플랫폼에서 공개하는 오류 명세를 자세하게 살펴보는 일이다.
- 오류 명세에 포함하지 않는 오류가 포함된 경우는 메시지, 오류 위치 등 파악을 위해 디버깅을 수행해 본다.

### 교수 방법

- 인공지능서비스 플랫폼에 대해 설명하고, 플랫폼 연동 방법이 플랫폼 제공 기업별 상이할 수 있으므로 연동 방법(API, RPC, gRPC 등)을 제시한 후 수업을 진행한다.
- 인공지능서비스 플랫폼 연동을 위한 개발 도구를 우선 선정하여 수업을 진행한다.
- 플랫폼 연동 사례로 구글 클라우드 플랫폼(GCP, Google Cloud Platform)을 활용하므로 가입 방법, 결제 수단 등록 방법에 대해 교수가 먼저 수행한 후 시행착오가 발생하지 않도록 지도한다(가급적 교수가 대표로 계정을 등록하고 학생이 실습 환경에서 활용할 수 있도록 구성하는 것을 권장한다.).
- 인공지능서비스 플랫폼 연동 구현에 대해 실습이 이루어지도록 지도한다.
- 설계 내용을 파악하고 이를 개발 언어를 활용하여 작성할 수 있는 절차를 구체적으로 제시한 후 내용에 대해 설명한다.
- 구현된 로직을 테스트할 수 있도록 디버깅 방법을 실습하고 실제 구현 코드에 대해 플랫폼 연동 결과를 해석하고 문제를 해결할 수 있도록 지도한다.

### 학습 방법

- 인공지능서비스 플랫폼에 대해 이해하고, 플랫폼 연동 방식(API, RPC, gRPC 등)을 파악한 후 인터페이스 연동 구현을 실습한다.
- 인공지능서비스 플랫폼 연동을 위한 개발 도구의 사용 방법을 학습한다.
- 수행하기의 인공지능 플랫폼 연동 실습을 위해 플랫폼 연동 기관의 API 가이드를 숙지한 후 연동 실습한다.
- 설계 내용을 파악하고 이를 개발 언어를 활용하여 작성할 수 있도록 학습하고 실제 소스 코드를 구현한다.
- 구현된 로직을 테스트할 수 있도록 디버깅 환경을 구성하고 실제 구현 코드에 대해 플랫폼 연동 부분에 대해 디버깅 및 문제를 해결한다.

## 학습 2      평    가

### 평가 준거

- 평가자는 학습자가 학습 목표를 성공적으로 달성하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습 내용	학습 목표	성취수준		
		상	중	하
플랫폼 연동 환경 구성	- 인공지능서비스 설계에 따라 인공지능서비스 플랫폼 연동 환경을 구성할 수 있다.			
플랫폼 연동	- 인공지능서비스 플랫폼에서 제공하는 가이드에 따라 애플리케이션 서비스에 플랫폼을 연동할 수 있다.			
플랫폼 연동 문제 해결	- 인공지능서비스 플랫폼 기능의 정상 작동 여부를 확인하고 문제 발생 시 문제를 해결할 수 있다.			

### 평가 방법

- 포트폴리오

학습 내용	평가 항목	성취수준		
		상	중	하
플랫폼 연동 환경 구성	- 인공지능서비스 플랫폼 연동 환경을 구성하는 능력			
	- 인공지능서비스 플랫폼 연동을 위해 인터페이스에서 제시하는 연동 관련 안내를 이해하는 능력			
플랫폼 연동	- 인공지능서비스 플랫폼 연동을 위해 인터페이스에서 제시하는 연동 관련 명세(예: 요청/응답/오류 등)를 이해하는 능력			
	- 개발 도구를 이용하여 플랫폼에서 제시하는 구현 사례를 비즈니스 로직에 포함하여 구현할 수 있는 능력			
	- 개발 도구를 활용하여 인공지능서비스 플랫폼 연동을 통해 구현 결과물을 표출하는 능력			
플랫폼 연동 문제 해결	- 인공지능서비스 플랫폼 연동 문제 해결을 위해 개발 도구에 디버깅 환경을 구성하는 능력			
	- 플랫폼에서 제시하는 오류와 관련 명세를 이해하고 문제를 해결할 수 있는 능력			
	- 플랫폼 연동 관련 구현 코드를 디버깅할 수 있는 능력			

• 문제 해결 시나리오

학습 내용	평가 항목	성취수준		
		상	중	하
플랫폼 연동 환경 구성	- 인공지능서비스 플랫폼 연동과 관련하여 교재에서 제시한 플랫폼 외에 타 플랫폼 연동을 위한 환경 구성 능력			
	- 타 플랫폼 연동 환경 구성 사항들을 팀원들에게 공유할 수 있는 능력			
플랫폼 연동	- 손 글씨 인식 연동 외에 다른 인공지능 엔진을 이용하여 서비스를 구성하고 실행할 수 있는 능력			
	- 인공지능서비스 플랫폼 연동과 관련 제시한 플랫폼에서 타 인공지능 API를 팀별 프로젝트로 수행하여 실행 결과를 제시할 수 있는 능력			
	- 예시한 플랫폼 외에 타 플랫폼의 인공지능서비스를 연동하여 실행 결과를 제시할 수 있는 능력			
플랫폼 연동 문제 해결	- 타 플랫폼의 연동 환경을 구성하고 발생된 문제를 해결하여 팀원들과 공유할 수 있는 능력			
	- 타 플랫폼의 인공지능 API를 활용하여 구현하고 발생된 문제를 해결할 수 있는 능력			

• 사례 연구

학습 내용	평가 항목	성취수준		
		상	중	하
플랫폼 연동 환경 구성	- 예시한 플랫폼 외에 타 플랫폼을 조사하고 연동할 수 있는 방법을 제시할 수 있는 능력			
	- 타 연동 환경과 교재의 연동 환경 간 차이점을 설명할 수 있는 능력			
플랫폼 연동	- 파이선 코딩 외에 Java, C++, Node.js, PHP 등 타 언어를 활용하여 인공지능서비스 플랫폼을 연동할 수 있는 능력			
	- 교재에서 제시한 플랫폼 외에 타 플랫폼에서 제시한 사례 코드를 바탕으로 하여 연동 환경을 구현한 사례 학습 능력			
플랫폼 연동 문제 해결	- 디버깅을 통한 문제 해결 사례 조사 및 연구 능력			
	- 다양한 플랫폼에서 제시하고 있는 오류 코드에 대해 조사 및 분석하고 정리할 수 있는 능력			

## 피드백

### 1. 포트폴리오

- 실습 과정을 문제 해결 시나리오에 따라 평가한 후에 개선해야 할 사항 등을 정리한 후 학습자에게 설명한다.
- 설계서를 파악하여 작성된 플랫폼 연동 로직이 설계서대로 개발되었는지 확인하고, 학습자에게 설명한다.
- 설계서를 파악하여 작성된 플랫폼 연동 로직이 설계서대로 개발되었는지 확인하고, 학습자에게 설명한다.
- 평가 결과가 미흡할 때에는 플랫폼 연동 구현 절차 및 방법, 작동 원리에 대해 구체적으로 설명해 준다.
- 성취수준이 낮은 학습자들에게는 실제 구현 코드 라인 단위로 구체적인 설명을 해 준다.
- 성취수준이 높은 학습자의 경우 좀 더 복잡한 인공지능 플랫폼 연동(대화 지능 등)에 대하여 심화 설명해 준다.

### 2. 문제 해결 시나리오

- 제출한 인공지능서비스 플랫폼 연동 로직 구현 결과를 평가한 후에 주요 사항에 대하여 의견을 제시한다.
- 구현된 플랫폼 연동 기능이 정상적으로 동작하는지를 보고 비정상 동작의 원인과 조치 방안에 대해 설명한다.
- 인공지능서비스 플랫폼 연동 로직 구현 결과가 '상'인 경우 모범 사례로 학습자들에게 공유한다.
- 인공지능서비스 플랫폼 연동 로직 구현 결과가 '중'인 경우 부족한 부분을 상세히 설명하고 보완할 수 있도록 한다.
- 인공지능서비스 플랫폼 연동 로직 구현 결과가 '하'인 경우 모범 사례를 참조하여 다시 구현할 수 있도록 유도한다.

### 3. 사례 연구

- 제시한 플랫폼 외에 타 플랫폼을 조사하고 연동할 수 있는 방법을 조사 및 제출하게 하여 평가 및 토론하게 하고 미흡한 사례 조사 내용에 대해 구체적으로 설명해 준다.
- 타 연동 환경과 교재의 연동 환경 간 차이점을 팀원들에게 설명할 수 있는 시간을 갖게 하고 성취수준이 낮은 학습자들에게는 연동 환경 간 차이점을 설명해 준다.
- 파이선 코딩 외에 Java, C++, Node.js, PHP 등 타 언어를 활용한 사례를 조사하고 팀원들과 공유하도록 한다.
- 연동 구현 관련 성취수준이 높은 학습자의 경우 타 언어 활용, 교재의 인공지능 모델 외에 타 모듈 연계 방법에 대해 심화 설명해 준다.
- 문제 해결 방법으로 교재에서 제시한 디버깅 및 응답 오류 코드에 대해 조사 및 분석하고 정리하여 팀원들과 공유하도록 하고, 성취수준이 낮은 학습자들에게는 응답 오류 코드 조사 및 분석 결과에 대해 설명해 준다.

학습 1	인공지능서비스 비즈니스 로직 구현하기
학습 2	인공지능서비스 플랫폼 연동하기
<b>학습 3</b>	<b>인공지능서비스 모델 연동하기</b>
학습 4	인공지능서비스 인터페이스 연동하기

## 3-1. 모델 연동 환경 구성

**학습 목표** • 인공지능서비스 설계에 따라 인공지능서비스 모델의 연동 환경을 구성할 수 있다.

### 필요 지식 /

#### ① 인공지능서비스 모델과 모델 선정

##### 1. 인공지능 모델

인공지능에서 모델(Model)이란 특정 유형의 패턴을 인식하도록 학습시키고 그 결과로 나타난 결과물을 의미한다.



출처: 집필진 제작(2021)  
[그림 3-1] 인공지능 모델의 개념

모델을 학습한 후에는 이를 사용하여 이전에 입력하지 않은 데이터를 추론하고 해당 데이터에 대한 예측을 만들 수 있다. 예를 들어 얼굴 표정에 따라 사용자의 감정을 인식하는 애플리케이션을 구현하려는 경우, 특정 감정으로 각각 레이블(Label)이 지정된 얼굴 이미지를 학습한 모델은 새로운 사람의 이미지를 입력받아 감정을 추론하는 데 이용할 수 있다.

## 2. 인공지능 모델 선정

인공지능 모델을 처음부터 학습하는 데 많은 시간과 노력이 필요하다. 이러한 시간 및 자원 한계에 따라 여러 인공지능 프레임워크는 다양한 인공지능의 모델 연동 방법과 재학습(전이 학습, Transfer Learning)을 위한 방법을 제공하고 있다.

### 전이 학습(Transfer Learning)의 개념

공통점을 이용한 지도 기반 학습 방법으로 기존 학습된 비슷한 모델의 하위층(lower layer)을 가져와 재사용함으로써 학습 속도를 빠르게 진행하고, 학습에 필요한 훈련 세트(Training set)도 훨씬 적게 이용되는 것이 특징이다.

## ② 인공지능서비스 모델 연동 환경 구성

인공지능서비스 모델을 연동하기 위해 인공지능 프레임워크를 많이 활용한다. 인공지능 프레임워크에는 파이토치, 텐서플로, 케라스 등 다양하다.

특히 인공지능 프레임워크는 인공지능 모델 생성의 제약 요소인 대규모의 학습 데이터와 대규모 HW 자원, 시간(하이퍼파라미터를 통한 튜닝 등)의 어려움 해소를 위해 주요 인공지능 모델에 대해 전이 학습의 방법을 제공하고 있다.

## 수행 내용 / 인공지능서비스 모델 연동 환경 구성하기

### 재료·자료

- 요건 분석서 / 비즈니스 로직 설계서 / 아키텍처 정의서
- 인공지능서비스 로직 구현을 위한 개발 도구  
(인터넷 가능한 컴퓨터 환경, 설치 파일을 다운로드받아야 하는 경우 대비)
- 운영 체제 - 윈도우 10 환경

### 기기(장비·공구)

- 컴퓨터
- 문서 열람 도구(ODF 등)

## 안전 · 유의 사항

- 설계자 또는 이해관계자들과 원활한 의사소통을 할 수 있는 역량을 갖춘다.
- 개인 컴퓨터(PC) 환경에서 GPU 기반 개발 환경을 구성할 것인지 판단한다.

## 수행 순서

### ① 인공지능서비스 모델 연동을 위한 환경을 구성한다.

인공지능서비스 모델 연동 환경에는 <표 3-1>과 같이 텐서플로, 케라스 및 파이토치 등이 있다.

<표 3-1> 인공지능 프레임워크 종류

프레임워크 종류	설명	비고(라이선스 종류)
텐서플로	구글이 2015년에 오픈 소스로 공개한 기계 학습 라이브러리이다. 데이터 플로 그래프 지원, 계산 구조와 목표 함수만 정의하면 자동으로 미분 계산 처리, 파이선, C++, Go, Java, R을 지원한다.	전이 학습 가능 (Apache2.0 Lic)
파이토치	페이스북 인공지능 연구팀에 의해 주로 개발된 딥러닝 라이브러리이다. 넘파이(NumPy)의 배열을 텐서(Tensor)로 가져올 수 있고, 이를 쉽게 GPU상에 올려 연산이 가능하다.	전이 학습 가능 (BSD Lic)
MXNet	가볍고, 이식성이 좋으며, C++, 파이선, Matlab, Julia, Javascript, Go, R을 지원하며, Linux, Windows, IOS, Android 등 플랫폼을 지원한다.	전이 학습 가능 (Apache2.0 Lic)
테아노(Theano)	파이선 기반의 딥러닝 라이브러리로 CPU 및 GPU의 수치 계산에 매우 유용하다.	(BSD Lic)
케라스(Keras)	테아노(Theano) 또는 텐서플로에서 동작하도록 구성할 수 있다. 파이선 기반으로 작성되었으며 매뉴얼이 잘 되어 있다.	전이 학습 가능 (MIT Lic)
체이너(Chainer)	define-by-run 원칙에 의해 설계된 딥러닝 프레임워크로 forward 함수만 정의해 주면 네트워크 구조가 자동으로 정해진다.	(MIT Lic)
CNTK	딥러닝 모델의 교육을 목적으로 하는 오픈 소스 딥러닝 도구로 파이선과 C++ 등의 언어를 지원한다. Cognitive Toolkit 사용으로 효율적인 강화 학습 모델, GAN(Generative Adversarial Networks)을 구현할 수 있다.	(MIT Lic)

지난 “학습1-1 인공지능서비스 개발 환경 구성하기”에서 이미 텐서플로 및 케라스는 구성을 하였으므로 이번 장에서는 파이토치 환경을 구성한다. 파이토치는 산업 현장에서 많이 이용하는 딥러닝 프레임워크이다. 파이토치 설치를 위해 Anaconda Prompt에 접속하여 새로운 가상 환경(aincs\_pt\_py38)을 만든다.

다음과 같이 가상 환경을 생성하고 파이토치를 설치할 가상 환경을 활성화해 놓는다.

### 가상 환경 생성 및 접속 명령어

```
(base) c:\ncs_prj> conda create -n aincs_pt_py38 python=3.8
(base) C:\ncs_prj\aincs_pt_py38> conda activate aincs_pt_py38
```

### 파이토치

파이토치(PyTorch)와 텐서플로는 산업 현장에서 많이 이용하는 딥러닝 프레임워크이다. 텐서플로보다 사용하기 쉽고, 직관적인 구조에 따라 사용자가 지속적으로 늘어나는 추세이다.

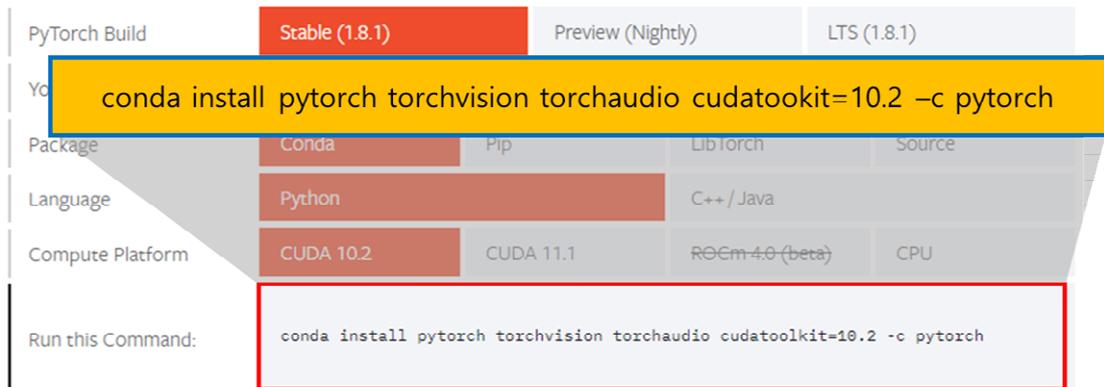
텐서플로는 Auto ML 등의 방대한 라이브러리 및 GCP(Google Cloud Platform) 클라우드 호환이라는 나름 고유의 장점이 있고 2.0 이후로는 Define-By-Run 방식을 채택하고 있어 파이토치와 기능상 큰 차이가 없지만 대부분의 오픈 소스 및 레퍼런스들이 1.x 버전 대의 코드로 공유되는 것이 큰 단점이다.

<표 3-2> 인공지능 프레임워크 종류 비교(텐서플로 vs 파이토치)

구분	텐서플로	파이토치
패러다임	Define and Run	Define by Run
그래프 형태	정적 그래프	동적 그래프

- 정적 그래프: 매 반복 단계에서 기존에 구축된 정적인 동일한 연산 그래프에 러닝(running)하는 방식으로 동작
- 동적 그래프: 동작할 때 매번 새로운 그래프를 만들게 되어 다양한 조건에 대해 대응 가능하므로 유연한 모델 제작에 유용

<https://pytorch.org/>에 접속한 후, “Install” 버튼을 클릭하면 [그림 3-2]와 같이 환경을 선택하는 화면이 나타난다.



출처: 집필진 제작(2021)  
[그림 3-2] 파이토치 설치 명령어 생성(<https://pytorch.org/>)

개인 컴퓨터의 환경 및 버전을 체크하면 install 명령어(Run this Command)가 자동 완성되고, 해당 명령어와 가상 환경을 결합하여 파이토치를 설치한다(명령어 실행). 이외에도 주피터 노트북 등 관련 라이브러리를 추가로 설치한다.

### 파이토치, 주피터 노트북 등 설치 명령어

```
(aincs_pt_py38) c:\ncs_prj\aincs_pt_py38> conda install pytorch torchvision torchaudio
cudatoolkit=10.1 -c pytorch -n ainc_s_pt_py38
(aincs_pt_py38) c:\ncs_prj\aincs_pt_py38> conda install -n ainc_s_pt_py38 jupyter notebook
ipython
```

파이토치 환경이 잘 설치되었는지 확인한다.

### 파이토치 환경 구성 여부 확인

```
import torch
import tensorflow
from tensorflow import keras
print('torch ' + torch.__version__)
print('tensorflow ' + tensorflow.__version__)
print('keras ' + keras.__version__)
```

위와 같은 파이썬 코드 실행하면 다음과 같은 결과가 나타나면 된다.

### 파이토치 환경 구성 여부 확인

```
torch 1.8.#
tensorflow 2.3.#
keras 2.4.#
```

- ② 인공지능서비스 모델 연동을 위해 파이토치 프레임워크에서 모델을 불러오기 한다.  
파이토치의 경우 다음과 같은 모델들을 연동하도록 지원하고 있다.

```
resnet18 = models.resnet18(pretrained=True)
alexnet = models.alexnet(pretrained=True)
squeezenet = models.squeezenet1_0(pretrained=True)
vgg16 = models.vgg16(pretrained=True)
densenet = models.densenet161(pretrained=True)

inception = models.inception_v3(pretrained=True)
googlenet = models.googlenet(pretrained=True)
```

```
shufflenet = models.shufflenet_v2_x1_0(pretrained=True)
mobilenet = models.mobilenet_v2(pretrained=True)
resnext50_32x4d = models.resnext50_32x4d(pretrained=True)

wide_resnet50_2 = models.wide_resnet50_2(pretrained=True)
mnasnet = models.mnasnet1_0(pretrained=True)
```

해당 모델 중에서 특정 하나의 모델을 연동하여 모델을 불러오기 한다.

#### 파이토치의 모델 연동을 위해 특정 모델을 불러오기 하는 예시

```
import torchvision.models as models
model = models.inception_v3(pretrained=True)
print(model)
```

#### 수행 tip

- 인공지능서비스 모델 연동을 위해 예제 코드를 모두 실행해 본다.

## 3-2. 모델 연동

### 학습 목표

- 인공지능서비스 모델이 제공하는 다양한 형상화 기술에 따라 애플리케이션 서비스에 인공지능 서비스 모델을 연동할 수 있다.

### 필요 지식 /

#### ① 인공지능서비스 모델 연동 요건 및 설계 파악

##### 1. 인공지능서비스 모듈 연동을 위한 모듈(Module) 명세 이해

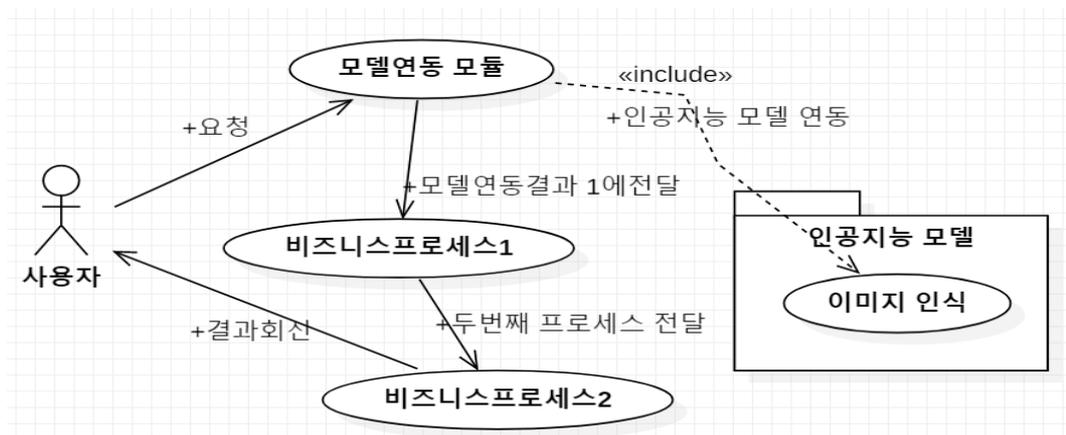
모듈의 명세는 <표 3-3>과 같은 원칙에 따라 기술되어 있으므로 인공지능서비스 모듈 연동을 위해 해당 원칙에 기반해서 명세를 확인하고 관련자들과 적극적으로 의사 소통해야 한다.

<표 3-3> 인공지능서비스 모델 연동 구현 시 확인해야 하는 모듈 명세 기준

주요 원칙	설명
정확성(Correctness)	해당 기능을 정확하게 작성
명확성(Clarity)	해당 기능이 일관되게 이해되어야 하고 한 가지의 의미로 해석되도록 작성
완전성(Completeness)	시스템이 구현될 때 필요하고 요구되는 모든 것을 기술
일관성(Consistency)	공통 기능들 간에 상호 충돌이 없도록 작성
추적성(Traceability)	공통 기능에 대한 요구 사항 출처와 관련 시스템 등의 유기적 관계에 대한 식별이 가능하도록 작성

##### 2. 인공지능서비스 비즈니스 흐름 파악

비즈니스 흐름은 유스케이스 다이어그램, 유스케이스 명세서 등을 통해 파악하고, 인공지능서비스의 역할도 파악한다.



출처: 집필진 제작(2021)

[그림 3-3] 비즈니스 흐름 파악 예시(유스케이스 다이어그램)

유스케이스 예시에 따르면 인공지능서비스에서 모델 부분은 핵심 모듈(include 관계)이다. 모델 연동 후 해당 결과를 바탕으로 하여 “비즈니스 프로세스 1”, “비즈니스 프로세스 2”를 수행하고 최종 비즈니스의 의미 있는 결과를 사용자에게 전달하는 흐름이다.

### 유스케이스 다이어그램

액터: 시스템 외부에 존재하는 것으로 시스템과 상호 작용하는 것을 나타냄.

유스케이스: 시스템이 어떤 서비스 또는 기능을 제공하는지를 명세

관계: 액터(actor)와 유스케이스(usecase), 유스케이스 간의 연결로 나타남.

1. 포함(include) 관계 - 해당 유스케이스를 실행하기 위해 반드시 필요한 유스케이스
2. 일반화 관계 - 유사한 유스케이스, 유사한 액터 등 그룹을 만들어 이해를 높이기 위한 관계
3. 확장(extend) 관계 - 확장 기능 유스케이스와 확장(선택 가능한) 대상 유스케이스 사이에 형성되는 관계
4. 연관 관계 - 유스케이스와 액터 간 상호 작용이 있음을 표현

유스케이스 다이어그램을 확인했다면, 유스케이스 명세서도 구체적으로 이해할 수 있어야 한다. 유스케이스 명세서는 다음과 같이 다양한 구성을 포함한다.

〈표 3-4〉 유스케이스 명세서의 구성 요소

구성 요소	설명
유스케이스명	액터가 시스템을 대상으로 달성해야 할 목적을 하나의 문장으로 명확하게 기술
액터명	시스템에서 수행하는 역할 이름, 실제 사람의 이름이나 시스템의 이름은 사용하지 않음.
개요	유스케이스를 수행하는 개요를 기술
사전 조건 (pre-condition)	유스케이스의 기본 흐름(Basic Flow)이 올바르게 동작하기 위해 유스케이스 실행 이전에 충족되어야 하는 조건을 구체적으로 기술
사후 조건 (post-condition)	유스케이스가 실행된 후에 만족해야 하는 조건을 구체적으로 기술
기본 흐름 (basic-flow)	시스템과 액터 사이의 목적 달성을 위해 기본적인 상호 작용 흐름을 명세 명세한 기본 흐름을 수행할 때에는 오류나 예외가 발생하지 않고 모든 흐름이 완전하게 수행되는 것을 전제 기본 흐름의 첫 번째 단계는 해당 유스케이스를 시작하는 사건을 기술(트리거)
대체 흐름 (alternative-flow)	경우에 따라 기본 흐름으로부터 선택적으로 실행되고 다시 기본 흐름으로 돌아오는 흐름, 오류, 예외 발생에 따른 처리 흐름 등을 명세

## ② 인공지능서비스 모델 연동을 위한 모델 선정과 전이 학습 기술 이해

### 1. 인공지능 모델 선정

인공지능 모델을 처음부터 학습하는 것은 시간과 노력이 많이 드는 일이다. 많은 데이터와 학습을 위한 시간이 필요하다. 이러한 시간 및 자원 한계에 따라 여러 인공지능 프레임워크는 다양한 인공지능의 모델 연동 방법과 재학습(전이 학습, Transfer Learning)을

위한 방법을 제공하고 있다. <표 3-5>는 인공지능 분야별 프레임워크에서 제공하는 인공지능 모델을 예시하였다.

<표 3-5> 인공지능 분야별 인공지능 모델 종류

AI 분야	프레임워크	사용 가능 모델	내용
비전(Vis ion)	파이토치(py torch)	resnet18	18개 계층으로 구성된 컨벌루션 신경망으로 ImageNet 데이터베이스의 1백만 개가 넘는 영상에 대해 훈련된 신경망의 사전 훈련된 버전의 모델을 사용 가능
		resnext50_32x4d	50개 계층으로 구성된 컨벌루션 신경망, 1,000개의 사물에 대해 분류 가능
		alexnet	2012년에 개최된 이미지넷 챌린지에서 사용된 알고리즘으로 8개의 레이어로 구성되었는데 5개의 컨볼루션 레이어, 3개 full-connected 레이어로 구성
		squeezenet1_0	alexnet에 비해 50배 정도 정확도 향상, 경량 AI 모델
		vgg16	옥스퍼드 대학의 연구팀 VGG에 의해 개발된 모델로 16개의 층으로 구성
		densenet161	특징 맵의 흐름을 개선시키고, 이전 레이어의 입력을 그대로 전송시켜 줌으로써 기울기 소실 문제 해결한 AI 모델
		inception_v3	VGG 모델을 제치고 1등을 차지한 모델로 컨볼루션 레이어 연결 구조와 행렬 연산을 개선
자연어	google	bert	컴퓨터 성능이 제한되거나 배터리 성능이 중요한 디바이스에서 사용될 목적으로 설계된 AI 모델
		kobert2	자연 언어 처리(NLP) AI 답러닝 모델
		kobert	BERT의 한국어 성능 한계를 극복하기 위해 개발(위키피디아, 뉴스 등의 문장으로 학습)
		gpt-3	한국어 답러닝 언어 모델
		kogpt-2	오픈 AI에서 개발한 자연어 처리 모델로 5천억 개 단어를 학습한 자연어 처리 인공지능 모델
		kogpt-2	한국어로 학습된 오픈 소스 기반 모델
		roberta	BERT의 하이퍼파라미터 및 학습 데이터 사이즈 등을 조절하여 bert의 성능을 극복한 자연어 모델

## 2. 전이 학습 기술 이해

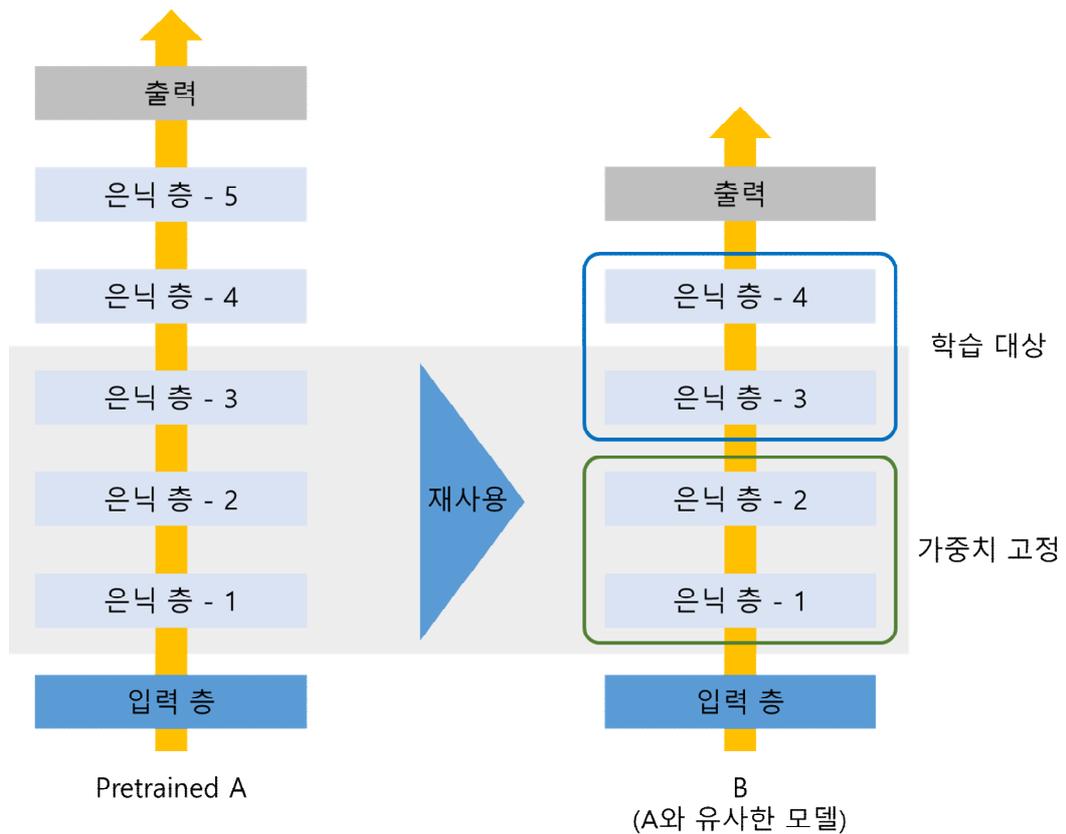
전이 학습이란 유사 이미지 분류 기술을 구현하는 데 특정 도메인에서 얻은 모델을 다른 도메인에 적용하는 기술이다. 이미 학습된 모델의 가중값을 가져와서 분류하고 싶은 다른 데이터 세트에 적용하는 것으로 이해할 수 있다.

잘 알려진 프레트레이드 모델(pretrained model, <표 3-5>의 각 모델)을 이용해서 마지막 파이널 레이어(final layer)만 수정해서 본인의 목적에 맞게 다시 학습시키는 방법(Fine tuning)을 전이 학습이라고 한다.

## 전이 학습(Transfer Learning)의 필요성

‘해결하려는 문제’에는 소량의 정답만이 존재하는 반면, “해결하려는 문제”와 “유사한 문제”에는 대규모의 정답이 존재할 경우 이용할 수 있는 지도 기반 학습(Supervised Learning)으로 공통점을 이용한 학습 방법이다.

대규모의 딥러닝 모델을 새롭게 학습시키는 경우에 학습 속도가 현저하게 느린 문제가 발생하는데, 이러한 문제를 해소할 방법으로 기존 학습된 비슷한 모델의 하위층(lower layer)을 가져와 재사용함으로써 학습 속도를 빠르게 할 수 있을 뿐만 아니라 학습에 필요한 훈련 세트(Training set)도 훨씬 적게 이용된다.



출처: 집필진 제작(2021)  
[그림 3-4] 전이 학습의 개념

전이 학습의 가장 큰 특징은 ‘적은 데이터 세트를 사용하여 모델을 학습’, ‘전이 학습한 모델이 처음부터 쌓은 모델보다 뛰어난 성능을 보임’, ‘시간 절약’으로 볼 수 있다.

## 수행 내용 / 인공지능서비스 모델 연동하기

### 재료·자료

- 요건 분석서 / 인공지능서비스 인터페이스 연동 명세서
- 인공지능서비스 모델 연동 구현을 위한 개발 도구
- 운영 체제 - 윈도 10 환경

### 기기(장비·공구)

- 문서 열람 도구(ODF 등)
- CASE 도구(SW 설계 지원 도구, SW 개발 지원 도구, SW 테스트 지원 도구 등)

### 안전·유의 사항

- 설계자 또는 이해관계자들과 원활한 의사소통을 할 수 있는 역량을 갖춘다.
- 인공지능서비스 인터페이스 연동 구현을 위한 명세서(설계서)를 이해할 수 있어야 한다.

### 수행 순서

#### ① 인공지능서비스 모델 연동을 위해 설계서를 확인한다.

사전 설계자들에 의해 구성된 설계서를 연동의 범위를 확인한다. 설계서에 비즈니스 목적 달성을 위해 “이미지 분류를 위한 인공지능 모델 연동”이 필요한 것으로 가정하고, 이미지 분류와 관련되어 “resnet18”을 연동을 위해 선택된 모델이라고 가정한다.

#### ② 인공지능서비스 모델 연동을 위해 코드를 구현한다.

##### 1. 인공지능 모델 연동을 위해 코드를 구현한다.

파이토치는 다양한 모델 사용을 지원한다. resnet18(pretrained=True)이라는 한 줄의 소스 코드만으로도 이전에 학습이 완료된 모델을 활용할 수 있다. 다음 코드를 통해 상세하게 이해한다. 다음 코드는 모델 연동을 위한 라이브러리 포함 코드이다.

#### 파이선(python) 구현 예제 - 모델 연동을 위한 라이브러리 포함

```
import torch, torch.nn.functional as F
from PIL import Image
from torchvision import transforms
from torchvision import models
```



이미지 인식을 위해 사전에 준비된 이미지를 메모리에 로딩하고, resnet 모델 연동을 위한 이미지 전처리를 한다. 토치비전(torchvision)의 트랜스폼스(transforms)는 이미지의 변환 기능을 제공하며, 예제는 Resize, CenterCrop, ToTensor, Normalize를 수행한다.

unsqueeze는 resnet 모델이 여러 이미지를 하나의 차원으로 입력하도록 구성됨에 따라 임시 배치 차원을 추가하는 용도의 소스 코드이다.

#### 파이선(python) 구현 예제 - 인식 대상 이미지의 메모리 로딩 및 전처리

```
input_image = Image.open("imgDir/170345022.jpg")
preprocess = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.4850, 0.4560, 0.4060], std=[0.2290, 0.2240, 0.2250]),
])
input_tensor = preprocess(input_image)
# 임시 미니 배치 차원 생성
input_batch = input_tensor.unsqueeze(0)
```

#### 트랜스폼스(transforms)의 세부 옵션

Resize: 이미지 크기 조정(resnet은 224, 256, 384, 480, 640 크기 지원)  
 CenterCrop: 가운데 부분을 숫자 크기로 자름  
 ToTensor: 이미지 데이터를 Tensor로 변환  
 Normalize: 이미지는 촬영된 주변의 환경에 따라 명도나 채도가 다르기 때문에 해당 resnet에 학습한 이미지들과 동일한 환경으로 맞춰 주는 정규화 과정 필요  
 ImageNet의 RGB 픽셀 평균 및 표준 편차는 [0.485, 0.456, 0.406] 및 [0.229, 0.224, 0.225]이므로 해당 평균과 표준 편차 사용

모델을 연동한 추론(학습이 아닌 경우)에 torch.no\_grad() 문장으로 감싸 autograd를 끄으로써 메모리 사용량을 줄이고 연산 속도를 높이는 코드로 구현한다. resnet 모델은 정규화되지 않은 1,000개(출력 분류 개수)의 벡터값을 output으로 내보낸다. 이러한 벡터값을 0~1 사이의 값으로 정규화하여 1,000개 클래스에 대한 확률값(probabilities)으로 저장한다.

#### 파이선(python) 구현 예제 - 모델에 이미지를 입력하고 출력값 확인하기

```
with torch.no_grad():
    output = model(input_batch)

probabilities = F.softmax(output[0], dim=0)
```

## 오류 역전파

〈오류 역전파〉 출력층 오차에 대한 가중값 및 편향의 기울기를 구하여 각 층의 가중값 및 편향값을 보정하고, 각 층의 입력값에 대한 기울기를 구해 왼쪽 이전층의 출력 경사값으로 공급하여 역전파한다. 역전파는 많은 파라미터와 여러 개의 레이어 상황에서 가중값  $w$ 와 편향  $b$ 를 학습시키기 어려웠던 문제를 해결하였다.

〈no\_grad()에 대한 이해〉 파이토치의 autograd는 Tensor로 수행한 모든 연산에 대하여 자동-미분(Automatic differentiation) 기능을 제공하는 프레임워크이다. autograd는 실행 시점에 정의되는(define-by-run) 프레임워크로 코드가 어떻게 실행되는가에 따라서 역전파가 정의된다는 것을 의미한다.

위 코드에 따라 output과 probabilities는 〈표 3-6〉과 같은 예시의 값들이 포함된다. 해당 결과에 따르면 “acoustic guitar”가 98.3264% 일치하는 것으로 인식하였다.

〈표 3-6〉 output과 probabilities, class 산출값 예시

개수	output[0]	probability	class
401	4.8416609764099100	0.000643093517282978000%	academic gown
402	2.1735975742340000	0.000044621958750212800%	accordion
403	16.7791748046875000	98.3264386653900000000000%	acoustic guitar
404	0.3991676568984980	0.000007567000182007180%	aircraft carrier

사전 준비된 이미지는 [그림 3-5]와 같으며, 인공지능은 “기타”라는 것을 정확하게 분류해 낸 것으로 확인하였다.



출처: 집필진 제작(2021)  
[그림 3-5] 인식 대상 이미지

순서	벡터값	확률값	클래스
1	16.77	98.32%	acoustic guitar
2	11.80	0.68%	electric guitar
3	10.29	0.15%	pick
4	10.23	0.14%	bow tie
5	10.01	0.11%	broom

## softmax 함수란

입력받은 값을 0~1 사이의 출력값으로 모두 정규화하고 출력값들의 총합이 항상 1이 되는 특성을 가진 함수로 분류하고 싶은 클래스의 개수만큼 출력층의 개수로 구성한다. 결괏값에서 가장 큰 출력값을 가진 클래스가 확률이 가장 높은 것으로 이용된다.

resnet에서 분류하는 1,000개의 카테고리 명칭(imagenet\_classes.txt)을 가져오고, softmax를 통해 확률값으로 보관된 probabilities에서 topk(5)만큼을 확률값과 카테고리값을 가져온다.

### 파이선(python) 구현 예제 - 모델의 카테고리 명칭 가져오기 및 topK 카테고리 인식

```
with open("imagenet_classes.txt", "r") as f:
    categories = [s.strip() for s in f.readlines()]
top5_prob, top5_catid = torch.topk(probabilities, 5)
for i in range(top5_prob.size(0)):
    print(categories[top5_catid[i]], top5_prob[i].item())
```

여기까지 모델 연동 구현 코드를 살펴보았다. 비즈니스 흐름 내에서 모델 연동 부분을 모듈화하고, 해당 모델의 산출값에 따라 이후 비즈니스 흐름을 구현하면 된다.

### ③ 인공지능서비스 모델 연동을 위해 전이 학습 코드를 구현한다.

#### 1. 인공지능 모델에 대해 전이 학습이 필요한지 파악한다.

사전 학습(pretrained)이 완료된 모델의 전이 학습이 필요한 이유는 사전 학습된 모델이 인공지능서비스에서 서비스할 대상(예: 분류)을 포함하고 있지 않을 경우, 또는 사전 학습된 모델의 일반화 능력이 방대하여 모델의 일반화 수준을 낮추고 도메인에 적합한 인공지능서비스를 제공하는 용도로 분류 수준의 축소가 필요한 경우이다.

앞에서 사례로 구현한 인공지능서비스 모델 연동은 resnet18 기반으로 1,000개의 사물을 분류하는 인공지능 모델을 연동하였다. 이번 전이 학습 코드 구현하기에는 설계서에 지정된 3개의 사물 분류 모델로 전이 학습이 필요한 것을 가정하여 구현한다. 따라서 전이 학습의 필요 이유에 대해 설계서를 통해 파악한 결과 “3개의 사물을 정확도 99% 이상으로 분류”하는 것이 명세되었다고 가정하고 전이 학습을 위한 소스 코드를 구현한다.

#### 2. 인공지능 모델의 전이 학습 코드를 구현한다.

1,000개의 사물을 분류하는 사전 학습이 완료된 인공지능 모델을 3개의 사물을 인식하는 모델로 변경하기 위해서는 재학습 관련된 소스 코드의 구현이 필요하다. 해당 소스 코드를 지면에 모두 포함하기에는 양이 많으므로 전체 코드는 부록에 포함하고, 수행하기에는 절차와 핵심 내용 위주로 설명하므로 자세한 사항은 부록 3-2를 참고하며 구현한다. 전이 학습은 <표 3-7>과 같은 절차로 수행한다.

〈표 3-7〉 인공지능 모델 전이 학습 절차

절차	설명	비고
AI 학습데이터 준비	학습 데이터, 검증 데이터 등 학습용 데이터 세트를 준비한다.	한국 차량 이미지 3종을 분류, 인식하는 것을 예시로 함
학습 및 검증 모듈 구현	에폭, 학습률, 옵티마이저, 배치 등을 적용하는 학습 모듈을 구현한다.	모듈화하여 구현 1. 에폭 수만큼 학습 2. 학습 최적화 등
학습 및 검증 모듈 호출 전처리 구현	<ol style="list-style-type: none"> <li>1. 사전 학습된 인공지능 모델 다운로드</li> <li>2. 마지막 계층(분류하는 계층, resnet의 경우 1,000개)을 분류 개수(3개)로 초기화</li> <li>3. 손실 함수 적용</li> <li>4. 최적화 함수 적용</li> <li>5. 학습률 적용 폭 적용(예시는 5 에폭 마다 0.1씩 감소하는 학습률 적용)</li> </ol>	<ol style="list-style-type: none"> <li>1. resnet18(pretrained)</li> <li>2. model.fc.in_features nn.Linear(num, 3)</li> <li>3. nn.CrossEntropyLoss</li> <li>4. optim.SGD(parameters(), lr=0.001, momentum=0.9)</li> <li>5. StepLR(optimizer, step_size=5, epoch=20)</li> </ol>
학습 및 검증 모듈 호출 구현	<ol style="list-style-type: none"> <li>1. 학습모듈을 호출한다.</li> <li>2. 학습 결과 이미지를 출력해 본다.</li> </ol>	

#### 수행 tip

- 절차를 이해하고, 절차에 따라 구현된 코드를 이해한다.
- 모델 연동 및 전이 학습의 목표를 명확하게 이해하고 코드를 구현한다.

## 3-3. 모델 연동 문제 해결

### 학습 목표

- 인공지능서비스 모델 연동 기능의 정상 작동 여부를 확인하고 문제 발생 시 문제를 해결할 수 있다.

### 필요 지식 /

#### ① 인공지능서비스 모델 연동 문제 해결

인공지능서비스 모델 연동과 관련된 문제는 대부분은 모델 연동 방법의 이해 부족 또는 코딩 오류로 인해 발생하는 문제이다. 오류의 종류와 문제를 해결하기 위해 일반적으로 사용하는 디버깅 방법에 대해 학습한다.

##### 1. 오류의 종류

###### (1) 구문 오류(컴파일 오류)

소스 프로그램 작성 중에 발생하는 오류로 문법(Syntax) 오류를 의미한다. 모델 연동 방법을 충분히 숙지하고 있다면 쉽게 발생하지 않는 오류이나, 반대로 이해가 부족할 때에는 해결하기 쉽지 않은 형태의 오류이다. 모델을 제공하는 인공지능 프레임워크에 대해 충분한 숙지가 중요하다.

###### (2) 런 타임 오류

구문 오류는 발생하지 않아 프로그램 실행은 가능하지만 프로그램의 실행 중에 더 진행되지 않거나 중단되는 경우의 오류를 문법 오류와 구분하여 실행 오류라고 한다. 런 타임 오류의 경우는 개발 도구에서 구체적인 파악이 가능하다.

###### (3) 논리 오류

실행 프로그램이 의도한 대로 수행되지 않을 때 발생하는 오류로 실행 프로그램을 검사하고 결과를 분석해야만 실행 프로그램이 정확하게 수행되는지 확인이 가능하다. 논리 오류는 오류 발생 원인을 파악하는 데 많은 시간이 필요하기 때문에 해결하기 어려운 형태의 오류이다.

##### 2. 디버깅

디버깅(debugging)은 프로그램 작성과 실행 과정에서 발생된 오류를 수정해 나가는 일련의 과정이다. 디버깅에 컴퓨터의 도움을 받는 가장 일반적인 방법은 프로그램 중간에 실행 상태를 출력하도록 하는 것이다. 이렇게 함으로써 변수에 잘못된 값이 포함되는지 여부를 프로그래머가 직접 계산하는 것보다 좀 더 쉽게 파악할 수 있다.

디버깅 과정에 컴퓨터의 도움을 받는 가장 일반적인 방법은 디버거(debugger)와 같은

SW를 사용하는 것이다. 디버거는 주로 원하는 코드에 중단점(break point)을 지정하고 프로그램의 실행 정지와 메모리의 저장값을 살펴보며 코드를 단계적으로 실행하거나 실행을 재개하는 등의 동작을 한다. 고가의 디버거 소프트웨어들은 메모리 누수 감지, 메모리 충돌 감지, 다중 스레드 관리 등의 기능도 지원한다.

## 수행 내용 / 인공지능서비스 모델 연동 문제 해결하기

---

### 재료·자료

- 요건 분석서 / 비즈니스 로직 설계서 / 아키텍처 정의서
- 인공지능서비스 인터페이스 연동 문제 해결을 위한 개발 도구(파이참 커뮤니티 버전)
- 운영 체제 - 윈도우 10 환경

### 기기(장비·공구)

- 컴퓨터
- 프로그램 개발 도구, 디버깅 도구
- 문서 열람 도구(ODF 등)

### 안전·유의 사항

- 모델에서 제공하는 각종 모듈 관련 매뉴얼을 면밀하게 분석하고 이해해야 한다.

### 수행 순서

#### ① 인공지능서비스 모델 연동 코드에 대해 디버깅을 수행한다.

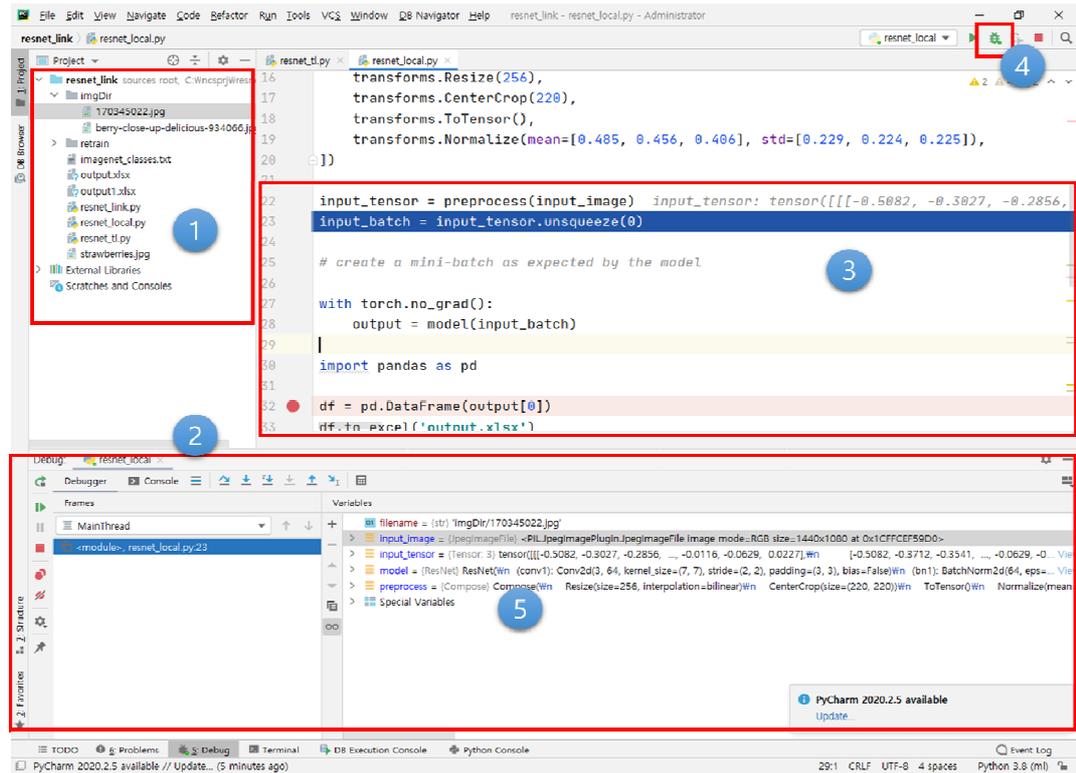
인공지능서비스 모델 연동에서 파이토치의 resnet18 모델을 연동하는 예제 코드를 활용하여 이미지를 분류하는 예제를 수행했다. 해당 코드를 디버깅 예제로 사용하고 다양한 오류를 디버깅을 통해 해결하는 예제를 수행해 본다.

#### 1. 디버깅을 수행한다.

디버깅은 인공지능서비스 모델 연동 예시 코드를 디버깅 대상으로 가정하고 다음과 같이 파이선 프로그램을 디버깅 대상 사례로 예시한다.

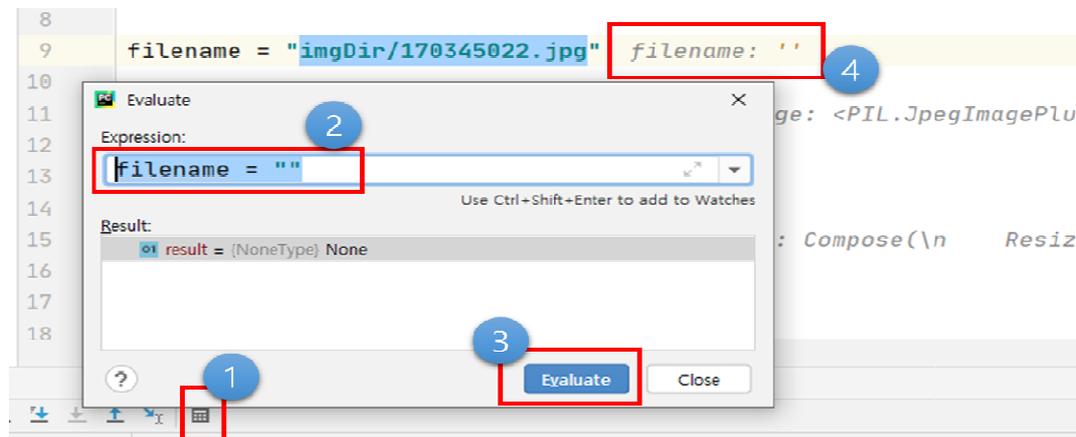
예제 파이선 파일(①)과 파이선 코딩의 디버깅이 필요한 곳에 중단점 표시(③)를 한

다. 이후 ④의 디버깅버튼(Shift+F9)을 클릭하면 ②와 같이 중단점에서 코딩 실행이 멈추고 F8(Step Over)을 클릭하면 라인 단위로 코드를 실행하면서 관찰을 할 수 있는 디버깅 모드가 된다.



출처: 집필진 제작(2021)  
[그림 3-6] 파이참(Community 버전) 개발 도구에서 소스 코드 디버깅 - 1

멈추게 된 상황에서 다양한 실행 옵션을 수행하면 ⑤의 변수 목록에 변수값들의 상황을 관찰할 수 있게 된다.



출처: 집필진 제작(2021)  
[그림 3-7] 파이참(Community 버전) 개발 도구에서 소스 코드 디버깅(변수값 변경 예시) - 2

이렇듯 실행된 코드에서 변수값들을 살펴볼 수 있는 것 외에도 [그림 3-7]과 같이 Evaluate Expression(①)을 클릭하면 변수값을 변경하는 것도 가능하다. 직접 해당 filename의 값을 변경해 보고 실행해 본다.

프로그램을 개발하는 과정에서 다양한 결함을 발견하게 된다. 해당 결함에 대해 어떤 경우는 로그(log)를 남겨야 하는 경우도 있고, 직접 실행하여 변수에 어떤 값이 할당되었는지 확인되어야 하는 경우도 있다.

## 2. 테스트 커버리지를 확인한다.

복잡한 로직에는 다양한 실행 경로가 있어 모든 케이스를 테스트하는 것은 어려운 일이다. 로직의 테스트 커버리지가 어떻게 되는지 확인을 통해 보다 효율적인 테스트 접근이 가능할 수 있다. 파이선에서 지원하는 테스트 커버리지 점검 도구를 활용하여 구현된 프로그램의 테스트 커버리지를 확인한다.

### 테스트 커버리지 확인 명령어

커버리지 점검 도구가 설치되지 않았다면 pip 명령어를 통해 커버리지 도구를 설치한다.

```
(v_aincs_py38) C:\ncsprj\resnet_link> pip install coverage
```

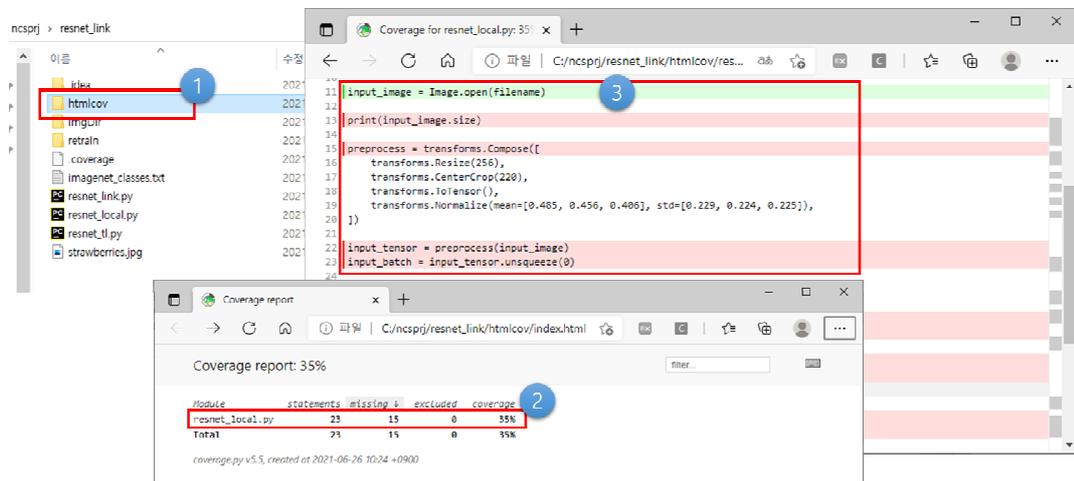
테스트 커버리지를 확인하는 명령어를 수행한다.

```
(v_aincs_py38) C:\ncsprj\resnet_link> coverage run resnet_local.py test
```

테스트 커버리지를 html로 확인하기 위한 명령어를 수행한다.

```
(v_aincs_py38) C:\ncsprj\resnet_link> coverage html
```

위의 명령어를 실행한다. 프로젝트 디렉터리에 htmlcov(①) 디렉터리가 생성되고 해당 디렉터리에 다양한 파일들이 생성되어 있는 것을 확인한다. index.html파일을 클릭하여 프로그램의 커버리지를 확인(②, ③)한다.



출처: 집필진 제작(2021)

[그림 3-8] 테스트 커버리지 확인 결과

### 수행 tip

---

- 유사한 문제를 커뮤니티 등을 활용해 문제를 해결하고 싶어 하는 개발자가 많기 때문에 가장 빠른 문제 해결의 방법은 인터넷을 활용하는 일이다.
- 오류 명세에 포함하지 않는 오류가 포함된 경우는 메시지, 오류 위치 등 파악을 위해 디버깅을 수행해 본다.

### 교수 방법

- 인공지능서비스 모델 연동을 위해 인공지능 모델을 구분해 설명하고, 모델 연동 방법이 모델을 포함하는 인공지능 개발 프레임워크(케라스, 파이토치 등)마다 다르다는 것을 사전에 파악하도록 하고 연동 방법에 대해 수업을 진행한다.
- 인공지능서비스 모델 연동을 위한 개발 도구를 우선 선정하여 수업을 진행한다.
- 설계 내용을 파악하고 이를 개발 언어를 활용하여 작성할 수 있는 절차를 구체적으로 제시한 후 실습하기를 진행한다.
- 인공지능 모델 연동 사례로 파이토치 기반 프레임워크를 활용하므로 파이토치의 모델 연동을 위한 로직 흐름을 설명한 후 수행하기를 진행한다.
- 인공지능서비스 모델 연동 구현에 대해 교재에서 예시한 resnet외에 다른 모델의 구현 실습이 이루어지도록 지도한다.
- 구현된 로직에 대해 디버깅 방법을 실습하고 실제 구현 코드에 대해 디버깅, 테스트 커버리지 현황 등 테스트 활동을 실습하도록 지도한다.

### 학습 방법

- 인공지능 모델에 대해 파악하고, 특히 교재의 파이토치 기반 인공지능 모델 연동 절차를 충분히 숙지한 후 모델 연동 구현을 실습한다.
- 인공지능서비스 모델 연동을 위해 인공지능 개발 프레임워크의 사용 방법을 학습한다.
- 설계 내용을 파악하고 이를 개발 언어를 활용하여 작성할 수 있도록 학습하고 실제 소스 코드를 구현한다.
- 수행하기의 인공지능 모델 연동 소스 코드 파악을 기반으로 실습한다.
- 구현된 로직을 테스트할 수 있도록 디버깅 환경을 구성하고 실제 구현 코드에 대해 모델 연동 부분에 대해 디버깅 등 문제를 해결한다.

## 학습 3      평 가

### 평가 준거

- 평가자는 학습자가 학습 목표를 성공적으로 달성하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습 내용	학습 목표	성취수준		
		상	중	하
모델 연동 환경 구성	- 인공지능서비스 설계에 따라 인공지능서비스 모델의 연동 환경을 구성할 수 있다.			
모델 연동	- 인공지능서비스 모델이 제공하는 다양한 형상화 기술에 따라 애플리케이션 서비스에 인공지능서비스 모델을 연동할 수 있다.			
모델 연동 문제 해결	- 인공지능서비스 모델 연동 기능의 정상 작동 여부를 확인하고 문제 발생 시 문제를 해결할 수 있다.			

### 평가 방법

- 포트폴리오

학습 내용	평가 항목	성취수준		
		상	중	하
모델 연동 환경 구성	- 인공지능서비스 모델 연동 환경을 구성하는 능력			
모델 연동	- 개발 도구를 이용하여 모델에서 제시하는 구현 사례를 실행할 수 있는 능력			
	- 개발 도구를 활용하여 인공지능서비스 모델 연동을 통해 구현 결과물을 표출하는 능력			
모델 연동 문제 해결	- 인공지능서비스 모델 연동 문제 해결을 위해 개발 도구에 디버깅 환경을 구성하는 능력			
	- 모델 연동 관련 구현 코드를 디버깅할 수 있는 능력			

• 문제 해결 시나리오

학습 내용	평가 항목	성취수준		
		상	중	하
모델 연동 환경 구성	- 인공지능서비스 모델 연동과 관련하여 교재에서 제시한 모델 외에 타 모델 연동을 위한 환경 구성 능력			
	- 타 모델 연동 환경 구성 사항들을 팀원들에게 공유할 수 있는 능력			
모델 연동	- 교재에서 제시한 모델 연동 외에 다른 타 인공지능 모델을 연동 구현하고 실행할 수 있는 능력			
	- 인공지능서비스 모델 연동과 관련 제시한 모델에서 타 인공지능 모델을 팀별 프로젝트로 수행하여 실행 결과를 제시할 수 있는 능력			
모델 연동 문제 해결	- 타 모델의 연동 환경을 구성하고 발생된 문제를 해결하여 팀원들과 공유할 수 있는 능력			
	- 타 모델의 인공지능 모델을 활용하여 구현하고 발생된 문제를 해결할 수 있는 능력			

• 사례 연구

학습 내용	평가 항목	성취수준		
		상	중	하
모델 연동 환경 구성	- 예시한 인공지능 개발 프레임워크(파이토치) 외에 타 인공지능 개발 프레임워크(케라스 등)를 조사하고 연동 환경을 구성하는 방법을 제시할 수 있는 능력			
	- 타 인공지능 개발 프레임워크의 연동 환경과 교재의 연동 환경 간 차이점을 설명할 수 있는 능력			
모델 연동	- 예시한 인공지능 모델 외에 타 인공지능 모델을 연동하여 구현하고 실행할 수 있는 능력			
	- 예시한 인공지능 개발 프레임워크(파이토치) 외에 타 인공지능 개발 프레임워크(케라스 등)의 인공지능 모델을 연동하여 구현하고 실행할 수 있는 능력			
모델 연동 문제 해결	- 디버깅을 통한 문제 해결 사례 조사 및 연구 능력			
	- 타 인공지능 개발 프레임워크의 모델 연동 소스 코드에 대한 테스트 커버리지를 검토하고 테스트 케이스를 추가할 수 있는 능력			

• 서술형 시험

학습 내용	평가 항목	성취수준		
		상	중	하
모델 연동 환경 구성	- 인공지능서비스 모델 연동을 위해 인공지능 개발 프레임워크(파이토치 등)에서 제시하는 연동 관련 절차를 이해하는 능력			
모델 연동	- 인공지능서비스 모델 연동을 위해 프레임워크(파이토치 등)에서 제시하는 연동 관련 명세를 이해하는 능력			
	- 인공지능 모델 연동의 필요성과 전이 학습의 목적을 이해하는 능력			
모델 연동 문제 해결	- 인공지능서비스 모델 연동 문제 해결을 위해 개발 도구의 디버거 기능들을 활용할 수 있는 능력			

1. 포트폴리오

- 실습 과정을 포트폴리오 결과에 따라 평가한 후에 개선해야 할 사항 등을 정리하여 학습자에게 설명한다.
- 케라스 등 타 개발 프레임워크의 인공지능 모델 연동 및 전이 학습 기능 구현 로직이 정상적으로 실행되는지 확인하고, 학습자에게 설명한다.
- 평가 결과가 미흡할 때에는 모델 연동 구현 절차 및 방법, 작동 원리에 대해 구체적으로 설명해 준다.
- 성취수준이 낮은 학습자들에게는 실제 구현 코드 라인 단위로 구체적인 설명을 해 준다.
- 성취수준이 높은 학습자의 경우 좀 더 복잡한 인공지능 모델 연동에 대하여 심화 설명해 준다.

2. 문제 해결 시나리오

- 케라스 등 타 개발 프레임워크의 인공지능 모델 연동 및 전이 학습 구현 내용에 대해 제출한 결과를 평가한 후 주요 사항에 대하여 의견을 제시한다.
- 구현된 모델 연동 기능이 정상적으로 동작하는지를 보고 비정상 동작의 원인과 조치 방안에 대해 설명한다.
- 인공지능서비스 모델 연동 로직 구현 결과가 '상'인 경우 모범 사례로 학습자들에게 공유한다.
- 인공지능서비스 모델 연동 로직 구현 결과가 '중'인 경우 부족한 부분을 상세히 설명하고 보완할 수 있도록 한다.
- 인공지능서비스 모델 연동 로직 구현 결과가 '하'인 경우 모범 사례를 참조하여 다시 구현할 수 있도록 유도한다.

3. 사례 연구

- 제시한 인공지능 모델 외에 타 모델을 조사하고 연동 방법을 조사 및 제출하게 하여 평가 및 토론하게 하고 미흡한 사례 조사 내용에 대해 구체적으로 설명해 준다.
- 교재에서 제시한 ResNet18 인공지능 모델에 대해 성취수준이 높은 학습자들에게는 타 모델 연동 방법에 대해 설명해 준다.
- 타 인공지능 모델 연동 구현 시 차이점을 팀원들에게 설명할 수 있는 시간을 갖게 하고 성취수준이 낮은 학습자들에게는 모델 연동 방법에 대해 설명해 준다.
- 문제 해결 방법으로 교재에서 제시한 디버깅 및 응답 오류 코드에 대해 조사 및 분석하고 정리하여 팀원들과 공유하도록 하고, 성취수준이 낮은 학습자들에게는 응답 오류 코드 조사 및 분석 결과에 대해 설명해 준다.

4. 서술형 시험

- 예시한 인공지능 개발 프레임워크(파이토치) 외에 타 인공지능 개발 프레임워크(케라스 등)를 조사하고 연동 환경의 차이를 서술하도록 하고 결과에 대해 공유한다.
- 타 인공지능 개발 프레임워크의 연동 환경과 교재의 연동 환경 간 차이점을 서술하도록 하고 결과에 대해 공유한다.
- 인공지능 모델 연동의 필요성과 전이 학습의 목적을 서술형으로 평가하고 결과를 공유한다.
- 인공지능서비스 모델 연동 문제 해결을 위해 개발 도구의 디버거 기능들을 서술하도록 요구하고 결과를 평가한다.
- 결과에 대해 '상'인 경우 모범 사례로, '중'인 경우 부족한 사항을 설명하여 보완하도록 하고, '하'인 경우는 재학습의 기회를 제공하여 숙지하도록 한다.

학습 1	인공지능서비스 비즈니스 로직 구현하기
학습 2	인공지능서비스 플랫폼 연동하기
학습 3	인공지능서비스 모델 연동하기
<b>학습 4</b>	<b>인공지능서비스 인터페이스 연동하기</b>

## 4-1. 인터페이스 연동 환경 구성

**학습 목표** • 인공지능서비스 설계에 따라 인공지능서비스 인터페이스 연동 환경을 구성할 수 있다.

### 필요 지식 /

#### ① 인공지능서비스 인터페이스 연동 환경 구성

인공지능서비스 인터페이스 연동은 인공지능서비스 구현을 위해 외부의 ‘인공지능 모델’을 연동 방식으로 활용한다는 것을 의미한다.

##### 1. 인공지능서비스 인터페이스 연동을 위한 구성

인공지능서비스 인터페이스 연동 구현은 인공지능 모델 서버를 통해 인공지능서비스를 실현하는 과정이다. 대부분의 상용화된 인공지능 모델 서비스의 인터페이스는 HTTP를 통해 REST API를 제공한다.

#### 오픈(공개) API, Restful API(Application Programming Interface)

(오픈 API) 개발자라면 누구나 사용 가능하도록 공개된 API를 의미하며, 개발자에게 프로그래밍을 위한 공개된 명세를 제공한다. 대표적으로 네이버 지도, 구글맵, 오픈스트리트맵 등이 있다.  
(Restful API) REST(Representational State Transfer)는 자원(RESOURCE, URI), 행위(Verb, HTTP METHOD), 표현(Representations)으로 구성되어 있다. 특히 Uniform Interface, Stateless, Cacheable, Self-descriptiveness, Client/Server 구조, 계층형 구조의 특징을 갖는다.

HTTP(Hyper Text Transfer Protocol)를 통한 REST 사용이 일반적이지만, RPC(Remote Procedure Call)를 사용하는 방법도 있다. 따라서 API를 사용하기 위해 서버에서 제공하는 사양을 이해할 수 있어야 한다.

## 2. REST

자원을 이름으로 구분하여 해당 자원의 상태를 주고받는 것이 REST이다. 일반적으로 협의의 REST는 HTTP를 통해 CRUD(Create, Read, Update, Delete)를 실행하는 API이다. REST는 HTTP 프로토콜을 사용하기 때문에 통합 자원 식별자(Uniform Resource Identifier, URI)로 서버 자원을 특정하고 HTTP Methods를 통해 수행할 트랜잭션을 지정하게 된다. HTTP를 통해 주고받는 데이터 형식은 JSON 또는 XML을 사용하는 것이 일반적이다.

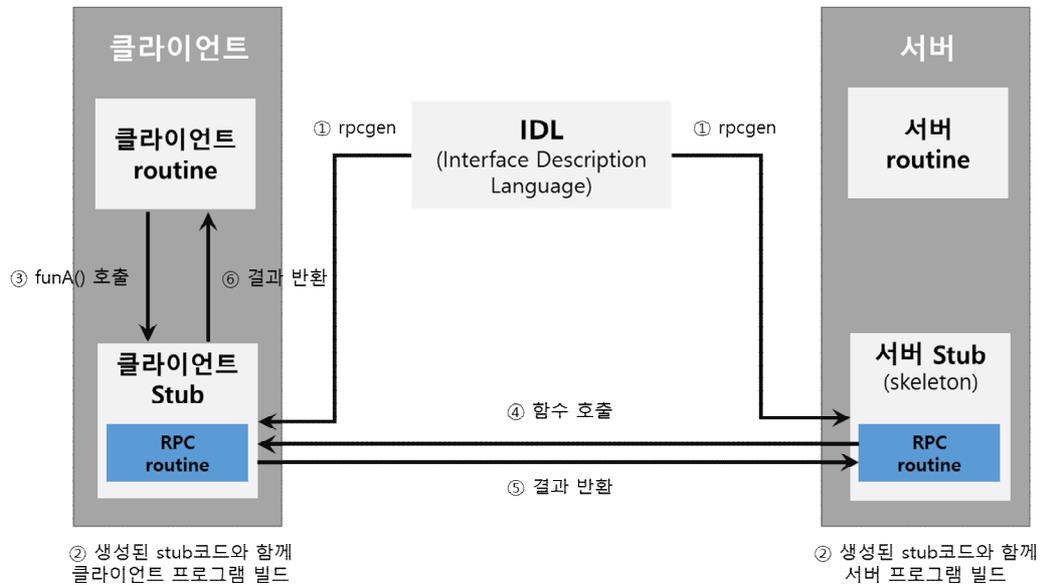
〈표 4-1〉 REST 특징

주요 특징	설명
'클라이언트-서버' 구조	자원을 요청하는 쪽이 클라이언트, 자원을 보유한 쪽이 서버
무상태(Stateless)	'서버'는 각각의 요청을 완전한 독립적 요청으로 인식하고 처리 이전 요청이 다음 요청의 처리에 독립성을 부여(이전 요청과 연관 없어야 함.)
캐시 처리 가능(Cacheable)	효율적인 대량의 요청을 처리하기 위해 캐시 정보를 저장 및 처리할 수 있게 함.
계층화(Layered System)	클라이언트가 대상 서버 또는 중간 서버를 경유하는지 여부 확인 불가 따라서 중간 서버의 로드 밸런싱 및 공유 캐시 기능을 제공함으로써 시스템 규모 확장 가능
Code on demand(optional)	JAVA Applet이나 JAVA Script를 이용하여, 클라이언트가 실행할 수 있는 로직을 서버를 통해 전송하여 확장하는 것 가능
인터페이스 일관성	아키텍처를 단순화시키고 작은 단위로 분리 클라이언트 및 서버의 각 파트가 독립적으로 분산될 수 있도록 함.

## 3. RPC

원격에서 상대 프로세스에 포함된 함수를 호출 가능하게 하는 프로세스 간 통신 기술로 상호 당사자들이 미리 정의된 규격을 준용하여 구현하는 것이 중요한 기술이다. 통상 함수 호출이란 프로세스 자신의 주소 공간 내에 있는 함수를 호출하여 기능을 실행하는 것을 의미하며, RPC 기술을 통해서서는 다른 주소에서 동작하는 프로세스의 특정 함수를 실행할 수 있는 기술이다.

RPC의 궁극적인 목표는 클라이언트/서버 간 커뮤니케이션에 필요한 상세 정보를 최대한 노출하지 않고(Information Hiding), 클라이언트 측면에서는 일반 함수를 호출하듯이 호출할 수 있도록 간결화한다. 대표적으로 Google의 protocolbuffer, Facebook의 Thrift, Twitter의 Finalge 등을 예로 들 수 있다.



출처: 집필진 제작(2021)  
 [그림 4-1] RPC 원격 프로시저 호출 절차

분산 컴퓨팅 환경에서 프로세스 간 통신 및 컴퓨팅 자원의 효율적 사용을 위해 발전된 기술로 서버/클라이언트, IDL(Interface Definition Language), Stub, XDR(eXternal Data Representation) 등의 요소 기술로 구성된다.

<표 4-2> RPC 구성 및 설명

구성	설명
IDL	서버의 호출 규약을 정의 함수명, 파라미터, 반환값의 데이터형이 정의된 IDL을 rpcgen 컴파일러를 통해 stub 코드로 자동 생성
stub	- 원시 소스 코드(C 코드 등) 형태로 생성되므로 클라이언트/서버 프로그램에 포함한 후 빌드 - 데이터형을 XDR 형식으로 변환하여 RPC 호출을 실행 ※ XDR 변환: 데이터 타입(정수형, 부동소수점 등)에 대한 메모리 저장 방식(리틀엔디안, 빅엔디안)이 CPU 아키텍처에 따라 다르며, 네트워크 전송 과정 중에 바이트 순서를 보장하기 위함.
rpcgen	RPC 프로토콜 컴파일러 RPC 프로토콜을 실현하기 위한 원시 코드 생성 도구
클라이언트 프로그램	자신의 프로세스 주소 공간의 함수를 호출하듯이 stub에 정의된 함수를 호출
서버 프로그램	수신된 함수 및 프로시저 호출의 처리 완료 후, 결과값은 XDR로 변환 후 반환

## 수행 내용 / 인공지능서비스 인터페이스 연동 환경 구성하기

### 재료·자료

- 요건 분석서(요구 사항 정의서, 요구 사항 명세서) / 아키텍처 정의서
- 인공지능서비스 인터페이스 연동을 위한 개발 도구  
(인터넷 가능한 컴퓨터 환경, 설치 파일을 다운로드받아야 하는 경우 대비)
- 운영 체제 - 윈도 10 환경

### 기기(장비·공구)

- 문서 열람 도구(ODF 등)
- 인터넷 브라우저(Chrome 등)

### 안전·유의 사항

- 비즈니스 로직 구현을 위한 인터페이스 연동 환경을 이해할 수 있어야 한다.
- 설계자 또는 이해관계자들과 원활한 의사소통을 할 수 있는 역량을 갖춘다.
- 인공지능서비스 인터페이스 연동을 위한 요구 환경을 갖출 수 있어야 한다.

### 수행 순서

#### ① 인공지능서비스 인터페이스 연동 환경 구성을 위해 정의서(설계서)를 확인한다.

인공지능서비스 인터페이스 연동 환경 구성을 위해 요건 분석서, 아키텍처 정의서의 내용을 확인한다.

##### 1. 요건 분석서를 확인한다.

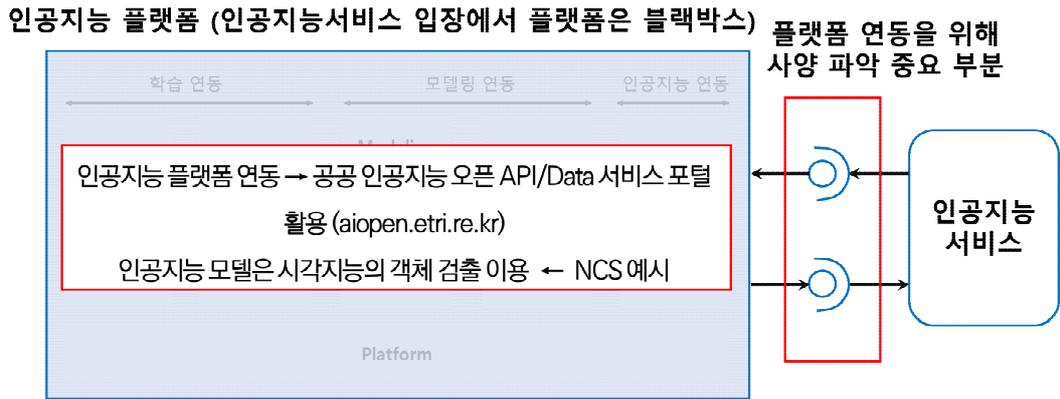
인공지능서비스를 제공하는 다양한 인터페이스 중에서 어떤 인터페이스를 이용해야 하는지 요건 정의서를 통해 확인한다.

〈표 4-3〉 요건 분석서 예시

대분류	중분류	소분류	요구 사항명	요구 사항 상세	근거
...	...	...	...	...	...
시스템 장비 구성 요구 사항	인터페 이스	인공지능 모델 인터페이스	인터페이스 연동 환경	인터페이스 연동은 공공 인공지능 오픈 API, Data 서비스 포털을 활용한다. (aiopen.etri.re.kr)	RFP
...	...	...	...	...	...

## 2. 아키텍처 정의서를 확인한다.

인공지능서비스 인터페이스 연동을 구현하기 위해 구성되어야 할 환경을 아키텍처 정의서를 통해 파악하고 상세 구성 내용을 해당 플랫폼에서 제공하는 상세 명세서를 통해 구성한다.



출처: 집필진 제작(2021)

[그림 4-2] 인공지능서비스 플랫폼 연동 개념도(아키텍처 정의서 일부 예시)

### ② 인공지능서비스 인터페이스 연동 환경을 구성한다.

요건 정의서, 아키텍처 정의서에 정리된 순서대로 수행하여 인터페이스 연동 환경을 구성한다. 인터페이스를 제공하는 환경마다 구성 순서가 다를 수 있으므로 환경에서 요구하는 준비 사항을 명확하게 이해할 수 있어야 한다. <표 4-4>는 예시를 통해 제시한 공공 인공지능 오픈 API를 사례로 수행해 본다.

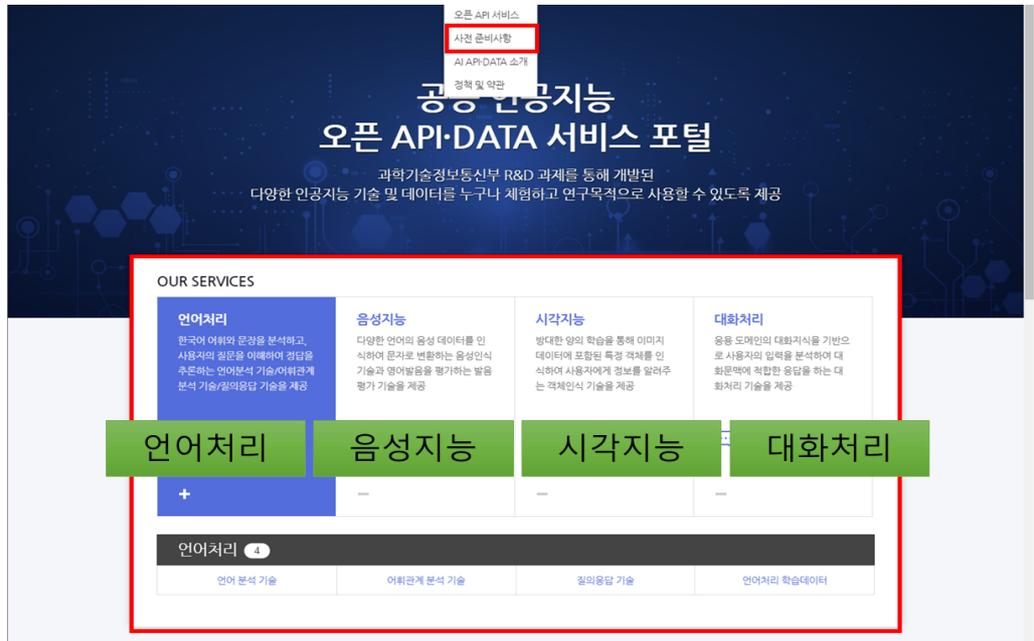
<표 4-4> 인공지능서비스 인터페이스 연동 환경 구성을 위한 순서

구성 순서	설명
공공 인공지능 인터페이스 이용 방법 파악	- 인공지능서비스 인터페이스 연동을 위한 안내 페이지를 통해 준비 사항 확인
공공 인공지능 인터페이스 이용을 위한 키 발급 신청	- 개인 정보 동의 및 인증 - Key 발급 신청
키 발급 신청 확인 및 키 관리	- 키 발급 확인 및 관리
인공지능 모델 확인	- 해당 인터페이스에서 제공하는 인공지능 모델 확인

## 1. 공공 인공지능 API 서비스의 이용 방법을 파악한다.

### (1) 공공 인공지능 API 서비스를 접속한다.

예시에서 제시한 공공 인공지능 오픈 API·DATA 서비스 포털을 접속한다. 해당 포털에 접속하면 언어 처리, 음성 지능, 시각 지능, 대화 처리와 관련된 인공지능을 확인할 수 있다(<https://aiopen.etri.re.kr/>).

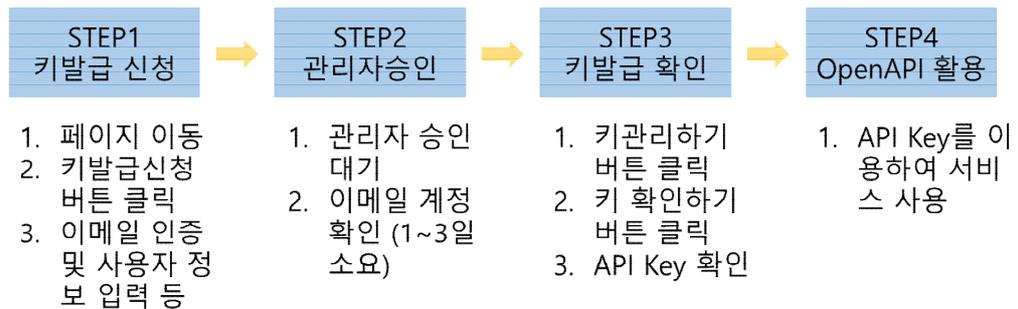


출처: 집필진 제작(2021)

[그림 4-3] 공공 인공지능 오픈 API·DATA 서비스 포털 접속 화면

(2) 공공 인공지능 API 서비스의 사전 준비 사항을 확인한다.

Step01~Step04를 확인한다. 환경 구성을 위해서는 Step01과 Step03을 수행해야 한다.

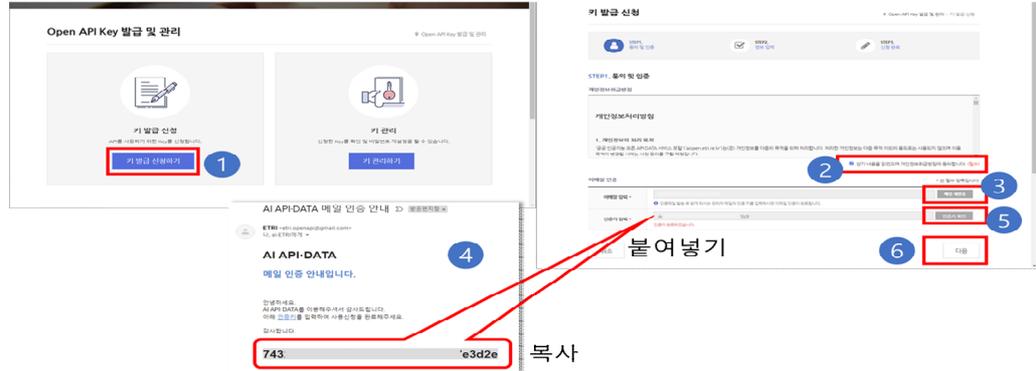


출처: 집필진 제작(2021)

[그림 4-4] 공공 인공지능 오픈 API 서비스 이용 절차

(3) 공공 인공지능 API 서비스 사용을 위해 Key 발급 신청을 한다.

Open API Key 발급을 위해 다음 순서에 따라 수행한다. “Open API Key 발급 및 관리(aiopen.etri.re.kr/key\_main.php)” 화면으로 이동한다. ① “키 발급 신청하기”를 클릭하면 “키 발급 신청(aiopen.etri.re.kr/key\_creation.php)” 화면이 나타나며, ② 동의, ③ 이메일 주소 입력 및 인증 메일 받기, ④ 메일 확인 및 인증 키 확인, ⑤ 인증 키 입력, ⑥ 인증 키 확인을 차례대로 수행하고 화면 하단의 “다음” 버튼을 클릭하면 “STEP2. 정보 입력” 화면으로 이동하게 된다.



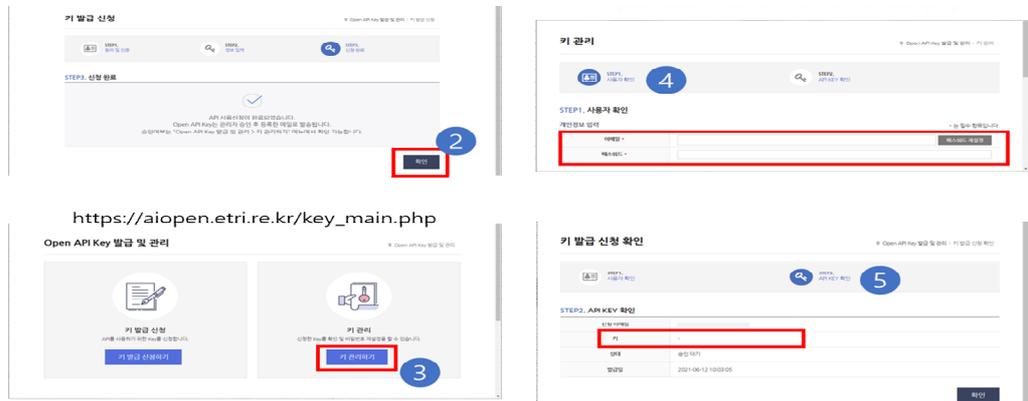
출처: 집필진 제작(2021)  
[그림 4-5] 공공 인공지능 API 서비스 이용 환경 구성 - 1

“STEP2. 정보 입력” 화면은 키 발급 신청을 위한 사용자의 기본 정보를 입력하는 단계이다. [그림 4-6]의 사용자 기본 정보, 사용자 부가 정보, 개인 정보 수집 등 동의 입력을 등록한 후 하단의 ①신청하기 버튼을 클릭한다.



출처: 집필진 제작(2021)  
[그림 4-6] 공공 인공지능 API 서비스 이용 환경 구성 - 2

신청하기 버튼을 클릭하면 ②의 화면으로 이동하며 관리자 승인 후 사용이 가능하다는 안내 메시지를 확인한다. API Key 발급 여부에 대해서는 “키 관리 (aiopen.etri.re.kr/key\_check.php)” 화면에서 확인한다(③, ④, ⑤ 참조).



출처: 집필진 제작(2021)  
[그림 4-7] 공공 인공지능 API 서비스 이용 환경 구성 - 3

여기까지가 인공지능서비스 인터페이스 연동을 위한 환경 구성이다. 지금까지 수행해 온 절차들을 통해 공공 인공지능 API 서비스 연동을 위한 환경 구성은 완료할 수 있다.

## 2. Open API 목록을 확인한다.

공공 인공지능 API 서비스에서 제공하는 인공지능 분야는 “언어 처리”, “음성 지능”, “시각 지능”, “대화 처리”이다. 인공지능 분야에서도 인공지능서비스로 제공되는 Open API는 다음과 같이 다양하게 제공되고 있지만, 요건 정의서, 설계서를 바탕으로 하여 활용해야 하는 Open API를 선정하고 해당 명세서를 살펴볼 수 있어야 한다.

〈표 4-5〉 공공 인공지능 오픈 API에서 제공하는 서비스 목록

인공지능 분야	인공지능서비스	Open API 유형	이용 조건
언어 처리	언어 분석 기술(문어)	형태소 분석 API	5,000건/일 (1회 입력은 1만 글자 이하)
		개체명 인식 API	
		다의어 분석 API	
		동음이의어 분석 API	
		의미역 인식 API	
		의존 구문 분석 API	
	언어 분석 기술(구어)	개체명 인식 API	5,000건/일
		형태소 분석 API	(1회 입력은 1만 글자 이하)
	어휘 관계 분석 기술	어휘 정보 API	5,000건/일
		다의어 정보 API	
		동음이의어 정보 API	
	질의응답 기술	어휘 간 유사도 분석 API	5,000건/일 (1회 입력은 1만 글자 이하)
질문 분석 API			
위키백과 QA API			
기계 독해 API			
음성 지능	음성 인식 기술	법률 QA API	1,000건/일 (60초 이내/건당)
		한국어 인식 API	
		영어 인식 API	
		아랍어 인식 API	
		중국어 인식 API	
		독어 인식 API	
		베트남어 인식 API	
		스페인어 인식 API	
		불어 인식 API	
		러시아어 인식 API	
	일본어 인식 API		
발음 평가 기술	한국어 발음 평가 API		
	영어 발음 평가 API		
시각 지능	이미지 인식 기술	객체 검출 API	250건/일
		얼굴 비식별화 API	

인공지능 분야	인공지능서비스	Open API 유형	이용 조건
		사람 속성 검출 API	
	동영상 인식 기술	장면 분할 API	백 건/일
대화 처리	대화 처리 기술	대화 처리 API	이천 건/일

출처: ETRI(2021. 08.). 오픈 API 목록. [https://aiopen.etri.re.kr/service\\_list.php](https://aiopen.etri.re.kr/service_list.php)에서 2021. 08. 07. 검색.

#### 수행 tip

- 타 인공지능 인터페이스 환경의 경우 일부 절차가 추가되거나 생략될 수 있으므로 이용 방법을 잘 살펴보아야 한다.
- Open API 키 발급은 일정 기간 이후 완료된다.

## 4-2. 인터페이스 연동

**학습 목표** • 인공지능서비스 인터페이스 가이드에 따라 애플리케이션 서비스에 인터페이스를 연동할 수 있다.

### 필요 지식 /

#### ① 인공지능서비스 인터페이스 연동 요건 및 설계 파악

##### 1. 인공지능서비스 인터페이스 연동을 위한 모듈(Module) 개념의 이해

모듈이란 전체 프로그램의 기능 중에서 특정 기능 또는 공통 기능을 처리할 수 있는 소스 코드를 의미하고 다른 프로그램에서 재사용이 가능하다.

##### 2. 인공지능서비스 인터페이스 연동을 위한 모듈(Module) 명세 이해

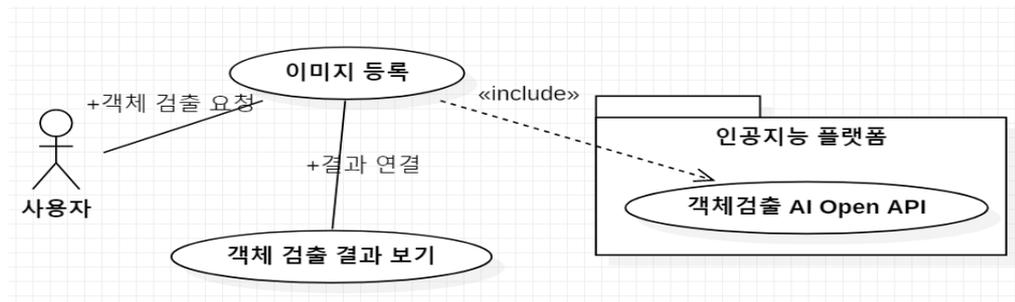
모듈의 명세는 <표 4-6>과 같은 원칙하에 기술되어 있으므로 인공지능서비스 인터페이스 연동을 위해 해당 원칙에 기반해서 명세를 확인한다.

<표 4-6> 인공지능서비스 인터페이스 연동 구현 시 확인해야 하는 모듈 명세 기준

주요 원칙	설명
정확성(Correctness)	해당 기능을 정확하게 작성
명확성(Clarity)	명세 기능은 일관되게 이해되어야 하고 한 가지 의미로 해석되도록 작성
완전성(Completeness)	시스템이 구현될 때 필요하고 요구되는 모든 것을 기술
일관성(Consistency)	공통 기능들 간에 상호 충돌이 없도록 작성
추적성(Traceability)	공통 기능에 대한 요구 사항 출처와 관련 시스템 등의 유기적 관계에 대한 식별이 가능하도록 작성

##### 3. 인공지능서비스 비즈니스 흐름 파악

비즈니스 흐름은 유스케이스 다이어그램, 유스케이스 명세서 등을 통해 파악하고, 인공지능서비스의 역할도 파악한다.

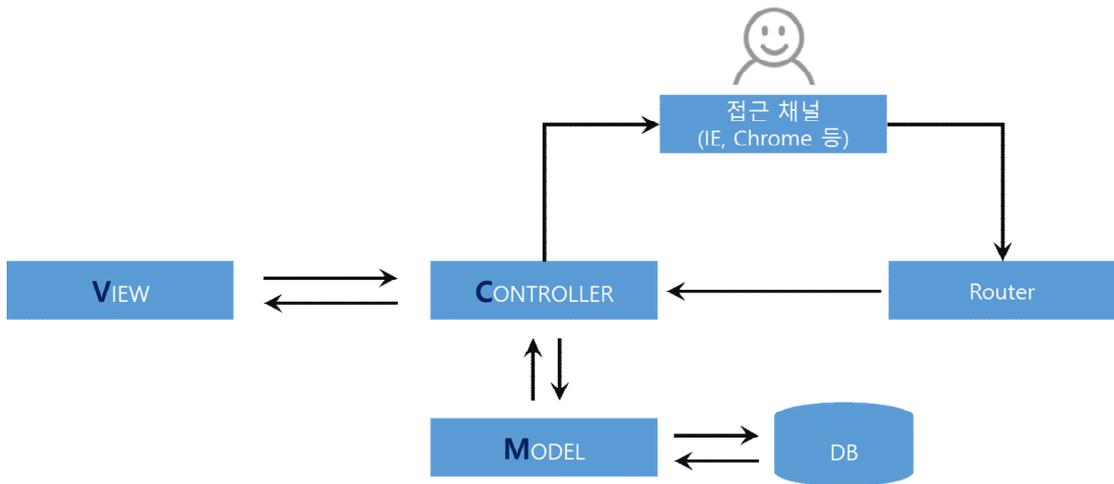


출처: 집필진 제작(2021)  
[그림 4-8] 비즈니스 흐름 파악 예시(유스케이스 다이어그램)

② 인공지능서비스 인터페이스 연동을 위한 MVC 모델 이해

사용자 인터페이스와 비즈니스 로직을 분리함으로써 애플리케이션의 시각적 요소와 이 면에서 실행되는 비즈니스 로직을 서로 영향 없이 유연하게 유지할 수 있도록 지원하는 소프트웨어 디자인 패턴이다.

MVC에서 모델은 서비스의 정보(데이터)를 의미하는 것이며, 컨트롤러는 데이터와 비즈니스 로직의 상호 흐름 등 동작을 관리하고, 뷰는 텍스트, 라디오 버튼 및 체크박스 항목 등과 같은 사용자의 인터페이스(UI) 요소를 나타낸다.



출처: 집필진 제작(2021)  
[그림 4-9] MVC 모델 개념

<표 4-7> MVC 모델의 구성 및 작동 원리

구성	작동 원리
모델	모델의 상태 변화를 컨트롤러와 뷰에 통보 통보 결과에 따라 뷰는 최신의 결과를 보여 주는 것이 가능하고, 컨트롤러는 모델 변화에 따라 적용이 가능한 명령들을 추가, 제거 또는 수정
뷰	MVC에서 모델(Model)은 여러 개의 뷰(view)를 가지는 것 가능 뷰는 모델에 질의(Query)를 하여 모델로부터 가져온 값을 사용자에게 보여 주는 역할
컨트롤러	MVC의 뷰는 다수의 컨트롤러(controller)를 가지는 것 가능 사용자는 컨트롤러를 사용하여 모델의 상태를 변경 모델 상태 변경 시 자신의 상태가 바뀌었다는 것을 등록된 뷰에 알리고 뷰는 변경 상태에 맞게 사용자에게 모델의 상태를 보여 줌

## 수행 내용 / 인공지능서비스 인터페이스 연동하기

### 재료·자료

- 요건 분석서 / 인공지능서비스 인터페이스 연동 명세서
- 인공지능서비스 로직 구현을 위한 개발 도구
- 운영 체제 - 윈도 10 환경

### 기기(장비·공구)

- 문서 열람 도구(ODF 등)
- CASE 도구(SW 설계 지원 도구, SW 개발 지원 도구, SW 테스트 지원 도구 등)

### 안전·유의 사항

- 설계자 또는 이해관계자들과 원활하게 의사소통할 수 있는 역량을 갖춘다.
- 인공지능서비스 인터페이스 연동 구현을 위한 명세서(설계서)를 이해할 수 있어야 한다.

### 수행 순서

#### ① 인공지능서비스 인터페이스 연동을 위한 명세서를 확인한다.

인공지능서비스를 위해 인터페이스 연동이 필요한 인공지능 API는 객체 검출로 가정한다. 이에 따라 공공 인공지능 OpenAPI에서 연동이 필요한 객체 검출 API의 연동 명세서를 확인한다.



출처: 집필진 제작(2021)  
[그림 4-10] Open API 연동 대상 명세 확인

1. 인터페이스 연동을 위한 Open API의 개요를 확인한다.

객체 검출 API 연동을 위해서 Open API에서 제공하는 인터페이스 개요를 확인한다. 예제로 제시한 해당 Open API는 전체 30개의 객체를 검출하는 인공지능 모델이며, 각 객체별로 작게는 301개부터 많게는 약 25,000여 개의 이미지를 학습한 인공지능 모델을 연동하는 것임을 파악할 수 있다.

객체검출 학습데이터 (업데이트 : 2020.11.25)

영상 데이터 인식 기술을 개발하기 위한 과학기술정보통신부 소프트웨어 분야 R&D 과제인 딥뷰(DeepView) 과제에서는 영상 데이터 이해 및 예측을 위한 플랫폼을 개발하고 있습니다. 딥뷰 과제를 수행하면서 구축한 약 20만장(기준 약 10만장) 정도의 객체 검출용 이미지 학습데이터를 배포하여, 유사 분야 연구에 도움이 되고자 합니다.

객체 검출용 이미지 학습데이터는 ImageNet 분류 체계 기반으로 약 560여개의 객체(사람, 개, 자동차 등)를 정의하였으며, 객체에 대한 정보(객체 이름)와 영상 내에 그 객체의 위치 좌표 정보가 태깅된 이미지로 구성되어 있습니다.

이러한 표와 같이 15개의 세부 카테고리로 구분된 학습데이터의 다운로드를 제공합니다. 다운로드되는 파일은 JPEG 이미지와 XML 객체 정보 파일입니다.

No	상위 카테고리	세부 카테고리	이미지 수
1	사람	사람	18,646
2		사람 2	10,069
3		얼굴	5,563
4		얼굴 2	301
~~ 생략 ~~			
24	자전거류 2	자전거류 2	6,038
25		도구, 기계, 장비, 가구	14,070
26		도구, 기계, 장비, 가구 2	4,996
27		가구, 가전, 악기, 장난감, 무기	15,180
28	가구, 가전, 악기, 장난감, 무기 2	7,451	
29	자연 구조물	지형, 자연구조물	5,916
30		지형, 자연구조물 2	1,322
총 합계			329,956

출처: 집필진 제작(2021)  
[그림 4-11] 인터페이스 연동 Open API의 객체 검출 개요

인공지능 플랫폼에서 제공하는 객체 검출 모델의 검출 대상 객체들과 비즈니스 구현을 위해 설계자가 설계(설계서)한 검출 대상 객체와 불일치한 사항이 있는지 확인해 본다.

2. 인터페이스 연동 Open API의 명세를 확인한다.

플랫폼에서 제공하는 객체 검출 인공지능은 이미지에서 객체에 해당하는 자동차, 사람 등 다양한 종류의 객체 카테고리를 인식 및 분류하고 객체의 위치 관련 정보(박스 좌표)를 식별한다. 객체 검출 API는 JSON 포맷 기반의 REST API 인터페이스로 입력 및 출력(오류 코드 포함)을 통해 연계한다.

(1) 객체 검출 API의 요청 명세를 확인한다.

객체 검출 API의 요청 파라미터로 access\_key, argument, file, type 네 개가 필요하다. 각 파라미터는 String, Object 타입이며, 네 개의 파라미터 모두 필수로 제출해야 한다.

〈표 4-8〉 객체 검출 API 요청 명세

필드명	타입	필수 여부	필드 설명
access_key	String	○	• Open API 사용을 위해 해당 플랫폼에서 발급한 API Key
argument	Object	○	• API 요청 시 분석을 위해 전달할 내용
file	String	○	• 객체 검출 대상 이미지를 Base64 문자열로 변환 • 512x512 이하 저해상도 이미지는 객체 검출률이 낮아질 수 있음.
type	String	○	• 이미지 파일의 확장자

출처: 공공 인공지능 포털(2020. 11. 25.). 요청 명세. aiopen.etri.re.kr/guide\_od.php에서 2021. 6. 12. 검색.

(2) 객체 검출 API의 응답 명세를 확인한다.

객체 검출 API를 오류 없이 요청하면, 〈표 4-9〉와 같이 데이터 배열에 검출 객체(class), 신뢰도, 객체의 위치 정보(박스 좌표)를 응답한다.

〈표 4-9〉 객체 검출 API 응답 명세

구분	Json Key 이름	설명
객체 검출 기본 정보	data	검출된 객체 배열
	class	인식된 객체명
	confidence	신뢰도
	x	x 좌표
	y	y 좌표
	width	길이
	height	높이

출처: 공공 인공지능 포털(2020. 11. 25.). 응답 명세. aiopen.etri.re.kr/guide\_od.php에서 2021. 6. 12. 검색.

단, 객체 검출이 제대로 이루어지지 않을 경우는 오류 메시지를 반환하며, Json Key의 값은 “error”이다. 오류 메시지의 예시로 “This file is not allowed file type”, “This file is not image contents” 등을 응답한다.

3. 인공지능서비스 인터페이스를 연동(구현)한다.

인터페이스 연동을 위한 예제 코드는 해당 플랫폼 URL을 통해 Java, PHP, C++, Python, Node.js로 제시하고 있다([https://aiopen.etri.re.kr/guide\\_od.php](https://aiopen.etri.re.kr/guide_od.php)). 수행하기를 통한 구현 예시는 파이썬(python) 코드로 수행하고, 예제는 다음 코드와 같다.

### 파이썬(python) 구현 예제([https://aiopen.etri.re.kr/guide\\_od.php](https://aiopen.etri.re.kr/guide_od.php))

```
import urllib3, json, base64

# 함수로 변형
APIURL = "http://aiopen.etri.re.kr:8000/ObjectDetect"
accessKey = "YOUR_ACCESS_KEY" → 변수 대체
imgFilePath = "IMAGE_FILE_PATH" → 변수 대체
type = "IMAGE_FILE_TYPE" → 변수 대체

fl = open(imgFilePath, "rb")
imgContents = base64.b64encode(fl.read()).decode("utf8")
fl.close()

# 요청 파라미터
requestJson = {
    "access_key": accessKey,
    "argument": {
        "type": type,
        "file": imgContents
    }
}

# 응답 파라미터
datahttp = urllib3.PoolManager()
res = datahttp.request(
    "POST",
    APIURL,
    headers={"Content-Type": "application/json; charset=UTF-8"},
    body=json.dumps(requestJson)
)

# 응답 결과 출력
print("[responseCode] " + str(res.status))
print("[responBody]")
print(res.data)
```

공공 인공지능 포털에서 예시한 코드를 함수로 일부 변형하고, 변형된 함수를 호출하여 결과 표출과 객체 검출 내용을 이미지로 보여 주는 연동 코드를 구현한다.

### 객체 검출 결과 표출 예시

```
import urllib3, json, base64
from PIL import Image, ImageDraw, ImageFont
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

def call_interface(accKey, imgPath, imgType): # 공공인공지능의 구현 예시 코드를 함수화
    ... 코드 생략 ...
    accessKey = accKey
    imageFilePath = imgPath
    type = imgType
    ... 코드 생략 ...
    return response.data

def pil_draw_rect(image, x1, y1, x2, y2, tclass): # 객체 검출 이미지 표현 함수
    font = ImageFont.truetype('./font/font_a.ttf',size=20)
    font_color = 'rgb(0, 0, 0)'
    draw = ImageDraw.Draw(image)
    draw.rectangle(((x1,y1), (x2,y2)),outline=(0, 0, 255),width=2)
    draw.text((x1,y1), tclass,(255, 0, 0), font=font)
    return image

res_data = call_interface("137add5-####-####-####-8523f3dc5f7a",imageFilePath, "jpg")
json_data = json.loads(res_data)
imageFilePath = "./imageDir/205253975.jpg"
image = Image.open(imageFilePath)
# 응답 결과로 이미지에 검출 객체를 표시하는 코드
for data in json_data["return_object"]["data"]:
    image = pil_draw_rect(image,int(data["x"]),int(data["y"]),
                          int(data["x"])+int(data["width"]),int(data["y"])+int(data["height"]),
                          data["class"])

df = pd.DataFrame.from_dict(json_data["return_object"]["data"])
print(df)
display(image)
```

위의 인터페이스 연동 코드는 특정 이미지에 대해 객체 검출 요청 시, response.data에 json 형식의 응답 결과를 받는다. <표 4-10>의 사례는 response.data를 객체 검출 영역과 전체 검출 객체를 출력해 본 결과이다. 전체 19개의 객체 검출과, 평균 약 84% 정도의 신뢰도를 확인할 수 있다.

<표 4-10> 객체 검출 결과 목록

검출 수	class	confidence	x	y	width	height
1	person	0.992294	227	368	546	692
2	person	0.977712	932	572	45	116
3	person	0.97558	1007	572	34	112
4	person	0.971795	187	559	42	114
5	person	0.971747	297	567	51	117
6	person	0.959623	272	557	36	125
7	person	0.949823	98	549	35	113
8	person	0.947702	973	605	41	79
9	person	0.912327	226	571	36	101
10	person	0.860002	171	558	31	114
11	person	0.800575	1	563	34	108
12	person	0.795295	646	516	106	203
13	person	0.783402	899	576	33	44
14	person	0.77616	64	458	27	97
15	person	0.703788	744	548	36	51
16	person	0.666111	1038	590	26	93
17	person	0.659189	301	496	32	55
18	person	0.630764	748	474	21	64
19	person	0.613189	29	541	28	54

위 json 결과를 바탕으로 하여 구현 코드 pil\_draw\_rect(image, x1, y1, x2, y2, tclass)를 통해 입력 이미지에 검출 객체를 박스로 표시한 결과이다.

<표 4-11> 객체 검출 결과 목록

요청 이미지	응답 결과로 생성한 객체 검출 결과
	
가. 원본 이미지	나. 인공지능 모델의 객체 검출 결과
<p>검출 개수: 19 검출 객체: 1번: person 2번: person 3번: person 4번: person 5번: person 6번: person 7번: person 8번: person 9번: person 10번: person 11번: person 12번: person 13번: person 14번: person 15번: person 16번: person 17번: person 18번: person 19번: person 검출 정밀도: 1번: 0.992294 2번: 0.977712 3번: 0.975580 4번: 0.971795 5번: 0.971747 6번: 0.959623 7번: 0.949823 8번: 0.947702 9번: 0.912327 10번: 0.860002 11번: 0.800575 12번: 0.795295 13번: 0.783402 14번: 0.776160 15번: 0.703788 16번: 0.666111 17번: 0.659189 18번: 0.630764 19번: 0.613189</p>	

가. 출처: 집필진 제작(2021)

나. 출처: 집필진 제작(2021)

### 수행 tip

- 주피터 노트북 환경을 통해 공공 인공지능에서 예시한 코드를 쉽고 빠르게 구현해 볼 수 있다.
- 주피터 노트북에서 예시 코드 구동을 한 후 모듈화 및 이미지에 검출 객체 표시 코드를 구현해 본다.

## 4-3. 인터페이스 연동 문제 해결

### 학습 목표

- 인공지능서비스 인터페이스 연동 기능의 정상 작동 여부를 확인하고 문제 발생 시 문제를 해결할 수 있다.

### 필요 지식 /

#### ① 인공지능서비스 인터페이스 연동 테스트

인공지능서비스 인터페이스 연동과 관련된 문제는 기초적으로는 프로그램 코딩 역량과 관련한 것이며, 대부분은 인터페이스 연동 방식에 대한 이해 부족 또는 잘못된 코딩으로 인한 문제 발생이다. 따라서 인공지능서비스 인터페이스 연동 방식의 명확한 이해가 중요하다.

##### 1. 인공지능서비스 인터페이스 연동 방식에 따른 문제 해결 방법

다양한 인터페이스 연동 방식에 따라 연동할 때 발생하는 문제 해결을 위해 인터페이스에서 제공하는 문제 해결 방법에 대해 이해하는 것이 중요하다. <표 4-12>는 다양한 연계 방식별 주요 사례 및 문제 해결 방법을 예시한 것으로 API의 경우는 요청/응답 방식으로 구현하므로 응답 분석을 통해 문제 해결을 시도해 볼 수 있다.

<표 4-12> 인공지능서비스 인터페이스 연동 방식과 문제 해결 방법 예시

연계 방식	설명	주요 사례	문제 해결 방법
EAI(Enterprise Application Integration)	서로 다른 플랫폼 및 애플리케이션들 간 정보 전달, 연계, 통합을 가능하게 해 주는 아키텍처	DB2DB 어댑터 등	로그 분석
ESB(Enterprise Service Bus)	애플리케이션 간의 통합 측면에서 유사 단 서비스 중심의 통합을 지향하는 아키텍처	Webservice 어댑터 등(데이터 유형 XML, Json 등)	로그 분석
API	ESB의 개념을 기반으로, ESB의 단점을 보완한 아키텍처 SON/REST 기반, 최소한의 기능을 처리하기 위한 경량화 서비스	XML, API 호출(HTTP REST)	응답 분석
RPC	필요 당사자들 간에 사전에 정의한 규격을 준수하여 원격에서 프로세스 내의 주소 공간에 위치한 함수를 호출 가능하도록 지원하는 통신 기술	Java의 RMI 등	디버깅, 로그 분석

## 2. 문제 해결 방법 유형

### (1) API 응답 분석을 통한 문제 해결

API 응답 분석에는 크게 두 가지로 분류하여 검토할 수 있어야 한다. 하나는 HTTP 자체의 응답 코드 분석이고, 다른 하나는 인터페이스 응답 코드 분석이다.

#### (가) HTTP 응답 코드 분석

HTTP 응답 코드는 RFC 2616 국제 표준을 통해 정의하고 있으며, 상태 코드라고 표현한다. 상태 코드는 3자리 숫자로 만들어져 있는데 첫 번째 자리는 1에서 5까지 제공되고, 첫 번째 자리가 4XX, 5XX의 경우에 대해 서버의 상황 점검이 필요함을 나타내는 정보이다.

1xx(정보): 요청을 받고 프로세스를 계속해서 진행

2xx(성공): 성공적으로 요청을 받았으며 요청에 대한 인식과 수용

3xx(리다이렉션): 요청 사항의 완료를 위해 추가적인 조치 필요

4xx(클라이언트 오류): 클라이언트로부터 요청된 내용 중에서 문법 오류 또는 요청을 처리할 수 없는 오류

5xx(서버 오류): 클라이언트의 유효 요청에 대해 서버의 명백한 실패

#### HTTP 404 Error와 HTTP 500 Error

**404번 오류:** 4xx(클라이언트 오류)의 대표적인 오류로 404를 들 수 있다. HTTP 404 오류는 클라이언트 측에서 잘못된 페이지 요청을 할 때 발생하는 오류이다. (예: 서버에 구현된 10.html을 요청해야 하는데 서버에 존재하지 않은 1.html을 요청할 경우 발생)

**500번 오류:** 5xx(서버 오류)의 대표적인 오류로 500을 들 수 있다. Internal Server Error는 서버의 프로그램 코딩 오류, 서버 통신 Timeout, 트래픽 과부하 등의 원인으로 발생한다.

HTTP 응답 코드의 분석을 통해 문제를 해결할 수 있어야 한다.

#### (나) 인터페이스 응답 코드 분석

인터페이스 응답 코드의 경우, 인터페이스를 제공하는 곳에서 정의하여 배포하는 경우가 대부분이므로 해당 인터페이스 사용 명세를 면밀하게 분석하고 이해할 수 있어야 한다.

### (2) 디버깅을 통한 문제 해결

통상 개발 및 단위 테스트 과정 중에 코드에 포함된 버그를 제거하는 것을 의미한다. 디버깅을 수행하는 방법은 여러 가지가 있는데 예를 들어 오타를 찾는 코드 검사, 코드 분석기를 사용한 디버그, 성능 프로파일러를 사용한 코드 디버깅을 들 수 있다.

### (3) 로그 분석을 통한 문제 해결

EAI/ESB 아키텍처의 중요한 요소 중의 하나는 트랜잭션 추적과 에러 처리 방식의 이해이다.

(가) 트랜잭션 로그

트랜잭션 로그는 EAI/ESB 시스템 장애가 발생할 때 송수신 시스템과 트랜잭션 내용을 맞춰서 장애/문제를 복구하는 데 사용한다. 트랜잭션 로그에 사용되는 트랜잭션 ID는 전사 표준 전문의 헤더에 정의하는 것이 일반적이고 이 트랜잭션 ID는 송신에서 수신 시스템까지 하나의 공통된 ID를 사용해야 한다.

(나) 에러 처리 로직

에러 처리 로직의 요건은 문제 감지, 문제의 원인 리포팅, 문제 해결 3단계로 이루어진다. 모든 종류의 시스템 문제를 신속하게 감지하는 것이 가장 중요한 역할이다. 문제 해결에 대한 처리 정책을 Fault-Policy라 하고 인터페이스마다 정의하여 인터페이스별로 문제를 처리할 수 있도록 한다.

(다) 장애 처리 정책

인터페이스와 관련하여 발생하는 문제들은 일반적으로 <표 4-13>과 같이 4가지 정책으로 정의하여 처리한다.

<표 4-13> 테스트 케이스 구성 요소(IEEE 829)

정책	설명	비고
무시	오류를 무시하고 인터페이스 제거	
보고	결함(오류/문제) 보고, 선택적 알림 전송(email, sms 등)	
재시도	일정(지정된) 시간 이후 정의된 시간에 해당 트랜잭션을 자동으로 재시도	재시도가 실패할 경우에 대해 오류 처리 정책의 정의 필요
수작업	결함(오류/문제) 보고 및 사용자가 무시, 보고, 재시도 등 정책을 선택하도록 요청	

무시하거나(Ignore), 장애/문제 내용을 관리자에게 알리거나(Report), 자동으로 재시도하거나(Retry), Work list 등에 추가하여 관리자가 에러 처리를 수동으로 처리하는 방식으로 정리할 수 있다.

## 수행 내용 / 인공지능서비스 인터페이스 연동 문제 해결하기

### 재료·자료

- 요건 분석서 / 비즈니스 로직 설계서 / 아키텍처 정의서
- 인공지능서비스 인터페이스 연동 문제 해결을 위한 개발 도구(파이참 커뮤니티 버전)
- 운영 체제 - 윈도 10 환경

### 기기(장비·공구)

- 컴퓨터
- 문서 열람 도구(ODF 등)

### 안전·유의 사항

- 인터페이스 오류 코드 및 메시지는 사용 명세를 면밀하게 분석하고 이해해야 한다.
- http 상태 코드와 인터페이스 오류 코드를 분류하여 문제를 해결할 수 있어야 한다.

### 수행 순서

#### ① 디버깅을 수행한다.

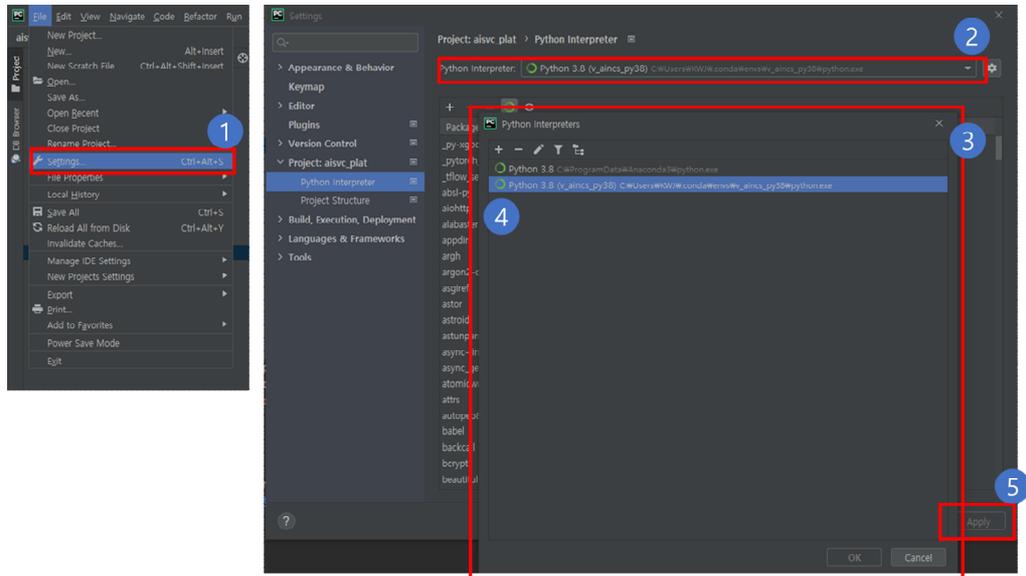
인공지능서비스 인터페이스 연동에서 공공 인공지능 포털의 구현 예제 코드를 함수화하고 해당 함수를 호출한 코드를 예제로 사용한다.

#### 1. 디버깅 환경을 구성한다.

파이선 개발 환경에서 인터페이스 연동 관련 함수의 디버깅 환경을 구성한다. 두 가지 환경 구성이 필요하다. 첫 번째는 가상 환경에 대한 구성이고, 두 번째는 디버깅 구성이다. 관련 구현 소스 코드는 참조를 통해 확인한다.

#### (1) 가상 환경을 구성한다.

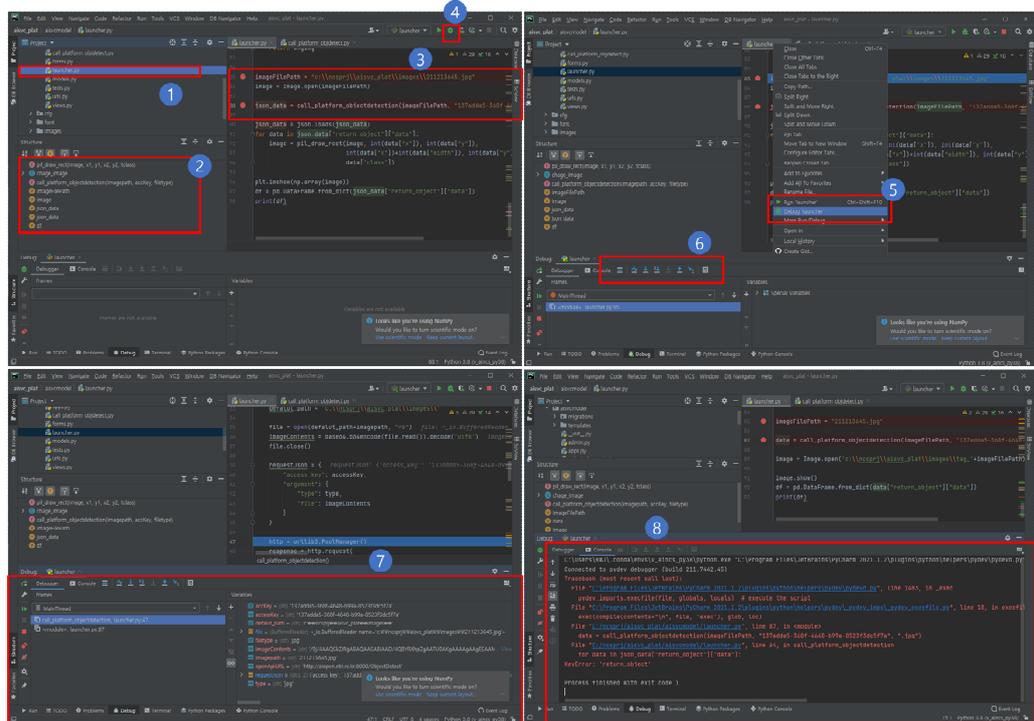
오픈 소스 개발 도구인 파이참(pycharm-평가 버전)의 환경 설정(File → Settings)①으로 이동하면 프로젝트의 파이선 인터프리터 옵션을 선택할 수 있다. 파이선 인터프리터에서 ②를 선택하고 “show all”을 선택하면 ③창이 뜨고, 해당 창에서 가상 환경을 선택하고 “OK④” 클릭 후 “Apply⑤”를 클릭하면 구성이 완료된다.



출처: 집필진 제작(2021)  
 [그림 4-12] 파이참(Community 버전) 개발 도구에서 가상 환경 설정 방법

(2) 디버깅 환경을 구성한다.

디버깅은 파이선 코드를 디버깅 대상으로 가정하고 [그림 4-13]과 같이 파이선 프로그램을 디버깅 환경으로 구성하는 것을 예시한다.



출처: 집필진 제작(2021)  
 [그림 4-13] 파이참(Community 버전) 개발 도구에서 디버 기능 수행 내용 예시

예제 파이선 파일①과 파이선의 내부 구조②를 확인하고, 파이선 코딩의 디버깅이 필요한 곳에 중단점(breakpoint, 코딩라인과 코딩 사이를 클릭하면 체크, 그림의 작은 붉은 동그라미 점) 표시③를 한다. 이후 ④ 또는 ⑤와 같이 디버깅 버튼을 실행하면 중단점에서 코딩 실행이 멈추며 ⑥과 같은 조작을 통해 각 변수들의 현재 상댓값⑦을 확인할 수 있게 된다. 이런 과정을 통해 소스 코드가 정상적으로 작동하는지 여부를 점검한다.

② API 응답 결과를 기반으로 문제를 해결한다.

인공지능서비스 플랫폼 연동과 관련하여 응답 결과를 바탕으로 하여 문제 해결 하는 방법을 수행해 본다.

1. 파이선 구현 코드를 실행한다.

다음 코드를 파이선 실행 환경으로 구동해 본다.

### 공공 인공지능 포털의 객체 검출 예제코드

```
# -*- coding:utf-8 -*-
import urllib3, json, base64
ApiURL = "http://aiopen.etri.re.kr:8000/ObjectDetect"
accessKey = "137adda5-####-####-####-8523f3dc5f7a"
imgFilePath = 'c:\\ncsprj\\aisvc_plat\\images\\211213645.jpg'
type = ".jpg"
fl = open(imgFilePath, "rb")
imageContents = base64.b64encode(fl.read()).decode("utf8")
fl.close()
reqJson = {
    "access_key": accessKey,
    "argument": {
        "type": type, "file": imageContents
    }
}
http = urllib3.PoolManager()
res = http.request(
    "POST", ApiURL,
    headers={"Content-Type": "application/json; charset=UTF-8"},
    body=json.dumps(reqJson)
)
print("[responseCode] " + str(res.status))
print("[responBody]")
print(res.data)
```

Access Key가 유효하지 않다는 응답을 회신받았다. 이럴 때 해결 방법은 정확한 Access Key를 입력하는 것이다.

### 오류 결과 - 1

```
~~\_aincs_py38\python.exe C:/ncsprj/aisvc_plat/aisvcmodel/example_coding.py
[responseCode] 200
[responBody]
b'{"request_id":"","
  "result":-1,
  "reason":"Invalid Access Key!:137adda5-####-####-####-8523f3dc5f7a"}'
Process finished with exit code 0
```

이전 장에서 공공 인공지능 Open API 연동 환경 구성을 통해 확인한 정확한 Access Key를 입력 후 재실행해 본다. 실행하면 다음과 같이 응답 코드 400, result는 -1, reason은 “File Type Not Support”로 나타난다. 메시지를 살펴보면 File Type을 지원하지 않는다는 의미로 해석된다. 해당 파일 타입에 임의로 “.jpg”로 했지만, “.”을 빼고 “jpg”만 요청 파라미터값으로 입력하도록 요구하고 있다.

### 오류 결과 - 2

```

~~\v_aincs_py38\python.exe C:/ncsprj/aisvc_plat/aisvcmodel/example_coding.py
[responseCode] 400
[responBody]
b>{"result":-1,"reason":"File Type Not Support!"}
Process finished with exit code 0
    
```

객체 검출에는 <표 4-14>와 같이 다양하게 오류 명세를 제시하고 있다. 해당 오류 명세를 참고하여 문제를 해결할 수 있다.

<표 4-14> 객체 검출 API 오류 명세

http 상태 코드	결과	이유	설명
400	-1	Required arguments is empty	파라미터 중 필수로 요구한 값이 없는 경우
400	-1	One or more arguments are not valid	파라미터의 값이 유효하지 않는 경우
413	-1	Request Entity Too Large	서버가 처리할 수 있는 요청 문장 및 어휘 크기에 비해 더 큰 경우
429	-1	Too Many Requests	사용자가 주어진 시간 내에 서버에 너무 많은 요청을 하는 경우
404	-1	Unknown Handler	등록되지 않는 서비스를 요청한 경우
408	-1	Handler Timeout	서버에 요청 후 대기 시간을 초과한 경우
500	-1	ETRI service server connection refused	ETRI 분석 서버에서 요청을 받지 못하는 경우
500	-1	ETRI service server is not exists	수행 가능한 ETRI 분석 서버가 없는 경우
500	-1	Recipient Failure	ETRI 분석 서버에서 요청을 처리하지 못하는 경우
500	-1	Unknown ReplyFailure	API 요청에 알 수 없는 내부 오류가 발생한 경우
500	-1	Unknown Exception	알 수 없는 내부 오류가 발생한 경우

출처: 공공 인공지능 포털(2020. 11. 20.), 객체 검출 오류 코드. [https://aiopen.etri.re.kr/guide\\_od.php](https://aiopen.etri.re.kr/guide_od.php)에서 2021. 08. 07. 검색.

### 수행 tip

---

- 문제 해결을 위해 가장 먼저 살펴보아야 할 것은 공공 인공지능 포털에서 공개하고 있는 오류 명세를 자세하게 살펴보는 일이다.
- 오류 명세에 포함하지 않는 오류 코드가 포함된 경우는 메시지, 오류 위치 등 파악을 위해 디버깅을 수행해 본다.

### 교수 방법

- 인공지능서비스 인터페이스에 대해 설명하고, 인터페이스 연동 방법이 서비스 기업별로 다를 수 있으므로 연동 방식(API, RPC 등)을 제시한 후 수업을 진행한다.
- 인공지능서비스 인터페이스 연동을 위한 개발 도구를 우선 선정하여 수업을 진행한다.
- 공공 인공지능 포털을 방문하여 다양한 인공지능 API 명세를 제시한 후 설명한다.
- 인공지능서비스 인터페이스 연동 실습이 이루어질 수 있도록 지도한다.
- 설계 내용을 파악하고 이를 개발 언어를 활용하여 작성할 수 있는 절차를 구체적으로 제시한 후 내용에 대해 설명한다.
- 구현된 로직을 테스트할 수 있도록 디버깅 방법을 실습하고 실제 구현 코드에 대해 인터페이스 연동 부분에 대해 디버깅 및 문제를 해결할 수 있도록 설명한다.

### 학습 방법

- 인공지능서비스 인터페이스에 대해 이해하고, 인터페이스 연동 방식(API, RPC 등)을 파악한 후 인터페이스 연동 구현을 실습한다.
- 인공지능서비스 인터페이스 연동을 위한 개발 도구의 사용 방법을 학습한다.
- 공공 인공지능 포털을 방문하여 다양한 인공지능 API 명세를 학습하고 특정 인공지능 API를 선정하여 인터페이스 연동을 실습한다.
- 설계 내용을 파악하고 이를 개발 언어를 활용하여 작성할 수 있도록 학습하고 실제 소스 코드를 구현한다.
- 구현된 로직을 테스트할 수 있도록 디버깅 환경을 구성하고 실제 구현 코드에 대해 인터페이스 연동 부분에 대해 디버깅 및 문제를 해결한다.

## 학습 4    평 가

### 평가 준거

- 평가자는 학습자가 학습 목표를 성공적으로 달성하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습 내용	학습 목표	성취수준		
		상	중	하
인터페이스 연동 환경 구성	- 인공지능서비스 설계에 따라 인공지능서비스 인터페이스 연동 환경을 구성할 수 있다.			
인터페이스 연동	- 인공지능서비스 인터페이스 가이드에 따라 애플리케이션 서비스에 인터페이스를 연동할 수 있다.			
인터페이스 연동 문제 해결	- 인공지능서비스 인터페이스 연동 기능의 정상 작동 여부를 확인하고 문제 발생 시 문제를 해결할 수 있다.			

### 평가 방법

- 포트폴리오

학습 내용	평가 항목	성취수준		
		상	중	하
인터페이스 연동 환경 구성	- 인공지능서비스 인터페이스 연동 환경을 구성하는 능력			
	- 인공지능서비스 인터페이스 연동을 위해 인터페이스에서 제시하는 연동 관련 안내를 이해하는 능력			
인터페이스 연동	- 인공지능서비스 인터페이스 연동을 위해 인터페이스에서 제시하는 연동 관련 명세(예: 요청/응답/오류 등)를 이해하는 능력			
	- 개발 도구를 이용하여 인터페이스에서 제시하는 구현 사례를 비즈니스 로직에 포함하여 구현할 수 있는 능력			
인터페이스 연동 문제 해결	- 개발 도구를 활용하여 인공지능서비스 인터페이스 연동을 통해 구현 결과물을 표출하는 능력			
	- 인공지능서비스 인터페이스 연동 문제 해결을 위해 개발 도구에 디버깅 환경을 구성하는 능력			
	- 인터페이스에서 제시하는 오류와 관련 명세를 이해하고 문제를 해결할 수 있는 능력			
	- 인터페이스 연동 관련 구현 코드를 디버깅할 수 있는 능력			

• 문제 해결 시나리오

학습 내용	평가 항목	성취수준		
		상	중	하
인터페이스 연동 환경 구성	- 인공지능서비스 인터페이스 연동과 관련하여 교재에서 제시한 인터페이스 외에 타 인터페이스 연동을 위한 환경 구성 능력			
	- 타 인터페이스 연동 환경 구성 사항들을 팀원들에게 공유할 수 있는 능력			
인터페이스 연동	- 객체 검출 API 외에 다른 인공지능 API를 이용하여 서비스를 구성하고 실행할 수 있는 능력			
	- 인공지능서비스 인터페이스 연동과 관련 제시한 인터페이스에서 타 인공지능 API를 팀별 프로젝트로 수행하여 실행 결과를 제시할 수 있는 능력			
	- 공공 인공지능 포털 외에 타 인터페이스의 인공지능서비스를 연동하여 실행 결과를 제시할 수 있는 능력			
인터페이스 연동 문제 해결	- 타 인터페이스의 연동 환경을 구성하고 발생된 문제를 해결하여 팀원들과 공유할 수 있는 능력			
	- 타 인터페이스의 인공지능 API를 활용하여 구현하고 발생된 문제를 해결할 수 있는 능력			

• 사례 연구

학습 내용	평가 항목	성취수준		
		상	중	하
인터페이스 연동 환경 구성	- 제시한 공공 인공지능 포털 외에 타 인터페이스를 조사하고 연동할 수 있는 방법을 제시할 수 있는 능력			
	- 타 연동 환경과 교재의 연동 환경 간 차이점을 설명할 수 있는 능력			
인터페이스 연동	- 파이선 코딩 외에 Java, C++, Node.js, PHP 등 타 언어를 활용하여 인공지능서비스 인터페이스를 연동할 수 있는 능력			
	- 교재에서 제시한 인터페이스 외에 타 인터페이스에서 제시한 사례 코드를 바탕으로 하여 연동 환경을 구현한 사례 학습 능력			
인터페이스 연동 문제 해결	- 로그 분석을 통한 문제 해결 사례를 연구할 수 있는 능력			
	- 디버깅을 통한 문제 해결 사례 조사 및 연구 능력			
	- 다양한 인터페이스에서 제시하고 있는 오류 코드에 대해 조사 및 분석하고 정리할 수 있는 능력			

## 피드백

### 1. 포트폴리오

- 실습 과정을 문제 해결 시나리오에 따라 평가한 후에 개선해야 할 사항 등을 정리한 후 학습자에게 설명한다.
- 설계서를 파악하여 작성된 인터페이스 연동 로직이 설계서대로 개발되었는지 확인하고, 학습자에게 설명한다.
- 설계서를 파악하여 작성된 인터페이스 연동 로직이 설계서대로 개발되었는지 확인하고, 학습자에게 설명한다.
- 평가 결과가 미흡할 때에는 인터페이스 연동 구현 절차 및 방법, 작동 원리에 대해 구체적으로 설명해 준다.
- 성취수준이 낮은 학습자들에게는 실제 구현 코드 라인 단위로 구체적으로 설명해 준다.
- 성취수준이 높은 학습자의 경우 좀 더 복잡한 인공지능 인터페이스 연동(대화 지능 등)에 대하여 심화 설명해 준다.

### 2. 문제 해결 시나리오

- 제출한 인공지능서비스 인터페이스 연동 로직 구현 결과를 평가한 후에 주요 사항에 대해 의견을 제시한다.
- 구현된 인터페이스 연동 기능이 정상적으로 동작하는지를 보고 비정상 동작의 원인과 조치 방안에 대해 설명한다.
- 인공지능서비스 인터페이스 연동 로직 구현 결과가 '상'인 경우 모범 사례로 학습자들에게 공유한다.
- 인공지능서비스 인터페이스 연동 로직 구현 결과가 '중'인 경우 부족한 부분을 상세히 설명하고 보완할 수 있도록 한다.
- 인공지능서비스 인터페이스 연동 로직 구현 결과가 '하'인 경우 모범 사례를 참조하여 다시 구현할 수 있도록 유도한다.

### 3. 사례 연구

- 제시한 공공 인공지능 포털 외에 인공지능 Open API를 제공하는 타 사례를 조사하고 연동할 수 있는 방법을 조사 및 제출하게 한다.
- 제출 결과에 대해 평가 및 토론하게 하고 미흡한 사례 조사 내용에 대해 구체적으로 설명해 준다.
- 타 연동 환경과 교재의 연동 환경 간 차이점을 팀원들에게 설명할 수 있는 시간을 갖게 하고 성취수준이 낮은 학습자들에게는 연동 환경 간 차이점을 설명해 준다.
- 파이선 코딩 외에 Java, C++, Node.js, PHP 등 타 언어를 활용한 사례를 조사하고 팀원들과 공유하도록 한다.
- 연동 구현 관련 성취수준이 높은 학습자의 경우 타 언어 활용, 교재의 인공지능 모델 외에 타 모듈 연계 방법에 대해 심화 설명해 준다.
- 문제 해결 방법으로 교재에서 제시한 Log 분석, 디버깅 및 응답 오류 코드에 대해 조사 및 분석하고 정리하여 팀원들과 공유하도록 하고, 성취수준이 낮은 학습자들에게는 응답 오류 코드 조사 및 분석 결과에 대해 설명해 준다.



- 강수철(2020. 2. 19.). 인공지능 전이 학습과 응용 분야 동향. [www.iitp.kr](http://www.iitp.kr).
- 공공 인공지능 포털(2021. 06. 07.). 공공 인공지능 인터페이스 연동. <https://aiopen.etri.re.kr/>에서 2021. 06. 07. 검색.
- 공공 인공지능 포털(2020. 11. 25.). 요청 명세. [aiopen.etri.re.kr/guide\\_od.php](http://aiopen.etri.re.kr/guide_od.php)에서 2021. 6. 12. 검색.
- 공공 인공지능 포털(2020. 11. 25.). 응답 명세. [aiopen.etri.re.kr/guide\\_od.php](http://aiopen.etri.re.kr/guide_od.php)에서 2021. 6. 12. 검색.
- 공공 인공지능 포털(2020. 11. 20.). 객체 검출 오류 코드. [https://aiopen.etri.re.kr/guide\\_od.php](https://aiopen.etri.re.kr/guide_od.php)에서 2021. 08. 07. 검색.
- 교육부(2019). 『인공지능 플랫폼 기능 구현(LM20010700105\_18v1)』. 한국직업능력개발원.
- 교육부(2019). 『인공지능서비스 활용 기획(LM2001070206)』. 한국직업능력개발원.
- 박외진(2018. 10. 03.). 「인공지능 플랫폼의 개념과 도입 전략」. 정보통신기술진흥센터.
- 박찬홍(2020. 06. 17.). 「인공지능 헬스케어 산업의 실용화 동향」. 정보통신기획평가원.
- 손재기(2020. 11. 25.). 「인공지능 OpenAPI 서비스 프레임워크 기술」. 정보통신기술진흥센터.
- 이진휘(2020. 05. 25.). 「AI 기술 동향과 오픈 소스」. 정보통신산업진흥원.
- 한국지능정보사회진흥원 정책본부 AI·미래전략센터(2021. 2. 23.). 「초대규모 AI 모델 GPT-3의 부상과 대응 방안」. 한국지능정보사회진흥원(NIA).
- ANACONDA(<https://www.anaconda.com/products/individual#download-section>). 2021. 06. 02. 검색.
- ETRI(2021. 08.). 오픈 API 목록. [https://aiopen.etri.re.kr/service\\_list.php](https://aiopen.etri.re.kr/service_list.php)에서 2021. 08. 07. 검색.
- pypi(2021. 06. 26.). python 패키지 및 설치 명령어. <https://pypi.org/>에서 2021. 06. 20. 검색.
- pytorch(2021. 06. 26.). pytorch 설치 명령어. <https://pytorch.org/get-started/locally/>에서 2021. 06. 15. 검색.
- TIOBE(2021년). TIOBE Index. <https://www.tiobe.com/tiobe-index/>에서 2021. 06. 15. 검색.
- tutorials(2021. 06. 05.). 인공지능 모델 연동 예제. <https://tutorials.pytorch.kr/>에서 2021. 06. 12. 검색.

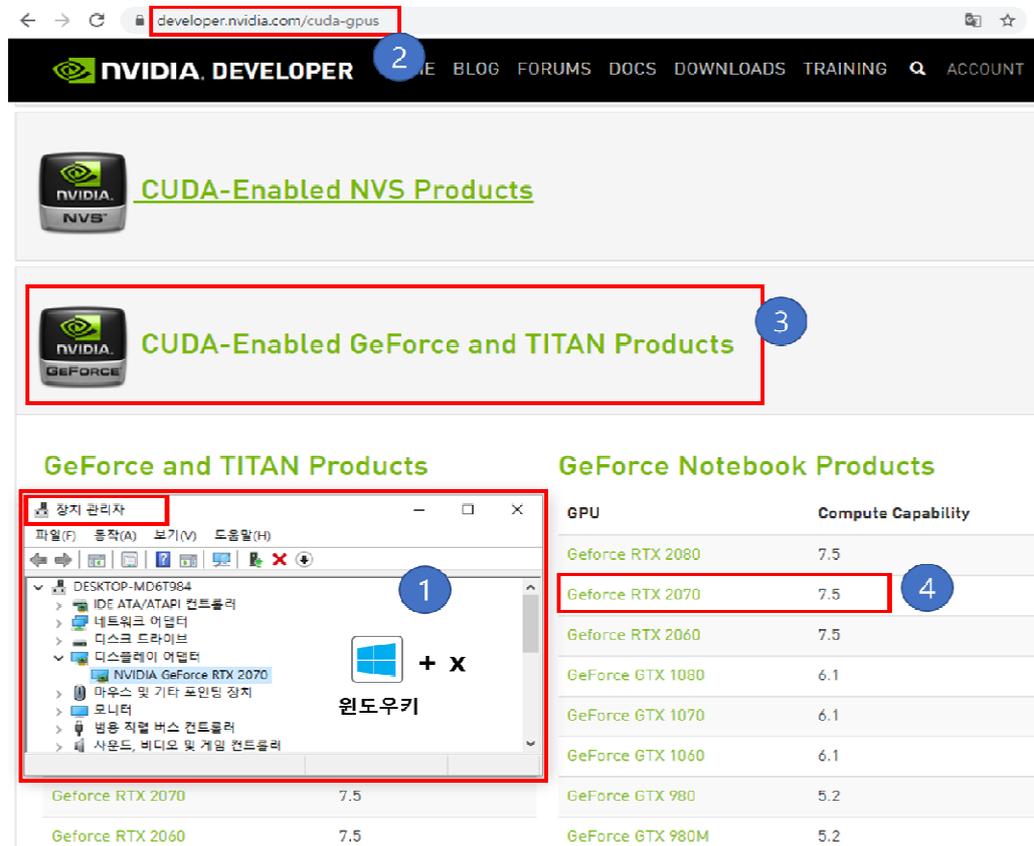


## 학습 1-1. 인공지능서비스 개발 환경 이해

① (개인 컴퓨터)인공지능서비스 개발 환경을 구성한다.

1. GPU 기반 개발 환경을 구성한다.

(1) GPU 모델 확인 및 CUDA 지원 여부를 확인한다. (GPU가 장착되지 않은 경우(2)번 진행)  
GPU가 CUDA 툴킷을 활용할 수 있는지 개인 컴퓨터의 디스플레이 어댑터(①)확인  
https://developer.nvidia.com/cuda-gpus(②) URL에 접속하여 “cuda-Enabled GeForce and TITAN Products”(③)를 클릭 후 개인 컴퓨터의 디스플레이 모델과 동일한 모델이 있는지 확인(④)한다.



출처: 집필진 제작(2021)

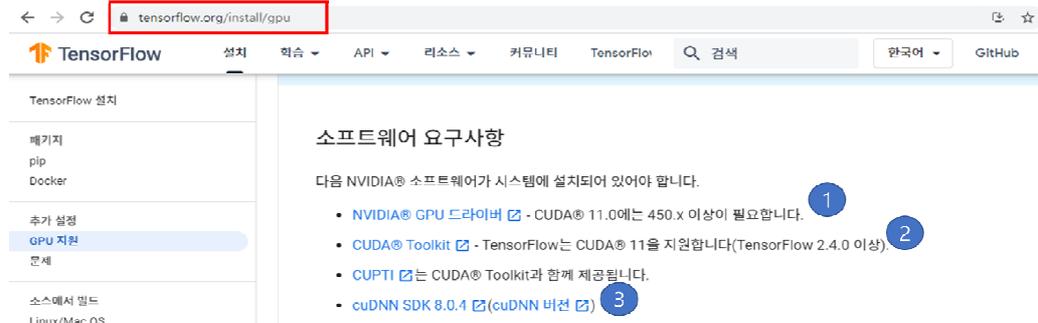
[그림 부록-1] 개인 컴퓨터의 GPU가 CUDA Toolkit을 지원하는지 여부 확인 과정 ① ~ ④

### GPU가 장착되어 있지 않은 경우

CUDA Toolkit과 Tensorflow-gpu를 설치하지 않고, Tensorflow-cpu 버전을 설치하여 구성한다.

(2) 아나콘다 소프트웨어를 설치한다.

(3) 텐서플로 설치를 위한 사전 호환성을 검토한다. (GPU가 장착되지 않은 경우(4)로 이동) 본 장에서는 텐서플로를 설치하기 위해 CUDA, cuDNN, Visual Studio와의 호환성을 먼저 검토한다.

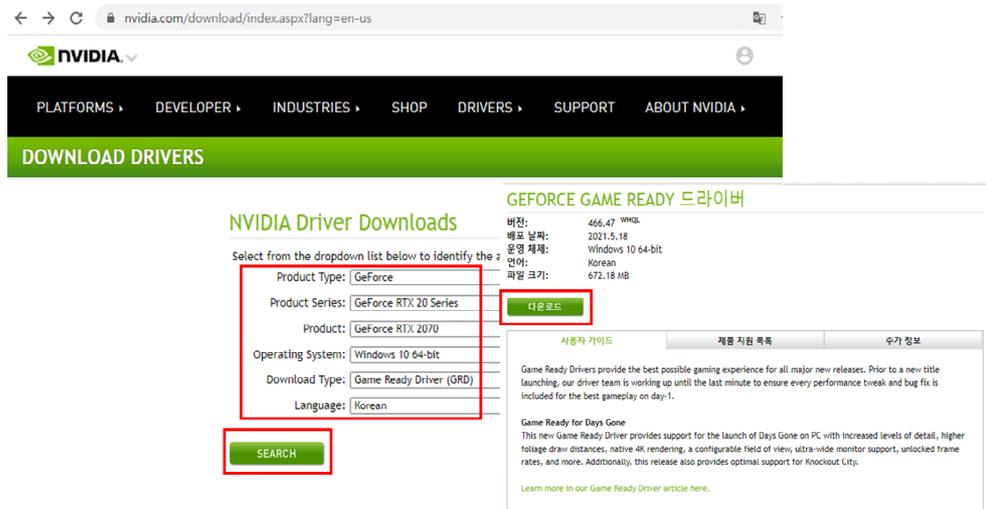


출처: 집필진 제작(2021)

[그림 부록-2] 텐서플로 설치 안내 문서(<https://www.tensorflow.org/install/gpu>)

(가) NVIDIA GPU 드라이버 설치

텐서플로 설치 안내 문서의 ① “NVIDIA® GPU 드라이버”를 클릭하면 다음 그림과 같이 개인 컴퓨터에 장착된 GPU 모델을 선택하여 검색 버튼을 클릭하면 드라이버 다운로드 창으로 이동하게 된다.



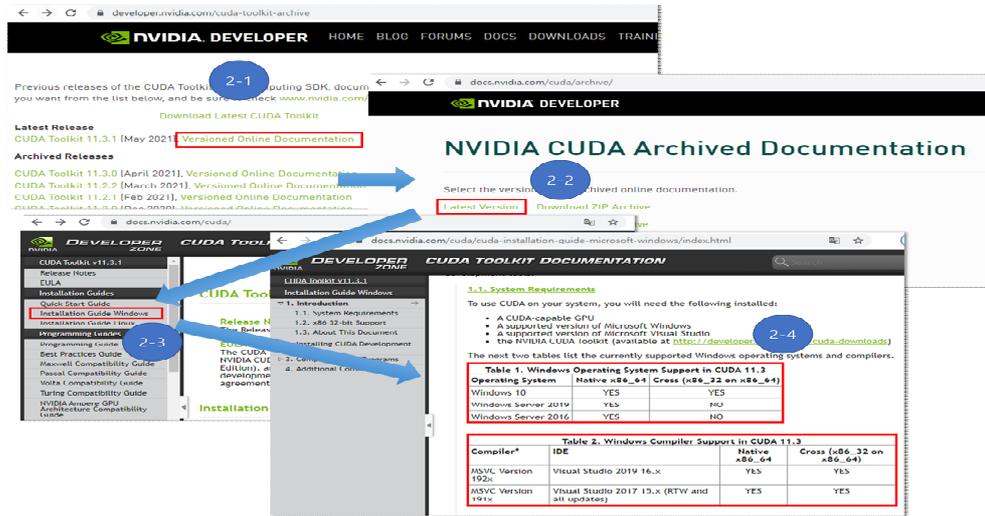
출처: 집필진 제작(2021)

[그림 부록-3] NVIDIA 드라이버 다운로드 화면

“CUDA 11에는 450.x 이상이 필요합니다.”라고 명시가 되어 있으므로 개인 컴퓨터에 설치된 NVIDIA GPU 드라이버의 버전은 450.x 버전 이상이 설치되어야 하고, 그렇지 않으면 다운로드 페이지에서 GPU 드라이버를 다운로드 후 드라이버를 설치한다.

(나) CUDA Toolkit 확인

텐서플로 설치 안내 문서 CUDA® Toolkit은 [그림 부록-2] ②와 같이 TensorFlow 2.4.0 이상에 대해 CUDA 11을 지원한다고 안내하고 있다. 해당 항목을 클릭하면 다음 화면의(2-1)과 같이 CUDA Toolkit 11.3.1 버전의 최신 정보를 확인할 수 있다. 오른쪽의 “Versioned Online Documentation”을 클릭하면 상세한 요구 사항을 확인할 수 있다.



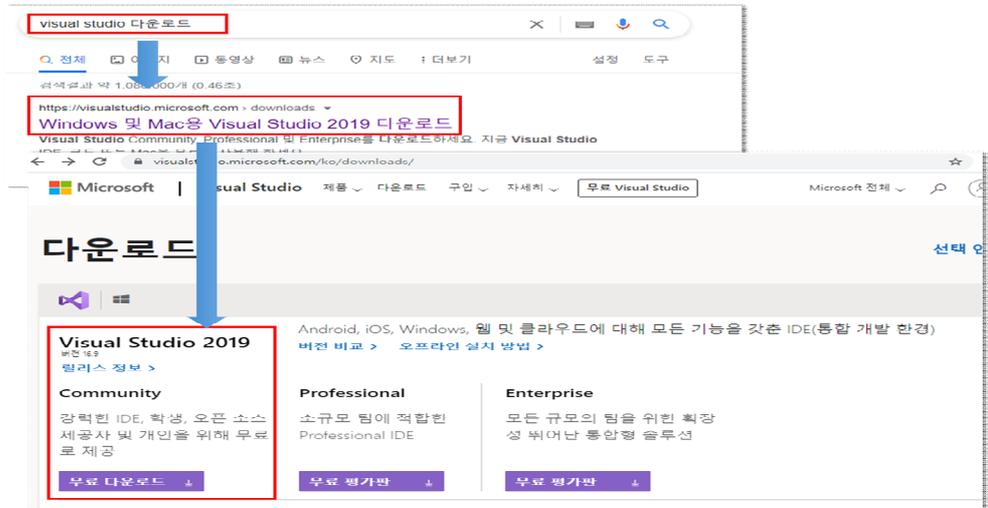
출처: 집필진 제작(2021)

[그림 부록-4] CUDA Toolkit 설치를 위한 시스템 요구 사항 확인 문서(버전 확인 필요)

CUDA 11.3이 윈도우 10에서 지원하는 것과 컴파일러 도구로 비주얼 스튜디오 2019 16.x 버전을 안내하고 있으므로 해당 버전을 설치한다.

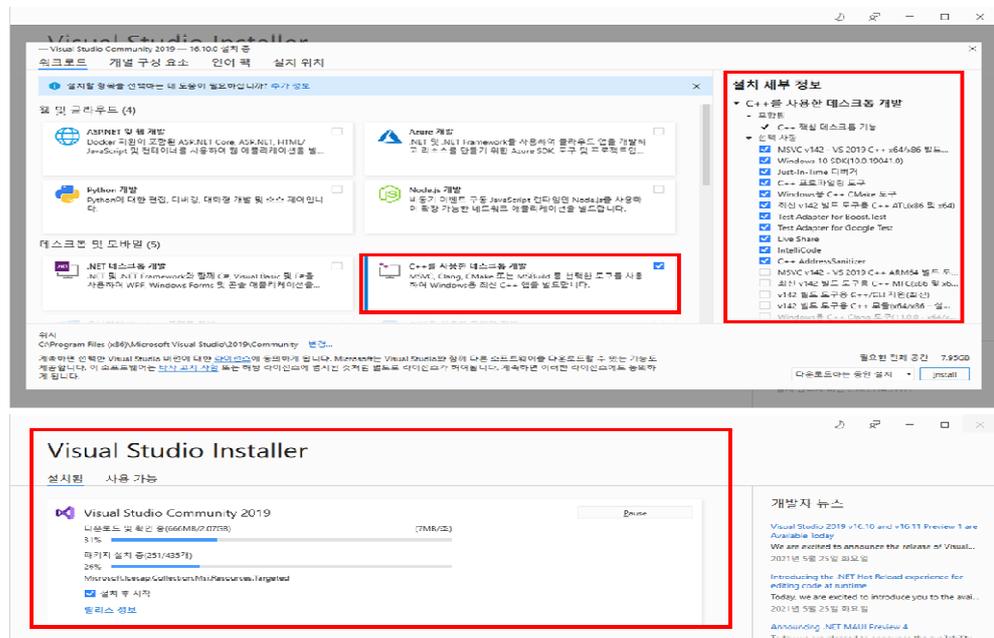
(다) Visual Studio 2019 16.x 설치

Visual Studio Community 버전을 다운로드 및 해당 파일을 더블 클릭하여 설치한다.



출처: 집필진 제작(2021)  
[그림 부록-5] Visual Studio 다운로드 화면

설치 화면이 나타나면, [그림 부록-6]과 같이 “C++를 사용한 데스크톱 개발”을 체크한 후 설치를 완료한다.

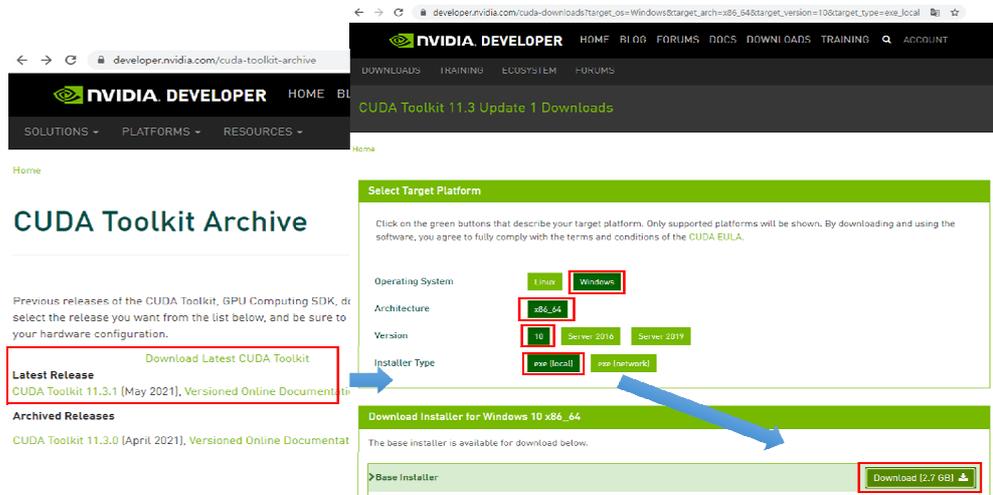


출처: 집필진 제작(2021)  
[그림 부록-6] Visual Studio 설치 화면

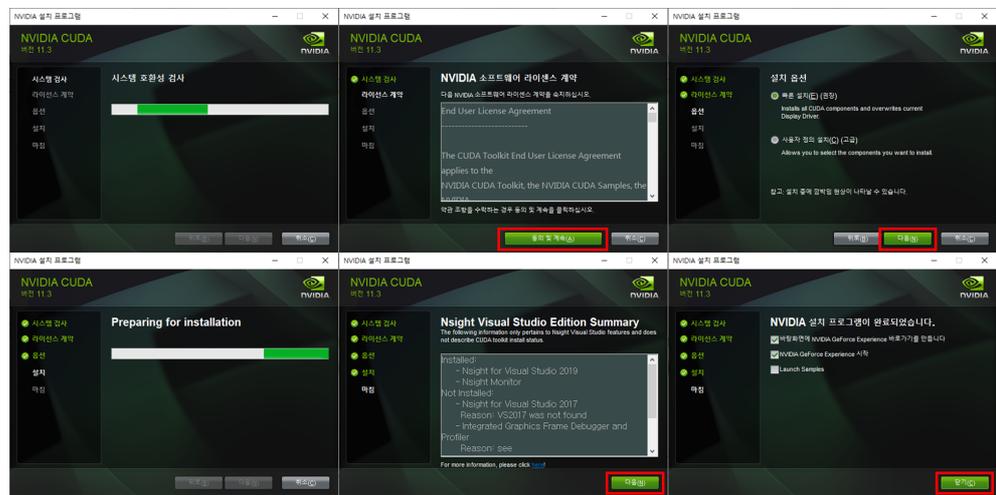
### (3) CUDA 및 cuDNN 설치

#### (가) CUDA Toolkit 설치

[그림 부록-2] TensorFlow 설치 안내 문서(<https://www.tensorflow.org/install/gpu>)로 돌아가 CUDA Toolkit을 클릭한 후 다음과 같이 개인 컴퓨터 환경(예제: Windows, x86\_64, 1.0, exe)에 맞는 값을 선택 후 다운로드하고, 완료되면 다운로드 파일을 관리자의 권한으로 실행 및 설치를 완료한다.



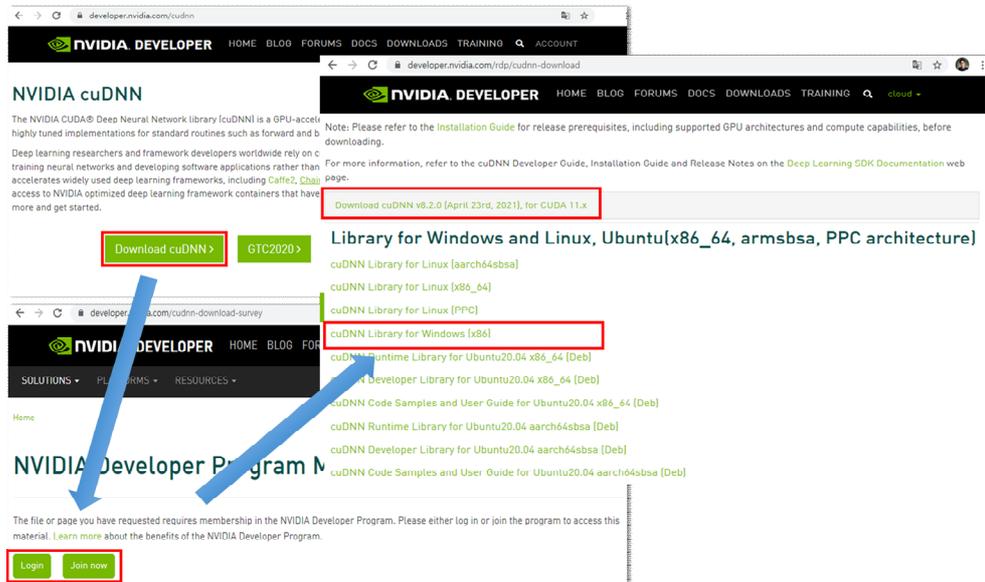
출처: 집필진 제작(2021)  
[그림 부록-7] CUDA Toolkit 다운로드 화면



출처: 집필진 제작(2021)  
[그림 부록-8] CUDA Toolkit 설치 화면

#### (나) cuDNN 설치

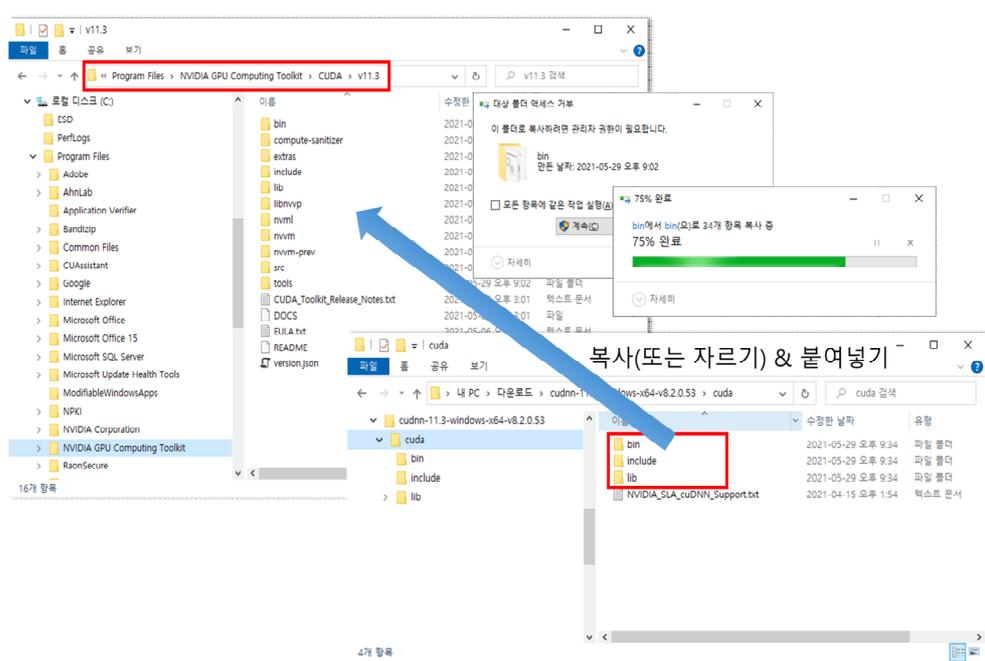
[그림 부록-2] TensorFlow 설치 안내 문서(<https://www.tensorflow.org/install/gpu>)로 돌아간다. CUPTI는 CUDA Toolkit과 같이 제공된다고 안내하고 있으므로 별도로 설치할 필요는 없다. cuDNN SDK를 클릭하면, cuDNN 페이지로 연결되며, Download cuDNN 버튼을 클릭하고, 이어서 Login 버튼을 클릭하여 로그인(회원 가입 필요) 후 cuDNN Library for Windows를 클릭하여 다운로드한다.



출처: 집필진 제작(2021)  
[그림 부록-9] cuDNN 다운로드 화면

다운로드받은 파일의 압축풀기 후 cuda 폴더 밑의 bin, include 및 lib 등 3개의 폴더를 CUDA가 설치된 폴더에 붙여넣기(덮어쓰우기) 하면 설치가 완료된다.

(C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.3 ← 설치 버전)



출처: 집필진 제작(2021)  
[그림 부록-10] cuDNN 설치 방법

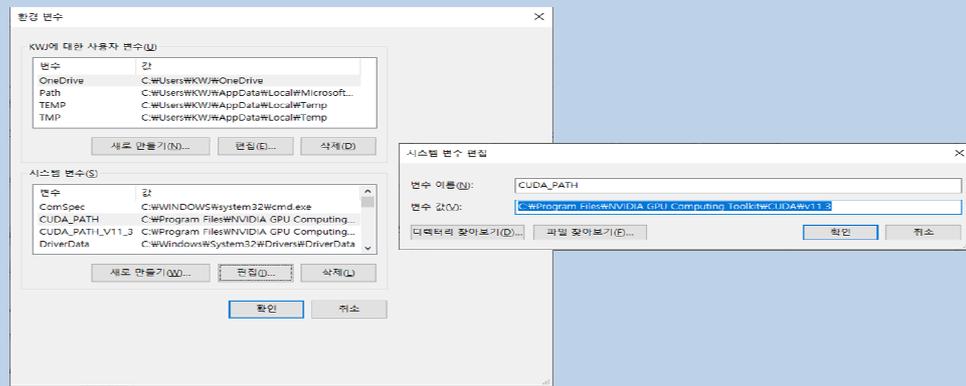
## CUDA 시스템 환경 변수 설정 확인

CUDA Path 환경 변수를 별도로 추가할 필요는 없지만, 환경 변수가 반영되었는지 확인은 필요하다. 환경 변수가 반영되지 않을 경우 파이선으로 프로그램을 구현할 때 오류가 발생한다.

방법 1) 윈도 명령 프롬프트(cmd.exe)에서 `control sysdm.cpl` → 고급 탭 선택 → 환경 변수 버튼 클릭

방법 2) (탐색창 → 내 PC 마우스 우클릭 → 속성 → 오른쪽 '관련 설정'의 고급 시스템 설정 → 고급 탭 선택 → 환경 변수 버튼 클릭)

시스템 환경 변수에 CUDA\_PATH 변수값을 잘 지정하였는지 다음의 환경 변수를 참고하여 확인한다.



출처: 집필진 제작(2021)

[그림 부록-11] 시스템 환경 변수 구성 예시

Variable Name: CUDA\_PATH

Variable Value: C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.3

← 설치 버전

(4) 가상 개발 환경 만들기 및 접속하기

(5) 텐서플로 및 기타 라이브러리 설치

(가) 주피터 노트북 설치

(나) 머신러닝/딥러닝 개발 시 활용되는 라이브러리 설치

머신러닝, 딥러닝에 자주 활용되는 관련 라이브러리를 설치한다. 해당 라이브러리를 설치할 때에는 특정 버전을 명시하여 설치할 수도 있다.

**인공지능개발에 자주 활용되는 라이브러리 설치 명령어(C:\ncs\_prj\v\_aincs\_py38) ← 생략**

(v\_aincs\_py38) `conda install -n v_aincs_py38 numpy #==1.20.2` 생략 가능, 특정버전 명시

(v\_aincs\_py38) `conda install -n v_aincs_py38 numpy scipy pandas scikit-learn matplotlib seaborn spyder h5py pillow tqdm`

(다) 텐서플로 및 케라스 설치

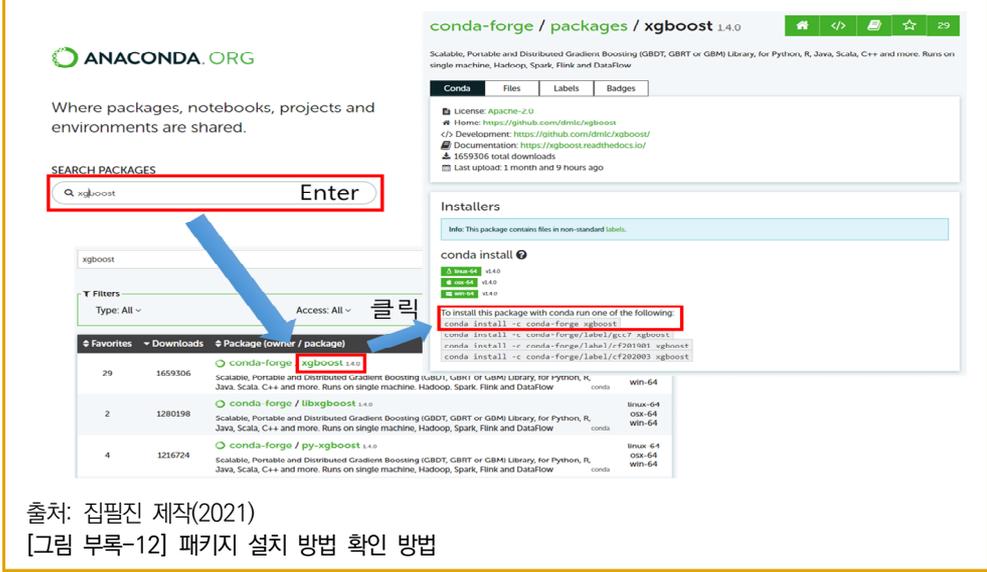
(라) 기타 라이브러리 설치

다음은 머신러닝 앙상블 기법으로 사용되는 xgboost, catboost, lightgbm 라이브러리 설치 명령어이다.

```
xgboost, catboost, lightgbm 설치 명령어(C:\ncs_prj\v_aincs_py38) ← 생략
(v_aincs_py38) conda install -n v_aincs_py38 xgboost
(v_aincs_py38) conda install -n v_aincs_py38 catboost
(v_aincs_py38) conda install -n v_aincs_py38 lightgbm
```

다음은 pydotplus, pydot, scikit-image, patsy, statsmodels, opencv 등 관련 라이브러리 설치 방법은 다음과 같다.

```
라이브러리 설치 명령어(C:\ncs_prj\v_aincs_py38) ← 생략
(v_aincs_py38) conda install -n v_aincs_py38 pydotplus
(v_aincs_py38) conda install -n v_aincs_py38 pydot
(v_aincs_py38) conda install -n v_aincs_py38 scikit-image
(v_aincs_py38) conda install -n v_aincs_py38 patsy
(v_aincs_py38) conda install -n v_aincs_py38 statsmodels
(v_aincs_py38) conda install -n v_aincs_py38 opencv
※ “conda install -n v_aincs_py38 패키지명”으로 설치되지 않고 오류가 발생할 경우
anaconda 공식 사이트를 통해 패키지 설치 방법을 확인한 후 방법에 따라 설치한다.
```



출처: 집필진 제작(2021)  
[그림 부록-12] 패키지 설치 방법 확인 방법

## 관련 라이브러리 참고

### <머신러닝/딥러닝에서 자주쓰이는 라이브러리>

넘파이(NumPy): 다차원 배열을 쉽게 처리하고 효율적으로 사용하도록 지원하는 라이브러리

사이파이(SciPy): 과학 기술 계산을 위한 파이썬(python) 라이브러리(Numpy, Matplotlib, pandas, Sympy 등 라이브러리와 함께 동작)

판다스(Pandas): 직관적인 관계형 또는 분류된 데이터로 작업할 수 있도록 설계된 빠르고 유연하며 다양한 데이터 구조를 제공하는 라이브러리

사이킷-런(Scikit-learn): 머신러닝 라이브러리 중의 하나로 분류, 회귀, 군집화, 의사 결정 트리 등 머신러닝 알고리즘을 적용할 수 있는 함수들 제공

맷플롯립(matplotlib): 다양한 데이터를 많은 방법으로 도식화할 수 있도록 하는 라이브러리

seaborn: 맷플롯립(matplotlib)을 기반으로 다양한 색상 테마와 통계용 차트 등의 기능을 추가한 시각화 라이브러리

spyder: 데이터 탐색, 대화식 실행, 심층 검사 및 과학 패키지의 시각화 라이브러리

h5py: 파이썬(python)에서 HDF5 바이너리 데이터 포맷을 사용하기 위한 인터페이스 라이브러리

pillow: 이미지 썸네일 생성, 다수 이미지 합성/포맷 변환/회전하기 등 라이브러리

tqdm: progress bar 생성, 함수나 반복문의 TTC(Time To Completion) 예측 관련 라이브러리

keras: 인공신경망을 쉽게 접근하도록 제공하는 라이브러리로 텐서플로 위에서 동작

### <부스팅 관련 라이브러리>

부스팅: 머신러닝 앙상블 기법 중의 하나로 무작위로 규칙을 선택하는 것보다 가능성이 높은 규칙(분류기[classifier] 또는 약한 학습기[weak learner]들을 순차적으로 여러개 결합하여 예측 또는 분류 성능을 높이는 접근 방법

xgboost: Gradient Boosting 알고리즘을 분산 환경에서도 실행할 수 있도록 구현해 놓은 라이브러리(Gradient Boosting - 회귀 및 분류 문제를 위한 기계 학습 기술)

catboost: 범주형(categorical) 변수를 처리하는 데 유용한 알고리즘

lightgbm: xgboost의 효율성 문제를 보완하여 나온 알고리즘

### <기타 라이브러리>

pydotplus: DecisionTreeClassifier 파이썬 코드에서 제공하는 DOT 파일과 GraphViz를 인터페이스하는 라이브러리

pydot: dot 스크립트 언어를 파싱하기 위한 라이브러리

scikit-image: 이미지처리에 특화된 파이썬(python) 이미지 라이브러리

patsy: 데이터 선택, 변환, 추가, 스케일링 관련 라이브러리

statsmodels: 파이썬 통계 분석 라이브러리(ARIMA, VAR 등)

opencv: 실시간 컴퓨터 비전 목적의 라이브러리

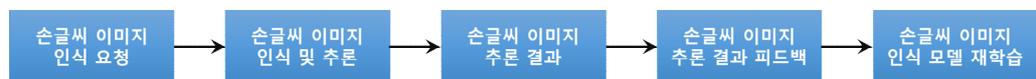
## 학습 1-2. 비즈니스 이해 및 로직 구현

### ① 인공지능서비스 로직을 구현한다.

1. 설계서를 확인한다.
2. 설계 내용에 따라 로직을 구현한다.

#### (1) 서비스 요청 기능을 구현한다.

인공지능서비스의 비즈니스 로직은 [그림 부록-13]과 같은 흐름이 있는 것으로 가정하고 로직을 구현한다.



출처: 집필진 제작(2021)

[그림 부록-13] 인공지능서비스 비즈니스 구성 예시

#### (가) 손 글씨 이미지 관련 정보 저장을 위한 모델(model.py)을 구현한다.

인공지능서비스의 첫 번째 이벤트로 손 글씨 이미지를 업로드하고 관련 정보를 데이터베이스 저장하는 것이다. 따라서 손 글씨 이미지 업로드 관련 정보 저장을 위한 데이터베이스 모델을 구현한다.

```
c:\ncsprj\aisvc\aisvcmodel\model.py
```

```
from django.utils import timezone
from django.db import models

class InfilImages(models.Model):
    image = models.ImageField(default='media/default_image.png')
    result = models.TextField()
    answer = models.CharField(max_length=200)
    created_date = models.DateTimeField(default=timezone.now)
    def __str__(self):
        return self.answer
```

#### (나) 데이터베이스 테이블을 생성한다.

Django model 코드를 이용하여 데이터베이스 테이블을 생성하는 코드를 생성하고 이를 이용하여 데이터베이스 테이블을 생성한다.

#### 데이터베이스 테이블 생성

```
(v_aincs_py38) C:\ncsprj\aisvc>python manage.py makemigrations aisvcmodel
```

```
(v_aincs_py38) C:\ncsprj\aisvc>python manage.py migrate aisvcmodel
```

모델에 정의된 항목을 기준으로 데이터베이스에 테이블이 생성된다.

(다) 이미지 파일 업로드를 위한 공통 설정을 변경한다.

```
c:\ncsprj\aisvc\cfg\settings.py
...
MEDIA_URL = '/images/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'images')
```

settings.py 파일 내용의 하단에 이미지 경로에 대한 설정을 추가한다.

(라) 공통 URL 정보를 변경한다.

손 글씨 이미지 인식 및 추론결과 제공 서비스를 위한 aisvcmodel 애플리케이션의 경로를 추가하고 이미지에 대한 정적자원 처리를 위해 static 경로를 추가한다.

```
c:\ncsprj\aisvc\cfg\urls.py
from django.urls import path
from django.contrib import admin
from django.conf import settings
from django.urls import include
from django.conf.urls.static import static
urlpatterns = [
    path('admin/', admin.site.urls),
    path('aisvc/', include('aisvcmodel.urls')),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

(마) 인공지능서비스 애플리케이션(aisvcmodel) URL 정보를 변경한다.

aisvcmodel 애플리케이션의 urls.py(python startapp aisvcmodel 명령어로 자동 생성되는 파일, 자동 생성이 되지 않았다면 수동을 생성해도 된다.)에 손 글씨 이미지 업로드를 위한 URL정보를 추가한다(인터넷 접근은 http://~/~/aisvc/inflnput).

```
c:\ncsprj\aisvc\aisvcmodel\urls.py
from django.urls import path
from . import views
urlpatterns = [
    path('inflnput', views.inflnput, name='inflnput'),
]
```

(바) 인공지능서비스 이미지 업로드 화면 구성을 위한 form을 구현한다.

html에서 모델에 따라 폼을 구성하는 설정 사항으로 손 글씨 이미지 업로드 form 구성을 위해 다음과 같이 구현한다(예: <form method="POST" action="").

```
c:\ncsprj\aisvc\aisvcmodel\forms.py
```

```
from django import forms
from .models import InflImages
class InflImagesForm(forms.ModelForm):
    class Meta:
        model = InflImages
        fields = ('image',)
```

(사) 인공지능서비스 손 글씨 이미지 업로드 화면을 구현한다.

손 글씨 이미지 인식 및 추론을 위해 손 글씨 이미지를 업로드하는 화면의 html 소스 코드를 구현한다.

```
c:\ncsprj\aisvc\aisvcmodel\templates\aisvc\inflInput.html
```

```
<!DOCTYPE html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>손 글씨 이미지 알아맞추기</title>
</head>
<style>
    .my_table {border-spacing:0px; border:1px solid #000;}
    .my_table td {padding:0px;}</style>
</style>
<body>
    <h1>손 글씨 인식을 위한 이미지 업로드</h1>
    <div>
        <div>
            <div>
                <form method="POST" enctype="multipart/form-data">
                    <table width="100%" class="my_table">
                        <tr class="my_table">
                            <td width="70%" class="my_table" align="center">
                                {% csrf_token %}{% form.as_p %}</div></td>
                            <td width="30%" class="my_table" align="center">
                                <button type="submit" class="save btn btn-default">Save</button>
                            </td>
                        </tr>
                        <tr class="my_table">
                            <td class="my_table" align="center" colspan="2">
```

```

        <a href="/aisvc/infList">손 글씨 이미지 인식 결과 보기</a></td>
    </tr>
</table>
</form>
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```

- (아) 인공지능서비스 손 글씨 이미지 업로드 화면을 띄우기 위한 연결 정보를 구현한다. “http://로컬주소/aisvc/infInput” 주소로 접속했을 때 손 글씨 이미지 업로드 화면으로 이동하도록 구현한다. 또한 손 글씨 이미지 업로드 화면의 <form> 구성 항목도 지정해 준다.

```

c:\ncsprj\aisvc\aisvcmodel\views.py

from django.shortcuts import render, redirect
from .forms import InImagesForm

def infInput(request):
    form = InImagesForm()
    return render(request, 'aisvc/infInput.html', {'form': form})

```

- (2) 손 글씨 이미지 인식 결과를 제공하는 처리기를 구현한다.

업로드할 이미지를 선택하고 저장하게 되면 서버에서 이미지를 전송받고 학습된 모델을 이용하여 이미지에 포함된 숫자를 판별하여 저장하는 처리기를 구현한다.

- (가) 손 글씨 이미지 인식을 위한 공통 함수를 구현한다.

손 글씨 이미지 인식 기능을 구현하기 위해 입력값을 받아 인공지능 모델에 요청하고 결과값을 받아 처리하는 기능을 담당한다.

```

c:\ncsprj\aisvc\aisvcmodel\views.py(코드 추가)

import numpy as np
import tensorflow as tf

from tensorflow.keras.models import model_from_json
from tensorflow.keras import utils as np_utils
from tensorflow.keras.preprocessing.image import img_to_array

```

```

from PIL import Image
graph = tf.Graph()

def HandwritingRecognition():
    with graph.as_default():
        json_file = open('./model/mnist_dnn_models.json', 'r')
        loaded_model_json = json_file.read()
        json_file.close()
        loaded_model = model_from_json(loaded_model_json)
        loaded_model.load_weights("./model/mnist_dnn_model.h5")
        loaded_model.compile(optimizer='sgd',
                              loss='sparse_categorical_crossentropy', metrics=['accuracy'])
        result = loaded_model.predict(img)
        answer = loaded_model.predict_classes(img)
    return result, answer

```

소스 코드의 mnist\_dnn\_model.json과 mnist\_dnn\_model.h5을 생성하는 별도의 파이썬 코드는 다음 코드를 참고한다.

#### mnist\_dnn\_model.json과 mnist\_dnn\_model.h5 생성 코드

```

import tensorflow as tf
import numpy as np

# (1) 데이터 준비(Preparing the data)
(x_trains, y_trains), (x_tests, y_tests) = tf.keras.datasets.mnist.load_data()
#print('x_train's shape:', x_train.shape)
#print('y_train's shape:', y_train.shape)
#print('x_test's shape:', x_test.shape)
#print('y_test's shape:', y_test.shape)

import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (5, 5)
plt.imshow(x_train[0])
plt.show()

x_trains = x_trains.reshape((60000, 28 * 28)) / 255.0
x_tests = x_tests.reshape((10000, 28 * 28)) / 255.0

# (2) 모델 구축(Building the model)

```

```

# Sequential model
model = tf.keras.models.Sequential()
model.add(tf.keras.layers.Dense(units=128, activation='relu', input_shape=(28*28,)))
model.add(tf.keras.layers.Dense(units=10, activation='softmax'))
model.summary()
# 케라스 모델의 시각화
tf.keras.utils.plot_model(model, to_file='model_plot.png', show_shapes=True)

# (3) 모델 컴파일(Compiling the model)
model.compile(optimizer='sgd', loss='sparse_categorical_crossentropy',
              metrics=['accuracy'] )

# (4) 모델 학습(Training the model)
results = model.fit(x_trains, y_trains, epochs=100, verbose=1, validation_split=0.2)

# (5) 모델 평가(Evaluating the model)
print(results.history.keys())
# 정밀도 요약
import matplotlib.pyplot as plt
plt.plot(results.history['accuracy'])
plt.plot(results.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# 손실률 요약
plt.plot(results.history['loss'])
plt.plot(results.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# (6) 예측(Prediction for new data)

```

```

preds = model.predict(x_test, batch_size=128)
# 확률적으로 가장 높은 값
np.argmax(preds[0])
plt.imshow(x_test[0].reshape(28, 28))
plt.show()

# (7) 모델 저장, 불러오기(Saving and Loading the model)
model.save('mnist_dnn_model.h5')
m_json = model.to_json()
with open('mnist_dnn_model.json', 'w') as json_file:
    json_file.write(m_json)

from tensorflow.keras.models import model_from_json
json_file = open("mnist_dnn_model.json", "r")
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)
loaded_model.load_weights("mnist_dnn_model.h5")

n_model = loaded_model.load_weights('mnist_dnn_model.h5')
loaded_model.summary()
preds = loaded_model.predict(x_test, batch_size=128)

# 새로운 모델 생성 예시
new_model = tf.keras.models.load_model('mnist_dnn_model.h5')
new_model.summary()

```

(나) 업로드한 손 글씨 이미지의 전처리 함수를 구현한다.

업로드 이미지를 모델에 입력에 적합한 형태로 전처리하는 함수를 구현한다.

```

c:\ncsprj\aisvc\aisvcmodel\views.py(코드 추가)

# 이미지 전처리 함수
def preProcessImage(image, target):
    image = image.convert("L") # 흑백 변환
    image = image.resize(target) # 사이즈 변경
    image = img_to_array(image) # 숫자형 배열 변환
    image = np.expand_dims(image, axis=0) # 차원 추가

```

```

image = np.where(image >= 0 , 255-image, image)
image = image.reshape(1, 784).astype('float32') / 255.0
return image

```

(다) 손 글씨 이미지 인식 결과를 저장하는 기능을 구현한다.

(가)에서 구현한 공통 함수를 사용하여 손 글씨 이미지를 인식하고, 인식 결과를 데이터베이스 저장하는 코드를 구현한다.

**c:\ncsprj\aisvc\aisvcmodel\views.py(코드 추가)**

```

def inflInput(request):
    User.objects.all()
    if request.method == "POST":
        form = InImagesForm(request.POST, request.FILES)
        if form.is_valid():
            inf = form.save(commit=False)
            inf.author = request.user
            inf.save()

            img = Image.open(request.FILES['image'])
            img = preprocessImage(img, target=(28, 28))
            inf.result, inf.answer = HandwritingRecognition(img)
            inf.save()
            return redirect('infList')
        else:
            form = InImagesForm()
            return render(request, 'aisvc/inflInput.html', {'form': form})

```

(라) 손 글씨 이미지의 인식 결과 보기를 위한 URL 정보를 추가한다.

손 글씨 이미지의 인식 결과는 (다)의 코드에 따라 데이터베이스에 저장되었다. 데이터베이스에 저장된 인식 결과를 보여 주는 URL 정보를 추가하여 구현한다.

**c:\ncsprj\aisvc\aisvcmodel"urls.py(코드 추가)**

```

urlpatterns = [
    path('inflInput', views.inflInput, name='inflInput'),
    path('infList', views.infList, name='infList'),
]

```

(바) DB 쿼리 결과를 화면에 매핑하는 코드를 구현한다.

데이터베이스에는 이전 손 글씨 이미지 인식 결과가 저장되어 있다. 데이터베이스에 저장된 결과를 조회하는 장고 소스 코드로 'InflImages.objects.all()'을 구현하면 된다. 해당 소스 코드는 ORM 매커니즘에 따라 DB 쿼리를 만들고 DB 조회 결과를 lists라는 구조체에 보관하게 된다. 그리고 조회 결과를 render라는 함수를 통해 html 문서에 표현하도록 구현한다.

```
c:\ncsprj\aisvc\aisvcmodel\views.py(코드 추가)
```

```
def infList(request):  
    lists = InflImages.objects.all()  
    return render(request, 'aisvc/infList.html', {'lists':lists})
```

(사) 손 글씨 이미지의 인식 결과를 보여 주는 화면을 구현한다.

'InflImages.objects.all()' (쿼리 세트)를 이용하여 데이터베이스에 저장된 결과를 화면에 보여 주도록 구현한다. 손 글씨 이미지 인식 결과가 정확하지 않다면 화면의 링크를 통해 교정 의견을 제공하도록 링크를 구성한다.

```
c:\ncsprj\aisvc\aisvcmodel\templates\aisvc\infList.html
```

```
<!DOCTYPE html> <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
    <title>손 글씨 이미지 인식 결과 보기</title>  
</head>  
<style>  
    .my_table {border-spacing:0px; border:1px solid #000;}  
    .my_table td {padding:0px;}</style>  
</style>  
<body>  
    <h1>손 글씨 이미지 인식 결과 보기</h1>  
    <table width="100%" class="my_table">  
        {% for list in lists %}  
        <tr class="my_table">  
            <td width="30%" class="my_table" align="left">  
                인식대상 이미지:   
            </td>  
            <td width="70%" class="my_table" align="left">  
                인공지능 다변: 답변: {{ list.answer }}</td>
```

```

</tr>
<tr class="my_table">
  <td width="30%" class="my_table" align="left">숫자 정정 의견:</td>
  <td width="70%" class="my_table" align="left">
    {% for i in '0123456789'|make_list %}
      <a href="/aisvc/correct?pk={{list.pk}}&Correction={{i}}&{{i}} ">{{i}} </a>
    {% endfor %}
    → <a href="/aisvc/correctView?pk={{list.pk}}">교정 의견 보기</a></td>
</tr>
{% endfor %}
</table>
</body> </html>

```

(3) 손 글씨 이미지 인식 결과에 대한 교정 의견 저장 및 재학습 기능을 구현한다.

(가) 사용자 교정 의견 데이터베이스 모델을 구현한다.

사용자 교정 의견(Correct라는 테이블)을 저장하기 위한 데이터 모델을 구성한다. models.py에 코드를 구현하고 데이터 모델을 재생성해야 한다. 생성 스크립트 (python manage.py makemigrations)를 실행하고, 테이블을 생성한다(python manage.py migrate).

c:\ncsprj\aisvc\aisvcmodel\models.py(코드 수정)

```

...
class Correct(models.Model):
    InflImages = models.ForeignKey('aisvcmodel.InflImages',
        related_name='correct', on_delete=models.CASCADE)
    correct = models.CharField(max_length=1)
    author = models.ForeignKey('auth.User', on_delete=models.CASCADE)
    created_date = models.DateTimeField(default=timezone.now)

    def __str__(self):
        return self.correct

```

(바) 사용자 교정 의견을 처리하기 위해 URL 정보를 추가한다.

다음 그림에서 인식된 결과가 원본 손 글씨 이미지와 다를 경우 사용자는 1~9까지 링크를 클릭해 의견을 전달하는 기능이다. 해당 기능 제공을 위해 사용자 교정 의견을 처리하기 위한 url을 추가한다.

인식대상 이미지: <b>5</b>	인공지능 다변: 답변: [3]
숫자 정정 의견:	<b>0 1 2 3 4 5 6 7 8 9</b> → <a href="#">교정의견 보기</a>
인식대상 이미지: <b>3</b>	인공지능 다변: 답변: [3]
숫자 정정 의견:	<b>0 1 2 3 4 5 6 7 8 9</b> → <a href="#">교정의견 보기</a>

출처: 집필진 제작(2021)  
 [그림 부록-14] 인공지능서비스 교정 의견 링크 예시

```
c:\ncsprj\aisvc\aisvcmodel\urls.py(코드 수정)

urlpatterns = [
    path('inflInput', views.inflInput, name='inflInput'),
    path('inflList', views.inflList, name='inflList'),
    path('Correct', views.Correct, name='Correct'),
]
```

(바) 사용자의 교정 의견을 저장하는 기능을 구현한다.

사용자가 0~9까지 링크를 클릭한 사용자 교정 의견을 데이터베이스에 저장하는 기능을 구현한다.

```
c:\ncsprj\aisvc\aisvcmodel\views.py(코드 수정)

...
from .models import Correct
def Correct(request):
    corr = Correct()
    corr.InflImages = InflImages.objects.get(pk=request.GET.get('pk'))
    corr.correct = request.GET.get('correction')
    corr.author = request.user
    corr.save()
    return redirect('inflist')
```

(4) 다양한 교정 의견 중의 “선택 기능”과 재학습 기능을 구현한다.

(가) 다양한 교정 의견을 보기 위한 URL 정보를 추가한다.

사용자들이 제시한 다양한 교정 의견을 확인하기 위한 URL 정보를 추가한다.

```
c:\ncsprj\aisvc\aisvcmodel\urls.py

urlpatterns = [
    path('inflInput', views.inflInput, name='inflInput'),
```

```
path('infList', views.infList, name='infList'),
path('Correct', views.Correct, name='Correct'),
path('CorrectView', views.CorrectView, name='CorrectView'),
]
```

(나) 교정 의견 테이블 쿼리 결과를 교정 의견 보기 화면에 매핑하는 코드를 구현한다.

DB 쿼리를 위해 'Inflimages.objects.get(pk=request.GET.get('pk'))'을 사용하고, 조회 결과를 render라는 함수를 통해 html 문서에 표현하도록 구현한다.

```
c:\ncsprj\aisvc\aisvcmodel\views.py
def CorrectView(request):
    inf = Inflimages.objects.get(pk=request.GET.get('pk'))
    lists = Correct.objects.filter(inference=inf)
    return render(request, 'aisvcmodel/CorrectView.html', {'lists': lists})
```

(다) 사용자의 다양한 교정 의견을 확인할 수 있는 화면을 구현한다.

```
c:\ncsprj\aisvc\aisvcmodel\templates\aisvc\CorrectView.html
<!DOCTYPE html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>사용자 교정 의견</title></head>
<body>
<h1>교정 의견 보기</h1><p>인식 대상 이미지:

</p>
{% for list in lists %}
    <div><p>정정 숫자: {{ list.correct }}
    --
    <a href="/aisvcmodel/CorrectApply?pk={{list.pk}}&infPk={{list.Correct.pk}}">
    교정 적용(재학습 수행)
    </a></p></div>
{% endfor %}
</body>
</html>
```

(라) 사용자 교정 의견을 처리할 URL 정보를 추가한다.

```
c:\ncsprj\aisvc\aisvcmodel\urls.py

urlpatterns = [
    .... 중간 생략 ....
    path('correctApply', views.correctApply, name='correctApply'),
]
```

(마) 사용자 교정 의견 중의 “선택된” 이미지를 이용한 재학습 기능을 구현한다.

다양한 사용자 교정 의견 중에서 선택된 특정 이미지 이용하여 재학습을 수행하는 기능을 구현한다. json, h5에 대한 처리는 위의 코드를 참고하고, 재학습에 필요한 이미지는 쿼리를 사용한다. 또한 재학습은 이미지와 정답의 세트를 이용하므로 교정 의견을 정답으로 해서 재학습을 하도록 구현한다.

```
c:\ncsprj\aisvc\aisvcmodel\views.py

def correctApply(request):
    json_file = open('./model/mnist_dnn_model.json', 'r')
    loaded_model_json = json_file.read()
    json_file.close()
    loaded_model = model_from_json(loaded_model_json)
    loaded_model.load_weights('./model/mnist_dnn_model.h5')
    loaded_model.compile(optimizer='sgd',
        loss='categorical_crossentropy', metrics=['accuracy'])

    inf = InflImages.objects.get(pk=request.GET.get('infPk'))
    corr = Correct.objects.get(pk=request.GET.get('pk'))
    img = Image.open(inf.image)
    train_data = preprocessImage(img, target=(28, 28))
    train_label = [int(corr.correction)]
    train_label = np_utils.to_categorical(train_label, 10)
    loaded_model.fit(train_data, train_label, epochs=100, batch_size=200, verbose=0)
    return redirect('infList')
```

## 학습 1-3. 비즈니스 로직 문제 해결

③ 테스트를 실시하고, 결과를 정리한다.

1. 장고에서 단위 테스트 케이스를 구현한다.

단위 테스트를 위한 테스트 코드(aisvcmodel/tests.py) - 1개 사례

```
from django.test import TestCase
from django.test import Client
from PIL import Image
from .views import infInput
from aisvcmodel.models import InflImages
from aisvcmodel.forms import InflImagesForm

class infInput(TestCase):
    def test_infInput_First(self): # 손 글씨 이미지 업로드 테스트
        c = Client() # 브라우저 대시 요청을 보내고 받는 처리를 해 주는 객체
        res = c.get('/aisvc/infInput')
        self.assertEqual(res.status_code, 200)

    def test_infInput(self): # 손 글씨 이미지 업로드 테스트
        c = Client() # 브라우저 대시 요청을 보내고 받는 처리를 해 주는 객체
        c.post('/admin/login/?next=/admin/', {'username':'aisvcusr', 'password':'1234'})
        res = c.get('/aisvc/infInput')
        self.assertEqual(res.status_code, 200)
        img = Image.open("C:\\ncsprj\\test\\8번.jpg")
        file_dict = {'image':img}
        form =InflImagesForm(data={'author':'aisvcusr'}, files=file_dict)
        self.assertTrue(form.is_valid())

class infList(TestCase):
    def test_infList(self): # 손 글씨 이미지 인식 결과 테스트
        c = Client() # 브라우저 대시 요청을 보내고 받는 처리를 해 주는 객체
        res = c.get('/aisvc/infList')
        self.assertEqual(res.status_code, 200)

class correct(TestCase):
    def test_correct(self): # 손 글씨 이미지 인식 결과 테스트
```

```

p = InfiImages(image="media/default_image.png", answer=[3],
               result=[[2.6863703e-17, 2.5095933e-26, 1.9207342e-09, 9.8896283e-01,
                        1.2229788e-28, 7.5619610e-08, 3.2866371e-20, 2.0767821e-27,
                        1.1037086e-02, 5.5814597e-15]])
p.save(force_insert=True)

c = Client() # 브라우저 대시 요청을 보내고 받는 처리를 해 주는 객체
url = "/aisvc/correct?pk={}&Correction=2".format(p.id)
res = c.get(url)
self.assertEqual(res.status_code, 302) # infList 리다이렉트 코드인지 확인

class correctView(TestCase):
    def test_correctView(self):
        p = InfiImages(image="media/default_image.png", answer=[3],
                       result=[[2.6863703e-17, 2.5095933e-26, 1.9207342e-09, 9.8896283e-01,
                                1.2229788e-28, 7.5619610e-08, 3.2866371e-20, 2.0767821e-27,
                                1.1037086e-02, 5.5814597e-15]])
        p.save(force_insert=True)
        c = Client() # 브라우저 대시 요청을 보내고 받는 처리를 해 주는 객체
        res = c.get('/aisvc/correctView?pk={}'.format(p.id))
        self.assertEqual(res.status_code, 200)

```

## 2. 장고에서 구현한 단위 테스트 케이스를 실행한다.

〈표 부록-1〉 가장 많이 사용되는 assertion

메소드	검사 내용
assertEqual(a, b)	a == b
assertNotEqual(a, b)	a != b
assertIs(a, b)	a is b
assertIsInstance(a, b)	isinstance(a, b)
assertFalse(x)	bool(x) is False
assertIn(a, b)	a in b
assertIsNot(a, b)	a is not b
assertIsNone(x)	x is None
assertIsNotNone(x)	x is not None
assertTrue(x)	bool(x) is True
assertNotIn(a, b)	a not in b
assertNotIsInstance(a, b)	not isinstance(a, b)

〈표 부록-2〉 예외, 경고, 로그 메시지의 발생 assertion

메소드	검사 내용
assertRaises(exc, fun, *args, **kwds)	fun(*args, **kwds)가 exc를 발생
assertWarnsRegex(warn, r, fun, *args, **kwds)	fun(*args, **kwds)가 warn을 발생하고 메시지가 정규식 r에 일치
assertWarns(warn, fun, *args, **kwds)	fun(*args, **kwds)가 warn을 발생
assertRaisesRegex(exc, r, fun, *args, **kwds)	fun(*args, **kwds)가 exc를 발생하고 메시지가 정규식 r에 일치
assertLogs(logger, level)	with블록이 최소 level로 logger에 로그를 남김

〈표 부록-3〉 구체적인 사항 검사 assertion

메소드	검사 내용
assertAlmostEqual(a, b)	round(a-b, 7) == 0
assertGreater(a, b)	a > b
assertLessEqual(a, b)	a <= b
assertLess(a, b)	a < b
assertNotAlmostEqual(a, b)	round(a-b, 7) != 0
assertNotRegex(s, r)	not r.search(s)
assertGreaterEqual(a, b)	a >= b
assertRegex(s, r)	r.search(s)
assertCountEqual(a, b)	순서와 상관없이 a와 b가 같은 개수의 같은 요소를 가졌는지

〈표 부록-4〉 형(type) 비교 assertion

메소드	검사 내용
assertMultiLineEqual(a, b)	문자열
assertSequenceEqual(a, b)	시퀀스
assertListEqual(a, b)	리스트
assertTupleEqual(a, b)	튜플
assertSetEqual(a, b)	집합 또는 불변 집합
assertDictEqual(a, b)	딕셔너리

## 학습 3-2. 모델 연동

③ 인공지능서비스 모델 연동을 위한 전이 학습 코드를 구현한다.

1. 인공지능 모델의 전이 학습이 필요한지 파악한다.
2. 인공지능 모델의 전이 학습 코드를 구현한다.

### 인공지능 모델 연동을 위한 전이 학습 구현

```
import os, time, copy, torch, torchvision
import matplotlib.pyplot as plt
import torch.nn as nn
import torch.optim as optim
import numpy as np
from __future__ import print_function, division
from torch.optim import lr_scheduler
from torchvision import datasets, models, transforms

# 모델을 전이 학습하기 위한 함수
def model_retrain(model, criterions, optimizers, scheduler, num_epoch=25):
    d_since = time.time()

    bst_model_wts = copy.deepcopy(model.state_dict())
    bst_acc = 0.0

    for epoch in range(num_epoch):
        print('Epochs {}/{}'.format(epoch, num_epoch - 1))
        print('-' * 10)

        # 각 에폭(epoch)은 학습 단계 및 검증 단계로 구분한다.
        for phase in ['trains', 'vals']:
            if phase == 'trains':
                model.train() # 학습 모드로 모델을 설정
            else:
                model.eval() # 평가 모드로 모델을 설정

        running_loss = 0.0
        running_corrects = 0
```

```

# 데이터 반복
for input, label in dataloaders[phase]:
    input = input.to(device)
    label = label.to(device)

    # 매개 변수 경사도를 0으로 설정
    optimizer.zero_grad()

    # 순전파
    # 학습 시에만 연산 기록을 추적
    with torch.set_grad_enabled(phase == 'trains'):
        output = model(input)
        _, preds = torch.max(output, 1)
        loss = criterion(output, label)

        # 학습 단계인 경우 역전파+최적화
        if phase == 'trains':
            loss.backward()
            optimizer.step()

    # 통계
    running_loss += loss.item() * inputs.size(0)
    running_corrects += torch.sum(preds == label.data)
if phase == 'trains':
    scheduler.step()

epoch_loss = running_loss / dataset_sizes[phase]
epoch_acc = running_corrects.double() / dataset_sizes[phase]

print('{} Loss: {:.4f} Acc: {:.4f}'.format(phase, epoch_loss, epoch_acc))

# 모델의 깊은 복사(deep copy)
if phase == 'vals' and epoch_acc > bst_acc:
    bst_acc = epoch_acc
    bst_model_wts = copy.deepcopy(model.state_dict())

print()

```

```

time_elapsed = time.time() - d_since
print('학습 완료: {:.0f}m {:.0f}s'.format(time_elapsed // 60, time_elapsed % 60))
print('Best val Acc: {:.4f}'.format(bst_acc))

# 가장 나은 모델 가중값을 불러옴.
model.load_state_dict(bst_model_wts)
return model

# 모델 예측값 시각화 함수
def model_visualize(models, num_image=5):
    was_training = models.training
    models.eval()
    images_so_far = 0
    fig = plt.figure()

    with torch.no_grad():
        for i, (input, label) in enumerate(dataloaders['val']):
            input = input.to(device)
            label = label.to(device)

            output = models(input)
            _, preds = torch.max(output, 1)

            for j in range(input.size()[0]):
                images_so_far += 1
                ax = plt.subplot(num_image//2, 2, images_so_far)
                ax.axis('off')
                ax.set_title('predicted: {}'.format(class_names[preds[j]]))
                imgView(input.cpu().data[j])

            if images_so_far == num_image:
                models.train(mode=was_training)
                return
    models.train(mode=was_training)

```

```

# 이미지 시각화 함수
def imgView(inp, title=None):
    inp = inp.numpy().transpose((1, 2, 0))
    mean = np.array([0.485, 0.456, 0.406]) #ImageNet의 평균
    std = np.array([0.229, 0.224, 0.225]) #ImageNet의 표준 편차
    inp = std * inp + mean
    inp = np.clip(inp, 0, 1)
    plt.imshow(inp)
    if title is not None:
        plt.title(title)
    plt.pause(0.001) # 갱신이 될 때까지 대기

def run():
    torch.multiprocessing.freeze_support()
    print('loop')

if __name__ == '__main__':
    run()
    plt.ion() # 대화형 모드

    """
    한국용 차량 데이터는 https://aihub.or.kr/aihub-data/vision/about 의 "한국 차량
    이미지" 전체 데이터를 다운로드받았고, 대규모의 데이터라 일부를 테스트용으로
    분류하였다.

    데이터를 불러오기 위해 torchvision과 torch.utils.data 패키지를 사용한다.
    여기서 풀고자 하는 문제는 한국용 차량 중 그랜저, 소나타, 아슬란을 분류하는
    모델을 학습하는 것이다.

    그랜저, 소나타, 아슬란 각각의 학습용 이미지는 대략 140장(전체 420여 장) 정도
    있고, 약 100여 개의(전체 300여 개) 검증용 이미지가 있다.
    """

    # 학습을 위해 데이터 증가(augmentation) 및 일반화(normalization)
    # 데이터 검증을 위해 입력 데이터의 일반화
    data_transforms = {
        'train': transforms.Compose([
            transforms.RandomResizedCrop(224),
            transforms.RandomHorizontalFlip(),

```

```

        transforms.ToTensor(),
        transforms.Normalize([0.4850, 0.4560, 0.4060], [0.2290, 0.2240, 0.2250])
    ]),
    'val': transforms.Compose([
        transforms.Resize(256),
        transforms.CenterCrop(224),
        transforms.ToTensor(),
        transforms.Normalize([0.4850, 0.4560, 0.4060], [0.2290, 0.2240, 0.2250])
    ]),
}

data_dir = 'retrain/Kcar_data'
img_datasets = {x: datasets.ImageFolder(os.path.join(data_dir, x),
        data_transforms[x]) for x in ['trains', 'vals']}
dataloaders = {x: torch.utils.data.DataLoader(img_datasets[x], batch_size=4, shuffle=True,
        num_workers=4) for x in ['trains', 'vals']}
dataset_sizes = {x: len(img_datasets[x]) for x in ['trains', 'vals']}
class_names = img_datasets['trains'].classes

device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")

# 학습 데이터의 배치를 가져온다.
input, class = next(iter(dataloaders['trains']))
# 배치에서 격자 형태의 이미지를 생성한다.
out = torchvision.utils.make_grid(input)

imgView(out, title=[class_names[x] for x in class])

# 미리 학습한 모델을 불러온 후 마지막의 완전히 연결된 계층을 초기화한다.
model_ft = models.resnet18(pretrained=True)
num_fetures = model_ft.fc.in_features
# 각 출력(class) 샘플의 크기를 3개로 설정한다.
# nn.Linear(num_fetures, len (class_names))로 일반화할 수 있다.
model_ft.fc = nn.Linear(num_fetures, 3)
model_ft = model_ft.to(device)
celoss = nn.CrossEntropyLoss()

```

```

# 모든 매개 변수가 최적화되었는지 관찰
optimizer_ft = optim.SGD(model_ft.parameters(), lr=0.001, momentum=0.9)
# 7 에폭마다 0.1씩 학습률 감소
exp_lr_scheduler = lr_scheduler.StepLR(optimizer_ft, step_size=5, gamma=0.1)

# 학습 및 평가
model_ft = model_retrain(model_ft, celoss, optimizer_ft, exp_lr_scheduler, num_epoch=25)
model_visualize(model_ft)

"""
고정된 특징 추출기로서의 합성곱 신경망
마지막 계층을 제외한 신경망의 모든 부분을 고정한다.
requires_grad == False 로 설정: 매개 변수를 고정하여backward() 중에 경사도가
계산되지 않도록 한다.
"""

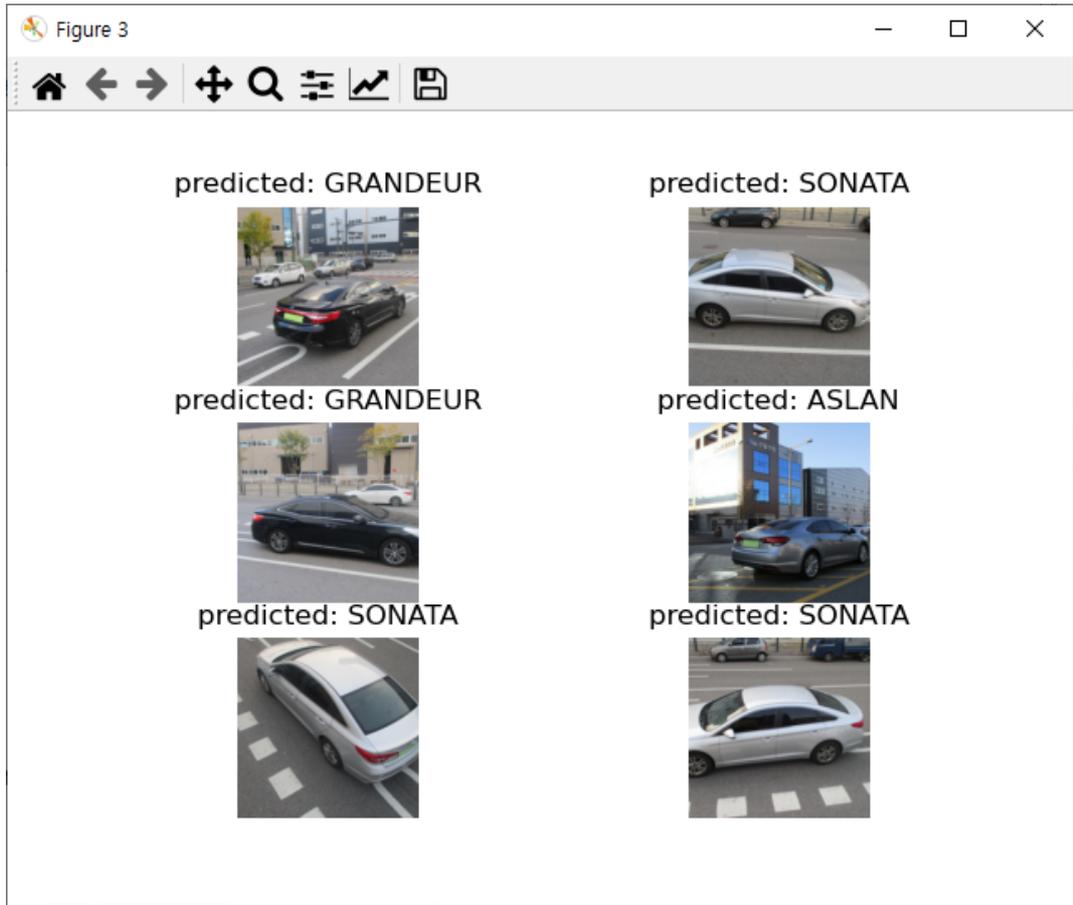
models_conv = torchvision.models.resnet18(pretrained=True)
for param in models_conv.parameters():
    param.requires_grad = False

# 새로 생성된 모듈의 매개 변수는 기본적으로requires_grad=True 임
num_fetures = models_conv.fc.in_features
models_conv.fc = nn.Linear(num_fetures, 3)
models_conv = models_conv.to(device)
celoss = nn.CrossEntropyLoss()

# 이전과는 다르게 마지막 계층의 매개 변수들만 최적화되는지 관찰
optimizer_conv = optim.SGD(models_conv.fc.parameters(), lr=0.001, momentum=0.9)
# 8 에폭마다 학습률 0.1씩 감소
exp_lr_scheduler = lr_scheduler.StepLR(optimizer_conv, step_size=8, gamma=0.1)
# 학습, 평가
models_conv = model_retrain(models_conv, celoss, optimizer_conv, exp_lr_scheduler,
num_epoch=25)
model_visualize(models_conv)

plt.ioff()
plt.show()
print("End")

```



출처: 집필진 제작(2021)  
[그림 부록-15] 검증용 데이터 세트에 대해 예측한 결과



## NCS학습모듈 개발이력

발행일	2021년 12월 31일	
세분류명	인공지능서비스구현(20010705)	
개발기관	한국정보통신기술사협회(개발책임자: 박철수), 한국직업능력연구원	
집필진	문재현(한국기술거래사회)*	김지영(LG CNS)
	김상복(㈜아이삭)	이규영(한국정보기술단)
	김선희(한국정보통신기술사협회)	장재경(KBS)
	김태형(한국정보통신기술사협회)	홍필두(한국폴리텍대학)
	윤상범(브리지텍)	검토진
	이돈희(SK(주))	
	정종민(산업통상자원부)	
	하광림(㈜씨에스리)	

\*표시는 대표집필자임

## 인공지능서비스 애플리케이션 개발(2001070504\_20v1)

저작권자	교육부
연구기관	한국직업능력연구원
발행일	2021. 12. 31.
ISBN	979-11-339-9512-7

※ 이 학습모듈은 자격기본법 시행령(제8조 국가직무능력표준의 활용)에 의거하여 개발하였으며, NCS통합포털사이트(<http://www.ncs.go.kr>)에서 다운로드 할 수 있습니다.



[www.ncs.go.kr](http://www.ncs.go.kr)