설계자동화 및 검토지능화를 위한 인공지능 활용 매뉴얼 2019.12.30 v.09

연세대학교 연구팀 건물정보모델링기반 설계자동화 및 검토지능화를 위한 인공지능과 실감미디어연계 설계기술 개발

2

연구원: 김진성, 송재열, 나성화, 이상훈, 조단규, 신은서, 최수형

연구책임자: 이진국

연세대학교 연구팀

목차

1.	목차	3
2.	요약	4
3.	건축/실내건축 인공지능 플랫폼 활용 가이드	5
3.1	가구 디자인 이미지 활용 가구 스타일 분류(Tensorflow)	5
3.2	가구 디자인 이미지 활용 가구 타입 분류(Google Colab)	12
3.3	데이터기반 전월세 추정 활용 매뉴얼(Microsoft Azure)	20
4.	건물정보모델링 기반 인공지능 플랫폼 활용 사례	25
4.1	BIM 건축객체 유형 추론	25
4.2	BIM 공간객체 용도 추론	28

요 약

본 매뉴얼은 건물정보모델링(Building Information Modeling, BIM)기반 설계자동화 및 검토지능화를 위한 인공지능과 실감미디어연계 설계기술 개발 과제의 1 차년도 중간 결과물 중 하나로써, 인공지능 기술을 활용이 건축/실내건축 분야에서 설계자동화 및 검토지능화를 위해 어떻게 활용될 수 있는지에 대한 내용 일부를 정리한 자료이다.

본 매뉴얼은 과제 연구 내용인 인공지능과 실감미디어를 연계한 설계자동화 및 검토지능화 요소개발 및 실제 수행에 이바지함을 목적으로 하며, 작게는 인공지능 플랫폼의 활용법과 건축/실내건축에서 이미지, BIM 데이터 등을 활용한 매뉴얼로 향후 활용될 수 있을 것이다. 이러한 목적에서 이하 내용은 다음 세 부분으로 작성되었다: 1) 가용 인공지능 플랫폼과 국내외 인공지능 설계 활용 사례, 2) 건축/실내건축분야의 인공지능 플랫폼 활용에 대한 매뉴얼, 3) 본 연구팀에서 수행된 BIM 기반 인공지능 플랫폼 활용 사례 소개

3 장의 건축/실내건축 인공지능 플랫폼 활용 가이드에서는 실내 디자인 이미지를 활용해 Tensorflow 와 Google Colab 를 기반으로 가구의 스타일 및 타입 분류 학습모델의 구축 및 활용을 위한 활용 매뉴얼에 대해 서술되었으며, 4 장에서는 건물정보모델링 기반 기존 진행되었던 인공지능 플랫폼 활용 사례에 대한 소개로 구성되어 있다.

4

3. 건축/실내건축 인공지능 플랫폼 활용 가이드

개요

실내디자인 이미지는 설계에 대한 직관적인 이해가 가능해 설계 과정에서 디자인 레퍼런스로 활 용되거나 설계에 대한 시각적 커뮤니케이션 수단으로 활용될 수 있는 중요한 자료이다. 이번 챕 터에서는 앞서 서술된 다양한 인공지능 플랫폼을 실내디자인에 실제 활용하는 예시에 대해 다루 고자 한다. 그 중 실내디자인 이미지기반 인공지능 활용방안을 주제로, 1) Google Colab 활용 인공 지능기반 가구 타입 분류 2) Tensorflow 활용 인공지능기반 가구의 스타일 분류 3) Microsoft Azure 활용 전월세 데이터 기반의 추론에 대해 다루고자 한다.

3.1 Google Colab를 활용한 인공지능기반 가구 타입 분류

1) Google Colab 개요

Google Colaboratory는 클라우드 기반 Jupyter 노트 형식으로 구성되어 Python 등 언어로 코드 작성 및 실행이 가능한 대화형 개발환경이다. Tensorflow, Keras, matplotlib 등 다양한 패키지들이 사전 설치되어 무설치로 활용할 수 있고 구글 드라이브 및 클라우드와 연동되어 컴퓨팅 파워의 활용이 가능한 특징이 있다. 사용을 위해서는 지메일 계정이 필요하다.





구글 코랩을 소개합니다, https://post.naver.com/viewer/postView.nhn?volumeNo=26447765&memberNo=38386150

기본 인터페이스는 그림 1과 같으며, Ctrl+F9를 통해 작성된 코드에 대한 전체 실행이 가능하고, Shift+Enter, 실행키를 활용해 라인별 실행도 가능하다.

CO [▲] UntitledO.ipynb ☆ 파일 수정 보기 삽입 런타임 도구 5	드음말 <u>저장중</u>		티 댓글 🚢 공유 👗 💠 📀
≔ ^{파일} 파일 탐색영역 ×	+ 코드 + 텍스트	코드 작성영역	✓ 다스크 📰 🗸 🖍 수정 가능 🔺
IE 파일 파일 탐색영역 × * 업료도 이 새로그림 · 오르드 이 새로그림 · 드라이브 다운트 · 그 · I sample_data	+ RE + RAE [1] print("Hello World") p hello World from socole colab isort drive drive.sount("/content/drive") []	코드 작성영역	✓ BAM + ✓ +87% / ▲ ★ ↓ ∞ □ ¢ ≡ :
[스크 79,64 08 사용 가능			

그림 1 Google Colab 기본 인터페이스

GPU/TPU 활용을 위해서는 런타임 -> 런타임 유형 변경 -> 하드웨어 가속기를 변경해야 한다.

제공되는 사양은 다음 명령어를 통해 확인 가능하다.

```
!cat /etc/issue.net
!head /proc/cpuinfo
!head -n 3 /proc/meminfo
!df -h
!nvidia-smi
```

C+	Ubuntu 18.04.3	LTS				+											
	processor	: 0				I NVID	IA-SML	440.44	1	Driver	Version:	418.	67	CLIDA	Versin	nn: 10.1	1
	vendor_id	: Genuinel	ntel					440.4	,	011101	10101011	4101		00011		1011	
	cpu family	: 6									+			-+			+
	nodel	: 63				L CDU	Mana		Dereie	topco-W	Ruo-Id		Dien A	L Vo	latila	Upcorr	ECC I
	nodel name	: Intel(R)	Xeon(P	CPU Ø	2.30GH:	I GFU	naire		Feisis	cence-m	Dus-Iu		DISP.A	1 10	latife	UNCOTT.	EUU I
	stenning	: 0	neonti	,	LIODORN	I Fan	Temp	Perf	Pwr:Hs	age/Capl		Memo	rv-llsage	I GPI	I–I I† i I	Comput	eM. I
	biorocodo	- 0-1				1 1 0.11	1 Omp		1 11 1 00	0.307 Oup		110110	.,	1 011	0	oompat	· :
	and WHa	- 2200 000				=====					-=======			+===			=====
	opu wnz	- 2000.000	,			1 0	Teels	Т4		Off	0000000	0.00.	0/ 0 Off	1			01
	cache size	- 46060 Kb)			1 0	resia	14		011	0000000	0.00.	04.0 011	1			01
	physical lo	- 0	-			I NZA	550	P8	10₩	/ 70₩	I OM	iB /	15079MiB		0%	Def	ault l
	Memiotai:	13335188 k	В					. =									
	MemFree:	10669052 k	в			+					+			-+			+
	MemAvailable:	12474136 k	B														
	Filesystem	Size Used	Åvail	Use% Mou	inted or												
	overlay	69G 32G	i 34G	49% /		+											+
	tmpfs	64M C) 64M	0% /de	i V	1 0											
	tmpfs	6.4G C	6.4G	0% /sy	s/fs/cs	Proc	esses.									ыри ме	mory
	/dev/sda1	75G 370	38G	50% /op	t/bin	I GPI	1	PID	Tuno	Process	nome :					Heana	1
	tmpfs	6.4G 12k	6.4G	1% /va	r/colat	i arc	,	110	Type	1100633	5 Houre					03496	
	sha	5.96 4.0k	5.96	1% /de	v/shn	=====											=====
	tmofe	6.46 0	6.46	0% /or	nc /acni	i 11-				- · ·							i
	tanto	6.40 0	6.40	0% /pr		I NO	runnin	g proce	esses t	ouna							1
	tupis	0.40 0	0.40	0% /91	00/505	+											+
	tmpis	0.46 U	0.40	UA /SY	S/T1FMN												
	Mon reb 3 04-2	4.23 2020															

그림 2 Colab 제공 사양 CPU/Memory 등(좌), 그래픽카드(우)

2) Google Colab 활용 이미지 기반 팔걸이 유뮤에 따른 의자타입 분류

학습 환경 설정

Google Drive Mount: 마운트는 저장 장치에 접근할 수 있는 경로를 디렉터리 구조에 편입시키는 작업을 의미하며, 구글 드라이브에 업로드된 데이터를 코랩에서 활용 가능하도록 한다.

▼ G	oogle Drive Mount
]] from google.colab import drive drive.mount('/content/drive')
e	Mounted at /content/drive

그림 3구글 드라이브 마운트

구글 드라이브에 ImageNet.Zip 데이터셋을 업로드 후 다음 코드를 실행해 Contents/ImageNet 경로에 데이터셋을 불러온다, ImageNet 폴더가 없다면 해당 경로에 폴더를 생성한다. 데이터셋은 팔걸이 유무별 이미지들과 학습 모델, 테스트 이미지들로 구성되어 있다.

•	Dov	wnload the data set exported from HyperLabel
	[3]	import os.path import shutil
		DOWNLOAD_LOCATION = '/content/ImageNet/' DRIVE_DATASET_FILE = '/content/drive/My Drive/ImageNet.zip'
		shutil.copy(DRIVE_DATASET_FILE, DOWNLOAD_LOCATION)
		print('Successfully downloaded the dataset')
	₽	Successfully downloaded the dataset

그림 4 구글 드라이브로부터 데이터셋 다운로드

다음 코드를 실행해 불러온 데이터셋의 압축을 해제한다.

!unzip /content/ImageNet/ImageNet.zip -d ImageNet/

Unzip the Dataset

0	lunzip /conte	ent/ImageNet/ImageNet.zip -d ImageNet/
C⇒	inflating: inflating:	<pre>ImageNet/armrest/image/Non_armrest/char_3UU16.jpg ImageNet/armrest/image/Non_armrest/char_30017.jpg</pre>
	inflating: inflating: inflating: inflating: inflating: inflating:	ImageNet/armrest/image/Non_armrest/char_30018.jpg ImageNet/armrest/image/Non_armrest/char_30019.jpg ImageNet/armrest/image/Non_armrest/char_30020.jpg ImageNet/armrest/image/Non_armrest/char_30021.jpg ImageNet/armrest/image/Non_armrest/char_30022.jpg ImageNet/armrest/image/Non_armrest/char_30023.jpg

그림 5 데이터셋 압축 해제 코드 실행 결과

다음 코드를 실행해 Tensorflow 1.8 버전을 설치하여 import 해 학습을 위한 환경 설정을 마친다.

!pip install tensorflow==1.8.0
import tensorflow as tf

학습 수행

다음 코드를 실행해 저장된 이미지에 대한 학습을 수행한다. Armrest, Non_Armrest 폴더 하위의 각 이미지들에 대해 bottleneck txt 파일이 생성되고 1000 번의 학습이 실행된다.

```
!python /content/ImageNet/retrain.py --
bottleneck_dir=/content/bottlenecks --how_many_training_steps 1000 --
model_dir=/content/ImageNet/imagenet --
summaries_dir=/content/ImageNet/armrest/training_summaries/basic --
output_graph=seatcapacity.pb --output_labels=seatcapacity.txt --
image_dir=/content/ImageNet/armrest/image
```

O	!python /content/ImageNet/r	etrain.py	bottleneck_dir=/content/bottleneckshow_many_training_steps 1000model_dir=/content/lm
D	2020-02-24 06:02:12.514320:	Step 760:	Cross entropy = U.1521U5
L*	2020-02-24 06:02:12.580295:	Step 760:	Validation accuracy = 97.0% (N=100)
	2020-02-24 06:02:13.152186:	Step 770:	Train accuracy = 94.0%
	2020-02-24 06:02:13.152254:	Step 770:	Cross entropy = 0.177800
	2020-02-24 06:02:13.207531:	Step 770:	Validation accuracy = 92.0% (N=100)
	2020-02-24 06:02:13.801610:	Step 780:	Train accuracy = 97.0%
	2020-02-24 06:02:13.801678:	Step 780:	Cross entropy = 0.160213
	2020-02-24 06:02:13.859975:	Step 780:	Validation accuracy = 90.0% (N=100)
	2020-02-24 06:02:14.434039:	Step 790:	Train accuracy = 96.0%
	2020-02-24 06:02:14.434107:	Step 790:	Cross entropy = 0.145142
	2020-02-24 06:02:14.488535:	Step 790:	Validation accuracy = 91.0% (N=100)
	2020-02-24 06:02:15.077416:	Step 800:	Train accuracy = 91.0%
	2020-02-24 06:02:15.077489:	Step 800:	Cross entropy = 0.225384
	2020-02-24 06:02:15.133267:	Step 800:	Validation accuracy = 96.0% (N=100)
	2020-02-24 06:02:15.715446:	Step 810:	Train accuracy = 92.0%
	2020-02-24 06:02:15.715515:	Step 810:	Cross entropy = 0.206235

그림 6 이미지 학습 과정 출력 예시

학습모델 테스트

다음 코드를 통해 테스트 이미지를 확인할 수 있다

```
def display_image(file_path = '/content/ImageNet/inference_test_image/'
):
    import cv2
    import matplotlib.pyplot as plt
    import os.path
    if os.path.exists(file_path):
        img = cv2.imread(file_path)
        show_img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        plt.imshow(show_img)
    else:
        print('failed to open file')
```

테스트 이미지를 활용한 학습모델 테스트 결과는 다음 코드를 통해 확인할 수 있다. 그림 7 은 98.5%의 확률로 팔걸이가 있는 의자라는 결과가 나타났으며, 그림 8 은 92.3%의 확률로 팔걸이가 없는 의자라는 결과가 나타나 학습모델의 테스트 결과를 확인할 수 있다.

```
!python /content/ImageNet/label_image.py /content/ImageNet/inference_te
st_image/test_10.jpg
display_image('/content/ImageNet/inference_test_image/test_10.jpg')
```



그림 7 테스트 이미지를 활용한 학습모델 테스트 예시(Armrest)

!python /content/ImageNet/label_image.py /content/ImageNet/inference_te
st_image/test_04.jpg

display image('/content/ImageNet/inference test image/test 04.jpg')



그림 8 테스트 이미지를 활용한 학습모델 테스트 예시(Non Armrest)

3.2 Tensorflow 활용 실내디자인 이미지 활용 가구 스타일 분류

1) Tensorflow 개요

Tensorflow 는 Google 에 의해 개발된 오픈 소스 소프트웨어 라이브러리로 가장 많이 활용되는 딥러닝 프레임워크 중 하나이다. 주로 Python 환경에서 실행 가능하나, 자바스크립트, C++, Go, C# 등의 언어에서도 활용 가능하다. 또한, Tensorflow-lite 등 모바일일 환경에서 실행 가능한 언어도 지원한다. 현재 2.0 버전이 출시되어 활용 가능하다. Google, Airbnb, 카카오모빌리티, 네이버 쇼핑 등 다양한 기업에서 Tensorflow 를 활용한 사례를 확인할 수 있다. (관련 링크: https://www.tensorflow.org/about/case-studies)

2) Tensorflow 활용 실내디자인 이미지 활용 가구 스타일 분류

학습 환경 설정

아나콘다 설치

아나콘다 홈페이지의 Download에서 Windows OS에 맞는 파이썬 버전을 클릭해 다운로드한다.

설치 링크: https://www.anaconda.com/distribution/#windows

Products Why Anaconda? Solutions Resources Company Do
Download
🔹 macOS 👌 Linux
07 for Windows Installer
Python 2.7 version
Download

그림 9 아나콘다 설치 페이지

가상환경 설정

윈도우 빠른 검색을 활용해 Anaconmda Prompt 실행 후 다음 코드를 통해 가상환경을 생성한다.

conda create -n tf python=3.6 anaconda (conda create -n 가상환경 이름 python=버전명 anaconda) 설치를 묻는 메세지에 'y' 후 설치 진행해 가상환경 생성 후 다음 명령어를 통한 가상환경 진입

activate tf (activate 가상환경 이름)

설치 이후에는 Anaconda Prompt 에서 activate 명령어를 통해 가상환경 실행 가능(그림 10)

Anaconda Prompt	-	×
(base) C:₩lsers₩Designit RA9>_activate tf		Î
(+f) C:₩lsers₩Designit RAR>		

그림 10 아나콘다 가상 환경 활성화 결과

Tenworflow 설치

Tensorflow CPU version

다음 코드를 통해 Tensorflow 설치, 설치를 묻는 메세지에 'y' 입력

```
pip install --ignore-installed -upgrade
https://storage.googleapis.com/tensorflow/windows/cpu/tensorflow-
1.13.1-cp36-cp36m-win_amd64.whl
```

파이썬 및 Tensorflow버전에 따라 명령어는 달라질 수 있으며, 설치 확인을 다음 명령어를 통해 설치 확인이 가능하다.

Python

```
import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```

Tensorflow GPU version

Tensorflow 의 GPU 버전을 활용하면 CPU 만 사용하는 경우보다 학습 등에 사용되는 시간을 단축할 수 있다. Tensorlfow 는 NVIDIA 의 Geforce 계열의 그래픽 카드만을 지원하고 있다.

다음 링크에서 PC GPU가 Nvidia Cuda를 지원 여부 확인

https://developer.nvidia.com/cuda-gpus

CUDA Toolkit 설치

https://developer.nvidia.com/cuda-downloads

CuDNN 설치 후 압축해제 후 CUDA 설치 경로에 붙여넣기

https://developer.nvidia.com/cudnn

CUDA 설치 경로 : C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\(버전명)

CUDA toolkit 환경 변수 추가

시작 -> 시스템 환경 변수 편집 -> 환경변수 -> 시스템 변수 -> path에 CUDA가 추가되어 있는지 확인하고 없을 경우 다음 환경변수 추가 (그림 11)

환경 변수 편집		×
C:#Program Files#NVIDIA GPU Computing Toolkit#CUDA#v9.2#bin	<u>^</u>	새도 만들기(N)
C:#Program Files#NVIDIA GPU Computing Toolkit#CUDA#v9.2#libnvvp		
C:#Program Files#NVIDIA GPU Computing Toolkit#CUDA#v9.2#include		편십(E)
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.2\extras\CUPTI\libx64		
C:#tools#cuda#bin		찾아보기(B)
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin		
C:#Program Files#NVIDIA GPU Computing Toolkit#CUDA#v10.0#libnvvp		삭제(D)
C:#Program Files#NVIDIA GPU Computing Toolkit#CUDA#v10.0#extras#CUPTI#libx64		
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\include		
C:#Windows#system32		위로 이동(U)
C:#Windows		
C:#Windows#System32#Wbem		아래로 이동(O)
C:#Windows#System32#WindowsPowerShell#v1.0#		
C:#Windows#System32#OpenSSH#		
C:#Program Files#NVIDIA Corporation#NVIDIA NGX		텍스트 편집(T)
C:#Users#designIT_RA2#.dnx#bin		
C:#Program Files#Microsoft DNX#Dnvm#		
C:#Program Files#Microsoft SQL Server#120#Tools#Binn#		
C:#Program Files#Microsoft SQL Server#130#Tools#Binn#		
C:#Program Files#Microsoft VS Code#bin		
	~	
	확인	취소
		.:

그림 11 시스템 환경 변수 추가

다음 코드를 통해 Tenworflow GPU를 설치할 수 있다.

```
pip install --ignore-installed --upgrade
https://storage.googleapis.com/tensorflow/windows/gpu/tensorflow_gpu-
1.13.1-cp36-cp36m-win_amd64.whl
```

Tensorflow CPU버전 설치 확인 명령어를 통해 동일하게 설치 확인 가능

Tenworflow 추가 라이브러리 설치

Pip install 기능을 통해 각 프로젝트에 필요한 라이브러리를 설치할 수 있다. 본 활용 가이드에서 필요한 라이브러리는 다음과 같다.

- TensorFlow-Plot
- Matplotlib
- Biwrap
- Pandas

가상환경 진입 후 다음 명령어를 통해 각 라이브러리를 설치할 수 있다.

Pip install tensorflow-plot
Pip install matplotlib
Pip install biwrap
Pip install pandas

학습 방법

학습 및 분류를 위해 활용되는 파일의 구조는 다음과 같다(그림 12).

🔄 imagenet	_
	Chain data at (turining tasting)
📊 Training_set_chair	Chair dataset(training, testing)
📄 label_image_folder 실행코드.txt	
Iabel_image_folder.py	Testing code
📄 retrain 실행 코드.txt	
PC retrain_v3.py	Training code
📄 tesorboard 실행 코드.txt	_

그림 12 학습 및 분류에 활용될 파일의 구조

학습 데이터셋은 클래식, 모던 스타일의 의자 이미지로 구성되어 있다. (그림 13).



그림 13 학습 데이터셋의 예시

Anaconda Prompt의 가상환경 진입 후 다음 명령어를 통해 디렉토리의 위치를 변경한다(그림 14).

Cd 디렉토리명



그림 14 경로 변경 결과

다음 코드를 입력해 학습을 시작한다.

```
Python retrain_v3.py --bottleneck_dir=bottlenecks --
image_dir=training_set_chair --how_many_training_steps 1000 --
model_dir=imagenet --summaries_dir=training_summaries/chair --
output_graph=chair.pb --output_labels=chair.txt --learning_rate 0.01 -
-validation_percentage 10 --testing_percentage 10 --train_batch_size 20
```

학습이 시작되면 다음과 같은 bottleneck txt파일이 생성된다(그림 15).

Anaconda Prompt - Python retrain_v3.pybottleneck_dir=bottlenecksi Tensorflow_ch	air > bottlenecks > classic			
Creating bottleneck at bottlenecks#classic#classic_106 Creating bottleneck at bottlenecks#classic#classic_110 이름	^	정한 날짜	유형	크기
Creating bottleneck at bottlenecksWclassicWclassic_111 Creating bottleneck at bottlenecksWclassicWclassic 113	1.jpg.txt 20)19-10-14 오후 7:23	텍스트 문서	22KB
Creating bottleneck at bottlenecks#classic#classic_115 📄 classic_00	3.jpg.txt 20)19-10-14 오후 7:23	텍스트 문서	22KB
Creating bottleneck at bottlenecksWclassicWclassic_118 classic_00	4.jpg.txt 20)19-10-14 오후 7:23	텍스트 문서	22KB
Creating bottleneck at bottlenecks#classic#classic_121 Classic_00	5.jpg.txt 20)19-10-14 오후 7:23	텍스트 문서	22KB
Creating bottleneck at bottlenecksWclassicWclassic_122	6.jpg.txt 20)19-10-14 오후 7:23	텍스트 문서	22KB
Creating bottleneck at bottlenecks#classic#classic_121 Classic 00	7.jpg.bxt 20)19-10-14 오후 7:23	텍스트 문서	23KB
Creating bottleneck at bottlenecksWclassicWclassic_128 classic 00	8.jpg.bxt 20)19-10-14 오후 7:23	텍스트 문서	23KB
Creating bottleneck at bottlenecks#classic#classic_130 📄 classic 00	9.jpg.txt 20)19-10-14 오후 7:23	텍스트 문서	22KB
Creating bottleneck at bottlenecks\classic\classic\classic_131	0.jpg.txt 20)19-10-14 오후 7:23	텍스트 문서	22KB
Creating bottleneck at bottlenecks#classic#classic_134 in classic 01	1 ipg bit 20)19-10-14 오후 7:23	텍스트 무서	22KB
Creating bottleneck at bottlenecksWclassicWclassic_133	2 ing txt 20)19-10-14 오후 7:23	텍스트 문서	22KB
100 bottleneck files created.	3 ing txt 20)19-10-14 오후 7:23	텍스트 문서	22KB
Creating bottleneck at bottlenecksWclassicWclassic_13d classic 01	4 ing byt 20	19-10-14 오克 7:23	텐스트 무서	22KB
Creating bottleneck at bottlenecks#classic#classic_141	5 ing tyt 20	19-10-14 오克 7:23	텍스트 무서	22KB
Creating bottleneck at bottlenecksWclassicWclassic_142	6 ing byt 20	10-10-14 Q = 7:23	텍스트 무서	22KB
Creating bottleneck at bottlenecks#classic#classic_144	7 ing tyt 20	10-10-14 Q = 7:23	테스트 무서	22KB
Creating bottleneck at bottlenecks#classic#classic_148 Classic_01	Ring byt 20	10-10-14 9 7.23	티스트 무서	22/0
Creating bottleneck at bottlenecks#classic#classic_150	Ding bit	10 10 14 0 = 7:23	티스트 문서	23/0
	Sipgint 20	10 10 14 0 = 7.23	ㅋㅡㅡ 단지 테스트 무서	22KB
Classic_02	0.jpg.o.c 20	/13-10-14 조우 7.23	ㅋ 군시	2288

그림 15 학습을 위해 생성된 Bottleneck 텍스트 파일 예시

bottleneck txt 파일 생성이 끝나면 학습이 시작된다(그림 16)

Im 선택 Anaconda Prompt - Python retrain_v3.pybottleneck_dir=bottlenecksimage_dir=training_set_chairhow_many_training_steps 1000mod		×
Use tf.cast instead.		~
WARNING:tensorflow:From C:#Users#Designit RA3#Anaconda3#envs#keras#lib#site-packages#tensorflow#pvthon#ops#cc	nfusion	mat
rix.pv:194: to int32 (from tensorflow.pvthon.ops.math ops) is deprecated and will be removed in a future vers	ion.	
Instructions for updating:		
Use tf.cast instead.		
2019-10-14 19:25:21.061398: Step 10: Train accuracy = 95.0%		
2019-10-14 19:25:21.062394: Step 10: Cross entropy = 0.441146		
2019-10-14 19:25:21.101360: Step 10: Validation accuracy = 90.0% (N=20)		
2019-10-14 19:25:22.651078: Step 20: Train accuracy = 100.0%_		
2019-10-14 19:25:22.6510/8: Step 20: Cross entropy = 0.288217		
2019-10-14 19:25:22.690056: Step 20: Validation accuracy = 95.0% (N=20)		
2019-10-14 19:25:24,175200; Step 30: Train accuracy = 90,0%		
2019-10-14 19:25:24.175200: Step 30: Cross entropy = 0.328330		
2019-10-14 19:25:24.21/17/7: Step 30: Validation accuracy = 100.0% (N=20)		
2019-10-14 19-20-20.60/699- Step 40- Train accuracy = 90.0% 2010-14 10:25-25 C02000 - Step 40- Correstorment - 0.201070		
2019-10-14 19-20-20.60/699- Step 40- Cross entropy = 0.2016/6 2019-10-14 19-25-25 20-60/699- Step 40- Volset-in an annual 90-0% (N=00)		
2019-10-14 13:20:20.726077; Step 40; Validation accuracy - 30.08 (N=20)		
2019 10 14 13-23-27,200400, Step 30, Hall accuracy = 100.0%		
2019-10-14 10:25:27.200400 Step 50: Volideting accuracy = 80.0% (N=20)		
2019-10-14 10-25-28 848701: Stap 60: Train acturacy = 100.0%		
2019-10-14 19:25:28 846701: Step 60: Cross entropy = 0 162192		
2019-10-14 19:25:28 888677: Step 60: Valide in accuracy = 100 0% (N=20)		
2019-10-14 19:25:30 364825; Step 70: Train accuracy = 100 0%		
2019-10-14 19:25:30.364825: Step 70: Cross entropy = 0.138688		
2019-10-14 19:25:30.404804: Step 70: Validation accuracy = 100.0% (N=20)		
2019-10-14 19:25:31.891692: Step 80: Train accuracy = 95.0%		
2019-10-14 19:25:31.891692: Step 80: Cross entropy = 0.172354		
2019-10-14 19:25:31.932670: Step 80: Validation accuracy = 100.0% (N=20)		
2019-10-14 19:25:33.439800: Step 90: Train accuracy = 100.0%		\sim

그림 16 bottleneck 텍스트 파일을 활용한 학습 시작

학습이 완료되면 4가지 파일이 생성된다.

- bottleneck 폴더: 각 이미지로부터 학습을 위해 변환된 텍스트 파일
- chair.pb: 학습 결과로 생성된 학습 모델
- Chair.txt: 학습에 활용된 라벨(Classic, Modern)
- Training summaries: Tensorboard를 활용해 학습의 결과를 시각화 가능

학습모델 테스트

Anaconda Prompt에서 가상환경 진입 후 다음 명령어를 통해 테스트 시작한다.

python label_image_folder.py result_chair Testing_set_chair chair python 파일명 테스트 결과 생성 파일명 테스트 이미지 위치 학습모델명

테스트 결과는 Anaconda Prompt 창에서 수치화된 값으로 확인할 수 있으며, 엑셀 파일로 저장된다.(그림 17)

	Anaconda Prompt												-
	Target_img	name_1st	score_1st	name_2nd	score_2n		А	В	С	D	E	F	G
0	Sq_classic_002.jpg	classic	0.95902	modern	0.0409	1		Target imo	name 1st	score 1st	name 2n	dscore 2nd	
$\frac{1}{2}$	Sq_classic_U36.jpg	classic	0.98039	modern	0.0196	2	0	Sq_classic_	classic	0.95902	modern	0.04098	
3	Sq_classic_059.jpg	classic	0.99841	modern	0.0015	3	1	sq_ciassic_	classic	0.98039	modern	0.01961	
4	Sq_classic_071.jpg	classic	0.99437	modern	0.0056	4	2	Sq_classic_	classic	0.97992	modern	0.02008	
С 6	Sq_classic_ll2.jpg Sq_classic_124.ipg	classic	0.99862	modern	0.0013	5	3	Sq classic	classic	0.99841	modern	0.00159	
7	Sq_classic_133.jpg	classic	0.99799	modern	0.0020	6	4	Sa classic	classic	0.99437	modern	0.00563	
8	Sq_classic_140.jpg	classic	0.93322	modern	0.0667	7	5	So classic	classic	0.99862	modern	0.00138	
9 10	Sq_classic_147.jpg Sq_classic_192.ipg	classic	0.99268	modern	0.0008	8	6	So classic	classic	0 98079	modern	0.01921	
11	Sq_classic_196.jpg	classic	0.99785	modern	0.0021	a	7	Sa classic	classic	0 00700	modern	0.00201	
12	Sq_classic_220.jpg	classic	0.98726	modern	0.0127	10	, ,	Sq_classic	classic	0.02222	modern	0.06679	
14	Sq_classic_228.jpg	classic	0.84637	modern	0.1536	11	<u> </u>	Sq_classic_	classic	0.95522	modern	0.00078	
15	Sq_classic_233.jpg	classic	0.99566	modern	0.0043	11	9	sq_classic_	classic	0.99937	modern	0.00063	
16	Sq_classic_236.jpg	classic	0.89053	modern	0.1094	12	10	Sq_classic_	classic	0.99268	modern	0.00732	
18	Sq classic 252.ipg	classic	0.99012	modern	0.0098	13	11	Sq_classic_	classic	0.99785	modern	0.00215	
19	Sq_classic_264.jpg	classic	0.99960	modern	0.0004	14	12	Sq_classic_	classic	0.98726	modern	0.01274	
20	Sq_modern_005.jpg	modern	0.99787	classic	0.0021	15	13	Sq_classic_	classic	0.99954	modern	0.00046	
22	Sa_modern_052.jpg	modern	0.96747	classic	0.0325	16	14	Sq_classic_	classic	0.84637	modern	0.15363	
23	Sq_modern_055.jpg	modern	0.89994	classic	0.1000	17	15	Sq_classic_	classic	0.99566	modern	0.00434	
24	Sq_modern_090.jpg	modern	0.98719	classic	0.0128	18	16	Sq_classic_	classic	0.89053	modern	0.10947	
26	Sq_modern_117.jpg	modern	0.99938	classic	0.0006	19	17	Sq_classic	classic	0.88294	modern	0.11706	
27	Sq_modern_2_007.jpg	modern	0.99911	classic	0.0008	20	18	Sa classic	classic	0.99012	modern	0.00988	
28	Sq_modern_2_013.jpg	modern	0.99930	classic	0.0007	21	19	Sa classic	classic	0 99960	modern	0 00040	_
						22	20	Sa moder	modern	0 99787	classic	0.00213	
						23	21	ISa moder	modern	0 99648	classic	0.00352	
						24	22	Sq_moder	modern	0.06747	classic	0.02252	
						24	22	Sq_moder	modern	0.00004	classic	0.03235	
							23	Isu moder	modern	0.69994	Classic	0.10006	

그림 17 테스트 결과 및 저장된 엑셀 파일 예시

3.3 Microsoft Azure ML Studio를 활용한 인공지능기반 계약 면적 추론

1) Microsoft Azure ML Studio개요

Microsoft Azure은 클라우드기반 클라우드 컴퓨팅 플랫폼으로, Jupyter Notebook기반의 코드 중심 개발환경과 노드 기반의 비주얼 랭귀지 기반 개발 환경을 함께 지원한다.



2) 서대문구 집값 데이터셋

1	A	В	с	D	E	
1	Dong Designated by Law	Deposit	Monthly Rent	Year of Construction	Contract Area	
2	HongJedong	9,000	40	2018	12~14	
3	HongEundong	6,000	50	2018	15~17	
4	NamGaJuadong	500	48	2018	3~5	
5	NamGaJuadong	1,000	45	2018	3~5	
6	HongEundong	500	45	2018	3~5	
7	HongJedong	500	40	2018	3~5	
8	HongJedong	500	40	2018	3~5	
9	HongJedong	1,000	50	2018	3~5	
10	HongJedong	1,000	50	2018	3~5	
11	BukGaJuadong	500	50	2018	3~5	
12	NamGaJuadong	1,000	48	2018	3~5	
13	BukGaJuadong	500	45	2018	3~5	
14	HongEundong	500	45	2018	3~5	
15	HongEundong	500	45	2018	3~5	
16	NamGaJuadong	500	43	2018	3~5	
17	NamGaJuadong	2,000	41	2018	3~5	
18	NamGaJuadong	2,000	37	2018	3~5	
19	HongEundong	3,000	35	2018	3~5	
20	NamGaJuadong	3,000	30	2018	3~5	
21	NamGaJuadong	7,000	15	2018	3~5	
22	NamGaJuadong	7,000	10	2018	3~5	
on	g Designated by Law	Deposit	Monthly Re	ent Year of C	Construction	Contract Area
llm	hu.		able	ا ليا		dia

4. 건물정보모델링 기반 인공지능 플랫폼 활용 가이드

건물정보모델링(Building Information Modeling; 이하 BIM) 및 응용으로 법규 검토, 에너지 시뮬레 이션, 동선 분석 등 다양한 디자인 검토가 가능해졌다. 이번 챕터에서는 본 연구실에서 수행된 다 양한 인공지능기반 설계 활용 중 BIM과 인공지능의 활용을 주제로 하는 두 가지 예시인 1) BIM 기반 건축객체 유형 추론 2) BIM기반 공간객체 용도 추론에 대해 다루고자 한다.

4.1 BIM 건축객체 유형 추론

개요

건물정보모델링(Building Information Modeling; 이하 BIM) 및 응용으로 설계안에 대한 다양한 검 토 및 데이터 활용이 가능해졌다. 그러나, 이러한 검토 응용의 실질적 활용을 방해하는 요소가 발 견되고 있다. 이중에서도 BIM 인스턴스의 데이터 품질 문제는 요구되는 정보의 누락이나 오류로 인해 응용 자체를 불가능하게 할 수 있다. 본 연구의 목적은 BIM 건축객체 정보를 추론하고 그 정보를 상기 BIM 응용 등에 활용할 수 있는 방법을 제안하는 것이다 (Figure).



Figure 1 Pre-Processing of BIM Instance for Rule-checking System

연구 방법 및 범위

구체적인 방법으로는 건축 객체의 기하정보, 시각정보, 관계정보 등 다양한 정보를 학습하여 유 형와 하위 타입 등 부가 정보를 인식할 수 있는 딥러닝 기반의 모델을 학습시키고, 이를 실제 BIM 소프트웨어에 적용한다. 본 글에서는 기하정보 및 시각정보를 각각 학습의 대상으로 하여, 인식 정보는 객체의 유형과 하위 타입으로 한정한다. 기하정보 학습은 인공신경망 기반 분류 알 고리즘이 활용되며, 시각정보 학습은 합성곱신경망 기반 분류 알고리즘이 활용된다. (Figure 2)



Figure 2 A data-driven approach of building object recognition using machine learning

기하정보 학습 모델 구축 및 활용

본 글에서 다루는 기하정보란, 사물의 높이, 너비, 깊이, 체적, 면적 등을 가리킨다. 이러한 객체의 기하정보를 변수로 해당 객체가 어떤 유형에 속하는지 추론할 수 있는 모델이 개발된다. 개발 프 로세스는 1) 학습 데이터 준비, 2)기계학습 기반 분류 모델 학습 및 테스트로 진행된다. 학습 데이 터의 수집은 BIM 저작도구인 Revit과 Dynamo 확장 도구를 활용하여 진행되었다. 실제 건물에 대 한 BIM 모델에 존재하는 객체의 이름, 체적, 면적, 너비, 높이, 깊이 정보를 추출하였다. Dynamo 기반의 자동 추출 프로그램이 활용되었다. (Figure 3)



Figure 3 Automated extracting geometric data of BIM objects using Revit and Dynamo 학습에 활용된 객체는 슬래브, 벽, 기둥, 창문, 문 등으로 한정하였다. Dynamo를 통해 약 4000여 개의 객체에 대한 기하정보가 추출되었다. 추출된 객체의 기하정보 데이터는 CSV파일로 저장되었 다. CSV파일은 객체 유형 분류 모델의 학습데이터로 활용되었다. 모델 학습의 파이프라인은 그림4 와 같다. 추론하고자하는 데이터는 Name 칼럼이며, 입력값으로 주어지는 데이터는 상기 6가지 기하정보 데이터이다. 전체기본적으로 제공되는 Multiclass Logistic Regression, Neural Networks 등 을 학습 알고리즘이 활용되었다. 각 알고리즘간 결과는 정확도, 정밀도(Precision), 재현율(Recall) 등으로 정량적으로 계산되며, 각 Name별 검증 결과는 Confusion Matrix로 시각화되어 분석된다.



Figure 4 The pipeline of Training ML models on top of MS Azure Machine learning studio MS Azure는 그림 5와 같이 학습 모델의 결과를 Confusion matrix로 시각화하는 기능을 제공한다. 학습 결과 기하정보를 학습한 모델의 정확도는



A) Multiclass logistic regression

B) Multiclass Neural Network

Figure 5 Comparsion of results training of Logistic regression and neural network using confusion matrix

시각정보 학습 모델 구축 및 활용

본 개발에서 다루는 시각정보란 3D 객체의 2D 이미지 데이터를 가리킨다. 일반적으로 png, jpg 등 포맷으로 존재한다. 대다수의 BIM 도구는 렌더링 기능을 제공함에 따라 2D 이미지로 저장될 수 있다. 객체를 여러각도에서 캡쳐한 이미지 데이터가 활용된다. 일반적으로 이미지 학습에 활용 되는 합성곱신경망(CNN)을 활용하여 각 이미지 데이터에 대한 전처리가 진행되었다. 전처리에는 Tensorflow와 Inception-v3모델이 활용된다. 전처리는 이미지에서 중요한 pixcel 데이터 값을 추출 하는 과정을 가리킨다. 전처리 결과 각 이미지 데이터는 Name 칼럼과 2048개의 float 특징데이 터로 표현된다. 해당 데이터는 CSV로 저장되고 MS Azure mahchine learning studio에서 모델 학습 에 활용된다. (Figure 5)



Figure 6 The process of generating and pre-processing BIM object visual data and BIM

MS Azure를 활용한 기계학습 모델링은 그림 6과 같다. 기본적으로 제공되는 Multiclass Logistic Regression, Neural Networks 등을 학습 알고리즘이 활용되었다.



Figure 7 The pipeline of Training ML models on top of MS Azure Machine learning studio 결과 분석

다음과 같은 하이퍼 파라미터를 설정하였을 때 가장 높은 정확도를 보였다. Hidden node 개수 : 50 학습 횟수 : 500 Learning rate : 0.1 Normalizer : Min-Max. 하이퍼 파라미터 변경에 따른 예측정 확도 변동 경향은 아래와 같이 정리 될 수 있다. Hidden node 개수는 일정 이상 증가하면 (200개) 정확도가 오히려 감소하는 경향을 보였다. 학습 횟수는 증가할수록 높은 정확도를 보였다 Learning rate 0.0001이하일 경우 오히려 정확도가 낮아지는 결과를 보였다. Normalizer의 경우에는 Binning 를 적용한 경우 약(50%)를 제외하고 약 95%에 가까운 정확도를 보였다.

4.2. BIM 공간객체 용도 추론

개요

BIM 공간객체 속성정보를 활용한 공간 용도 분류

법규상의 표준화된 용도분류와 실제 BIM 공간객체에 입력된 공간용도의 차이로 발생하는 오류를 줄이기 위한 공간 용도 매핑 자동화

건축물 설계품질검토 과정의 효율성 향상을 위해 BIM을 활용한 검토 방안에 관한 연구개발이 여러 국가에서 꾸준히 수행되고 있으며, 이를 통해 설계품질 검토 과정의 효율성 향상을 추구하 고 있다. BIM 모델에서 제공하는 각 객체의 속성정보를 활용하여 보다 정확하고 신속한 설계검토 방안이 개발되고 있으며, 특히 BIM의 공간객체를 활용하여 공간을 중심으로 한 설계 요구사항 검 토가 가능하게 되었다. 법규검토 또는 설계요구사항 검토과정에서 유사한 크기, 형태의 공간이라 도 그 공간의 용도에 따라 적용되는 요구사항은 상이하기 때문에 용도에 따른 검토가 필요하다. 이를 위해선 BIM 모델의 공간객체와 법규 내에서 정의된 특정 공간 용도가 일치하는지 검토하는 과정이 필요하며, 동시에 공간에 적용되어야 하는 검토 항목을 검색 및 분류하는 작업이 필요하 다. 하지만, BIM 모델 저작과정에서 공간의 용도는 BIM 모델을 제작하는 회사, 개인에 따라 상이 하게 나타날 수 있으며, 저작과정에서 오류나 누락이 생길 수 있다. 이는 BIM 기반 검토를 수행 함에 있어 정확한 검토를 방해하는 요인으로 작용한다. 본 사례는 BIM의 공간객체를 표준화된 체 계로 분류하는 학습모델을 설계품질검토과정에 활용하는 방안을 제안하며, 그 적용 사례 중 하나 로 공간 용도별 요구사항 검색 체계 구현방안에 대하여 서술한다.



Requirement Searching system

연구 방법 및 범위:

본 학습모델은 BIM 공간객체의 기본적인 속성정보가 입력값으로 주어지면 공간의 용도를 출력하는 학습모델 개발 방안에 대해 서술하며, 학습모델의 구현은 MS Azure Machine learning studio를 활용하여 진행하였다. 학습을 입력값으로 제공되는 속성정보의 다양한 조합에 따라 학습의 성능 을 비교하는 시험을 통해 최적의 분류모델을 도출하고자 하였다. 학습 모델은 Azure에서 제공하는 Multiclass neural network 모듈을 사용하여 구성하였으며, 모델에 사용된 hyper parameter는 아 래와 같다.

- Number of hidden nodes : 100
- Learning rate : 0.01
- Number of learning iteration: 300
- The type of normalizer: Min-Max normalizer

Space Usage Classification model



입력값으로 활용되는 공간의 속성정보를 선택하기 앞서 아래 그림에 나타나는 바와 같이 공간객 체의 속성정보를 분류하는 작업을 선행하였다. 공간객체의 속성정보는 크게 일반속성, 형태속성, 관계속성, 계산속성 4가지로 분류할 수 있으며, 본 연구에서 활용한 속성은 공간객체의 이름, 면 적, 높이, 문의 개수 4가지이며, 이를 조합한 속성정보 집합(Feature set)을 학습에 활용하였다. 공 간객체의 이름의 경우 텍스트의 형태를 갖고있으나 학습을 위해서는 벡터 행렬로 변환되어야 한 다. 텍스트를 수치적으로 표현하는 여러 방식 중 본 학습모델에는 인공 신경망 기반의 워드임베 딩 방식을 활용하였으며, 워드임베딩 모델에서 가장 대표적인 word2vec 모델을 활용하였다. word2vec 모델은 건축법규 문장을 학습하여 각 단어의 의미적 특성을 학습하였고, 실제 구현은 python 라이브러리인 gensim 라이브러리에 구현된 word2vec 클래스를 통해 구현하였다.



Space Usage Classification model

학습 데이터

학습에 활용된 데이터는 학교 건물을 대상으로 진행하며, 데이터는 한국교육개발원 교육시설환경 연구센터 (EDUMAC)에서 제공하는 우수교육시설 사례의 평면도와 Space program을 기반으로 제 작된 BIM 모델을 활용하였다. 초중고 5개 건물에서 수집된 404개의 공간정보를 학습데이터로 구 성하였으며, 학습과정에서 학습데이터의 80%는 학습용 데이터(training data), 20%는 검증용 데이 터(validation data)로 사용되었다.

Space Usage Classification model

- collected from the best practice examples of educational facilities in Korea, Education Facilities Research & Management Center (EDUMAC).
- Total 404 Spaces



	LongName	Name	Storey	Area	Height	N_Doors	Class
	시청각실(대강의실)	1 [시청각실(대강의실)]	Level 1	123.24	3500	2	9
	영양사실	10 [영양사실]	Level 1	11.98545	3500	0	26
	탈의/휴게실	11 [탈의/휴게실]	Level 1	12.08505	3500	0	26
	종합교실(1-1)	12 [종합교실(1-1)]	Level 1	61.836	3500	0	C
	종합교실(1-2)	13 [종합교실(1-2)]	Level 1	63.1	3500	0	C
	종합교실(1-3)	14 [종합교실(1-3)]	Level 1	63.11	3500	0	C
	종합교실(1-4)	15 [종합교실(1-4)]	Level 1	63.094	3500	0	C
	소강의실	16 [소강의실]	Level 1	38.4	3500	1	C
	중강의실	17 [중강의실]	Level 1	92.04	3500	2	C
	기술/가정실습실	18 [기술/가정실습실]	Level 1	91.104	3500	2	21
	체육실(국악실)	19 [체육실(국악실)]	Level 1	61.62	3500	4	8
	전시홀	2 [전시홀]	Level 1	191.4681	3500	9	10
4	홑	20 [흘]	Level 1	195.1164	3500	13	13
	다목적공간	21 [다목적공간]	Level 1	122.3557	3500	5	10
	교사연구실	22 [교사연구실]	Level 1	29.5554	3500	1	2
	학생휴게실	23 [학생휴게실]	Level 1	24.552	3500	1	24
	준비실	24 [준비실]	Level 1	12.098	3500	2	25
	보건실	3 [보건실]	Level 1	51.06153	3500	1	20
	숙직실	4 [숙직실]	Level 1	14.7917	3500	1	16
	식당	5 [식당]	Level 1	370.8964	3500	3	11
	보일러실	6 [보일러실]	Level 1	8.7	3500	1	27
	부식창고	7 [부식창고]	Level 1	11.89	3500	1	26
24	전처리실	8 [전처리실]	Level 1	25.05146	3500	3	26
	조리실	9 [조리실]	Level 1	146.7591	3500	2	26
	종합교실(2-1)	25 [종합교실(2-1)]	Level 2	61.8305	3200	0	C

학습 결과

속성정보 조합에 따른 학습 정확도가 다르게 나타난 것을 확인할 수 있었고, 4개의 속성정보 (공 간의 이름, 면적, 층고, 문의 개수)를 모두 사용한 학습모델의 성능이 93.91%로 가장 높게 측정되 었다. 하지만 아래 표에서도 볼 수 있듯 공간객체의 이름이 사용된 경우 나머지 속성여부의 조합 과는 관계없이 일정수준 이상의 정확도를 나타내는 것을 확인할 수 있다. 이는 공간의 이름이 해 당 공간의 용도를 드러내는 명확한 속성인 점과 공간의 형태적인 속성의 경우 현대 건물에서는 한 건물 내에 배치되는 공간은 용도에 관계없이 거의 유사하기 때문에 용도분류에는 적합하지 않 은 점을 나타내는 것으로 볼 수 있다.

Space Usage Classification model

Training Results

Input feature set	Overall Accuracy
All (Name + Area + Height + Number of doors)	0.939103
Name + Area + Height	0.935897
Name + Number of doors	0.934295
Name	0.932691
Area + Height + Number of doors	0.419753
Area + Height	0.382716



Space Usage Classification model



0.419753

0.949544 0.419753

NaN 0.419753

NaN

