



知齒知己 튼튼 하나, 튼튼 한 이

디지털 덴티스트리 부산 데이터톤 대회

킹공지능

팀장 설민규

팀원 신동우 안지민 하서영

INDEX

01 시장분석

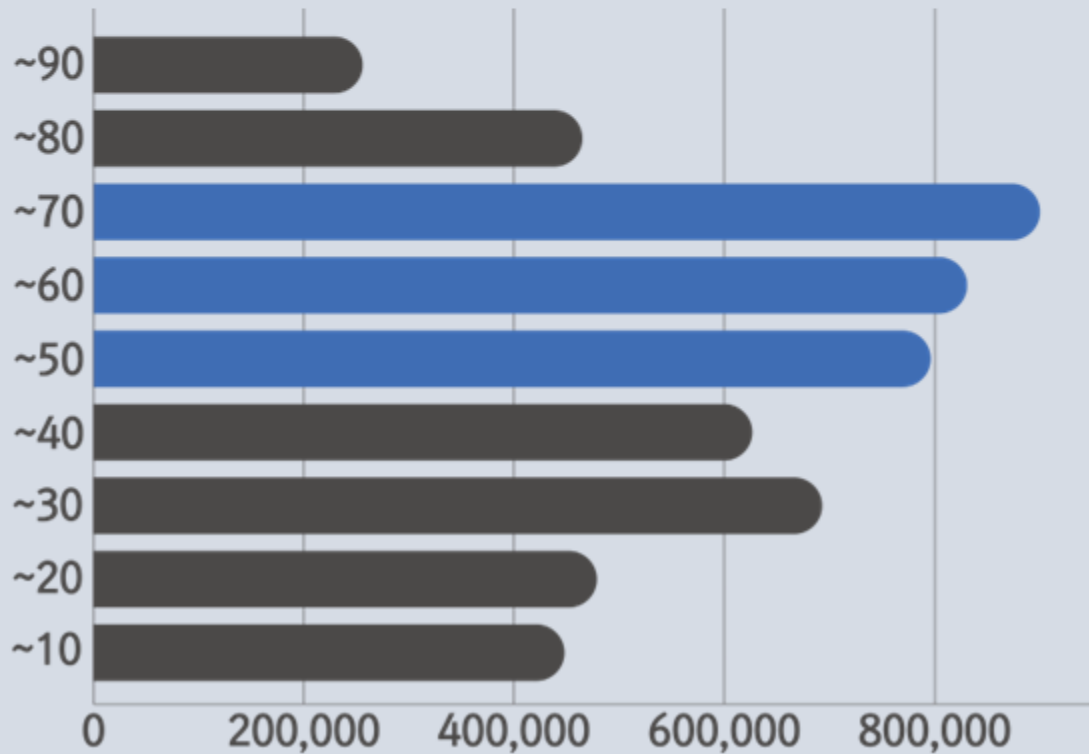
02 코드

03 전망

04 기대효과

우리나라 연령대가 점점 높아지고 있으며
실버 산업이 확장되고 있는 추세

고령인구 현황 (2020)



실버 산업 종류



주거



생활



의료



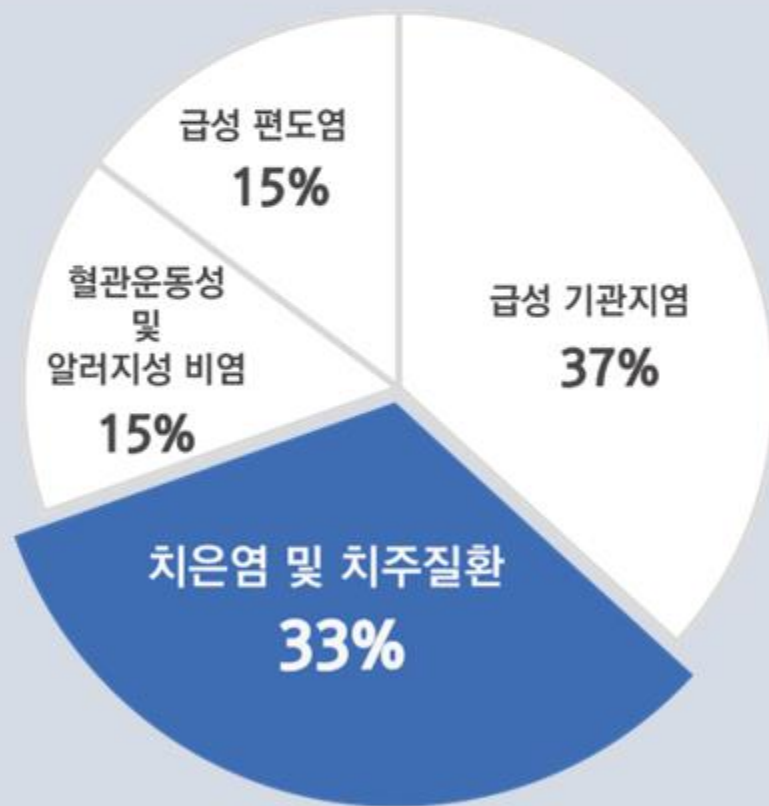
금융



여가활동

시장분석

확장성이 높은 의료분야



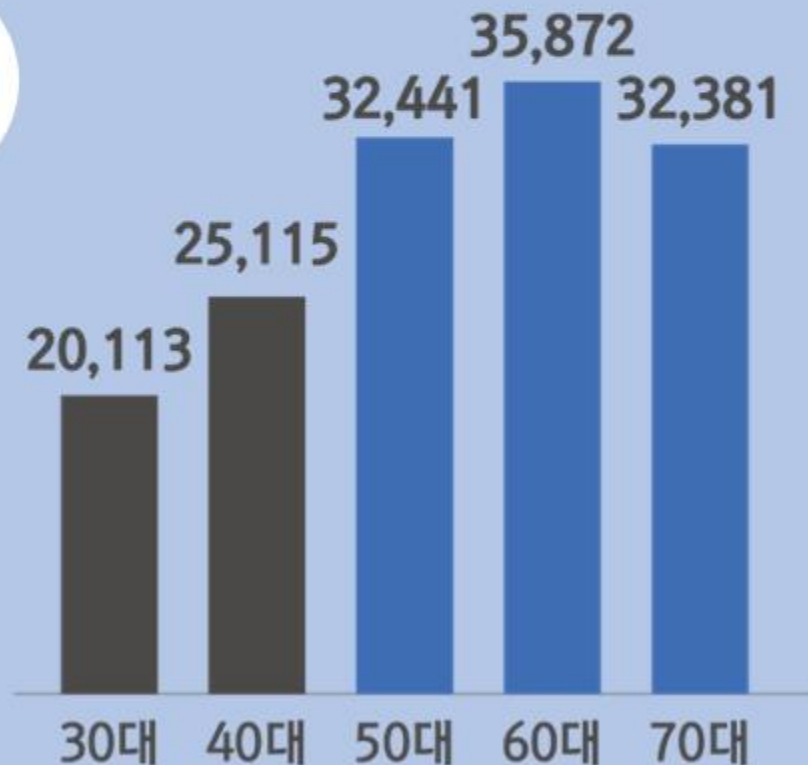
외래 다빈도 상병 순위별 현황



주요
치료 항목



치과



연령별 치주질환 진료인원

자료 = 건강증진개발원(2018)

특히 부산에 구강질환 환자가 증가하고 있으며,
구강관리에 도움되는 어플의 성능 향상 시 **긍정적인 효과**가 발생

효과



소비자

치과



친숙



병원

환자수
증가



이익
증대

사업 확장 가능성 ▲

나아가 부산시 뿐만 아니라

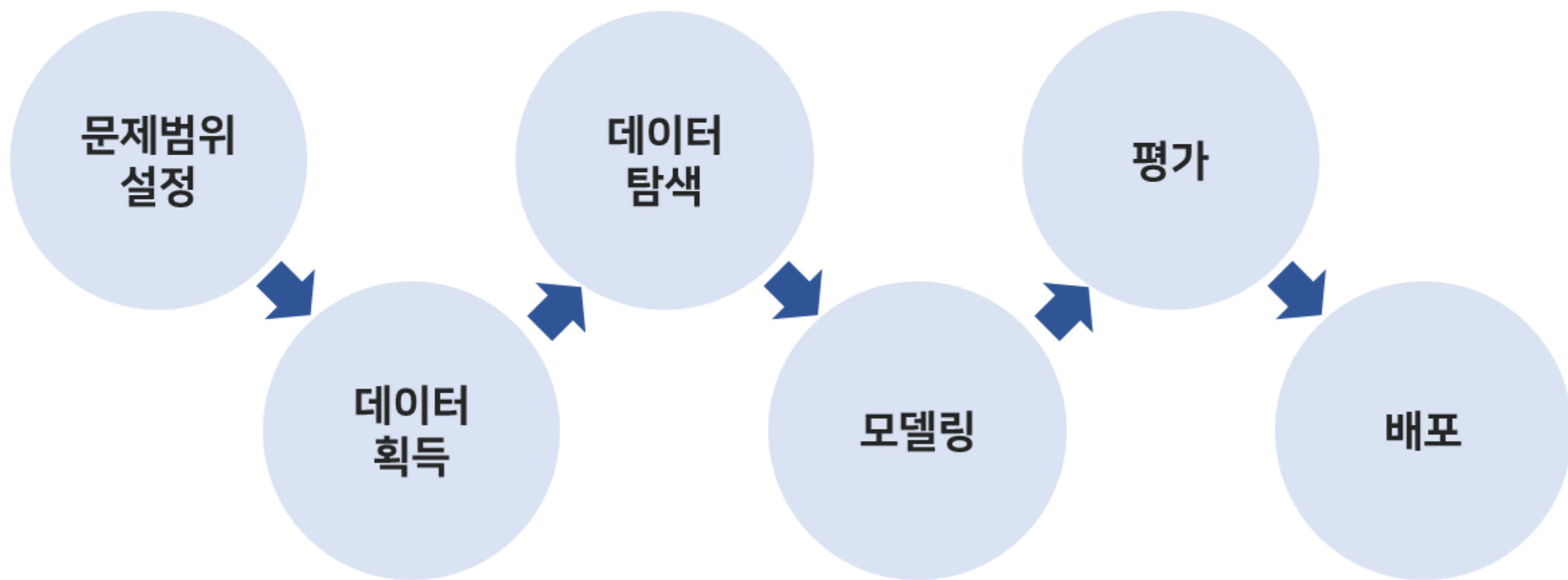
**전국적으로
확장 가능성**





코드

치아번호식별을 위한 프로젝트 사이클 구성



코드

그렇다면 어떤 코드를 사용할까? 데이터 처리 과정



데이터 수집

001_00095_01	2022-09-16 오전 11:06	JPG 파일
001_00095_01_a091	2022-09-27 오후 6:47	JSON 파일
001_00095_01_c031	2022-09-16 오전 11:06	JSON 파일
001_00097_05	2022-09-16 오전 11:06	JPG 파일
001_00097_05_a091	2022-09-27 오후 8:50	JSON 파일
001_00097_05_b057	2022-08-31 오전 10:42	JSON 파일
001_00097_05_c031	2022-09-16 오전 11:06	JSON 파일
001_00097_12	2022-09-16 오전 11:06	JPG 파일
001_00097_12_b057	2022-08-25 오후 3:33	JSON 파일
001_00097_12_c031	2022-09-16 오전 11:06	JSON 파일
001_00100_01	2022-09-16 오전 11:06	JPG 파일



데이터 정제

```
[4] # train data / json file 경우
train_filepaths = list(glob('/content/real_data/C project v21 darknet/train/* .txt'))

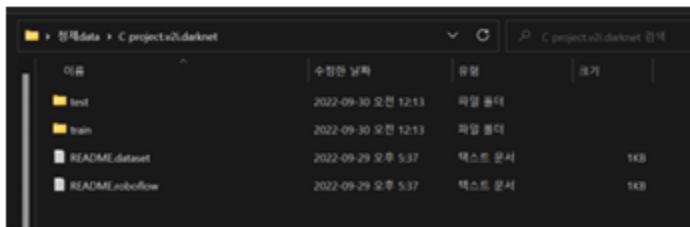
len(train_filepaths)

1740

# test data / json file 경우
test_filepaths = list(glob('/content/real_data/C project v21 darknet/test/* .txt'))

len(test_filepaths)

133
```



데이터 모델링

```
from glob import glob

for filename in glob('/content/real_data/C project v21 darknet/train/* .txt'):
    print(filename)

/content/real_data/C project v21 darknet/train/003_00073_12_c031_rf_c33a39276f03cf0f0543a548701fbc20.txt
/content/real_data/C project v21 darknet/train/003_00026_07_c031_rf_1b66f757acfc4e42e43c773af76c8c05.txt
/content/real_data/C project v21 darknet/train/003_00044_03_c031_rf_598600837bcb425c71d2584d74022251.txt
/content/real_data/C project v21 darknet/train/003_00028_10_c031_rf_7858d53325d45f097ab15f11465656f0d.txt
/content/real_data/C project v21 darknet/train/018_00053_07_c031_rf_611b148e4aafac5c5e4300cc2c906219.txt
/content/real_data/C project v21 darknet/train/003_00039_05_c031_rf_affa6405fa358a1255f4588e1b5e57be.txt
/content/real_data/C project v21 darknet/train/001_00085_11_c031_rf_06571d0c3924a9096ccb8a2fb49901a8.txt
/content/real_data/C project v21 darknet/train/003_00023_04_c031_rf_568e4287ccf752c2e51e32bcc924f110e.txt
/content/real_data/C project v21 darknet/train/002_00082_12_c031_rf_262c9d450a5bac61a9fe983feabf44b0.txt
/content/real_data/C project v21 darknet/train/002_00021_05_c031_rf_732897e24ee3ef6b7040b0dec9174dc8.txt
```

YOLOv5



코드

그렇다면 어떤 코드를 사용할까?
데이터 처리 과정



학습&테스트



AI모델



유지보수



데이터 추가
및
데이터 학습

코드

그렇다면 어떤 코드를 사용할까?

Training Data

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	Class	Images	Instances	P	R	mAP50	mAP50-95	100% 5/5 [00:12<00:00, 2.48s/it]
86/89	4.64G	0.02171	0.04294	0.02862	164	640: 100% 109/109 [00:57<00:00, 1.88it/s]	all	133	1398	0.363	0.569	0.368	0.27	
	Class	Images	Instances	P	R	mAP50	11	133	72	0.425	0.75	0.473	0.391	
	all	133	1398	0.355	0.59	0.348	12	133	56	0.33	0.696	0.328	0.249	
						mAP50-95: 100% 5/5 [00:02<00:00, 1.91it/s]	13	133	1	1	0	0	0	
							14	133	47	0.358	0.7	0.385	0.302	
							15	133	35	0.327	0.653	0.311	0.22	
87/89	4.64G	0.02117	0.04227	0.02818	156	640: 100% 109/109 [01:00<00:00, 1.80it/s]	16	133	29	0.275	0.655	0.313	0.191	
	Class	Images	Instances	P	R	mAP50	17	133	20	0.248	0.45	0.237	0.168	
	all	133	1398	0.361	0.546	0.356	18	133	18	0.285	0.421	0.302	0.185	
						mAP50-95: 100% 5/5 [00:02<00:00, 1.81it/s]	21	133	10	0.185	0.3	0.224	0.146	
88/89	4.64G	0.02151	0.04353	0.02821	165	640: 100% 109/109 [00:58<00:00, 1.86it/s]	22	133	70	0.432	0.759	0.484	0.398	
	Class	Images	Instances	P	R	mAP50	23	133	61	0.44	0.77	0.496	0.378	
	all	133	1398	0.359	0.58	0.361	24	133	50	0.41	0.707	0.454	0.365	
						mAP50-95: 100% 5/5 [00:02<00:00, 2.41it/s]	25	133	39	0.378	0.655	0.443	0.331	
89/89	4.64G	0.02145	0.04267	0.02826	204	640: 100% 109/109 [00:58<00:00, 1.86it/s]	26	133	34	0.357	0.588	0.335	0.228	
	Class	Images	Instances	P	R	mAP50	27	133	25	0.352	0.48	0.405	0.281	
	all	133	1398	0.353	0.584	0.364	28	133	20	0.394	0.553	0.431	0.295	
						mAP50-95: 100% 5/5 [00:02<00:00, 1.94it/s]	31	133	7	0.111	0.286	0.3	0.164	
							32	133	69	0.397	0.71	0.43	0.311	
							33	133	70	0.349	0.671	0.423	0.323	
							34	133	67	0.353	0.707	0.374	0.291	
							35	133	60	0.323	0.567	0.418	0.312	
							36	133	48	0.317	0.542	0.371	0.285	
							37	133	43	0.343	0.545	0.403	0.327	
							38	133	33	0.378	0.545	0.525	0.409	
							41	133	25	0.528	0.493	0.573	0.375	
							42	133	69	0.421	0.747	0.436	0.322	

90 epochs completed in 1.596 hours.

Optimizer stripped from drive/MyDrive/yolov5-master/runs/train/yolov5_tooth3/weights/last.pt, 14.6MB

Optimizer stripped from drive/MyDrive/yolov5-master/runs/train/yolov5_tooth3/weights/best.pt, 14.6MB

코드

그렇다면 어떤 코드를 사용할까?

Precision & Recall rate

```
!python /content/drive/MyDrive/yolov5-master/detect.py --weights /content/drive/MyDrive/yolov5-master/runs/train/yolov5_tooth3/weights/best.pt --img 320 --conf 0.5 --source /content/drive/MyDrive/darknet/test --c

detect: weights=['/content/drive/MyDrive/yolov5-master/runs/train/yolov5_tooth3/weights/best.pt'], source=/content/drive/MyDrive/darknet/test, data=drive/MyDrive/yolov5-master/data/coco128.yaml, imgs=[320, 320]
YOLOv5 2022-9-29 Python-3.7.14 torch-1.12.1+cu113 CUDA:0 (Tesla T4, 15110MiB)

Fusing layers...
Model summary: 157 layers, 7104520 parameters, 0 gradients, 16.0 GFL0Ps
image 1/133 /content/drive/MyDrive/darknet/test/001_00097_05_c031.rf.febeacld13eda9e78f7d0b9986838d4.jpg: 320x320 1 33, 2 43s, 8.4ms
image 2/133 /content/drive/MyDrive/darknet/test/001_00097_12_c031.rf.583054232c21141c36b1d40b6b9747ec.jpg: 320x320 (no detections), 7.9ms
image 3/133 /content/drive/MyDrive/darknet/test/001_00100_01_c031.rf.dfc404deb6a6cfcac8880ee05e2b6d.jpg: 320x320 (no detections), 24.9ms
image 4/133 /content/drive/MyDrive/darknet/test/001_00100_03_c031.rf.34c2bf1a85f2aeld6d08e55effb0b18c.jpg: 320x320 1 24, 8.0ms
image 5/133 /content/drive/MyDrive/darknet/test/001_00100_09_c031.rf.39557d0c1e5a8d208b6446e9dcca769.jpg: 320x320 1 25, 12.3ms
image 6/133 /content/drive/MyDrive/darknet/test/003_00009_02_c031.rf.6el1d85c9ba4679287cb09f1b8471090.jpg: 320x320 (no detections), 7.9ms
image 7/133 /content/drive/MyDrive/darknet/test/003_00009_12_c031.rf.a403b6ca9ff5c9addb490357a82779cc.jpg: 320x320 3 33s, 2 43s, 7.7ms
image 8/133 /content/drive/MyDrive/darknet/test/003_00010_01_c031.rf.89ed3894bf4837086e6b1d2c82971a7a.jpg: 320x320 (no detections), 7.7ms
image 9/133 /content/drive/MyDrive/darknet/test/003_00010_02_c031.rf.201dea432ca5a4b9f4a34607f19f1fb6.jpg: 320x320 1 33, 8.1ms
image 10/133 /content/drive/MyDrive/darknet/test/003_00011_01_c031.rf.87f57dc914539ac6f0a0b09f13fa2120.jpg: 320x320 (no detections), 7.9ms
image 11/133 /content/drive/MyDrive/darknet/test/003_00011_02_c031.rf.1082384af31d55bfffca4ee538ea02c5.jpg: 320x320 1 36, 7.7ms
image 12/133 /content/drive/MyDrive/darknet/test/003_00011_05_c031.rf.98b50520763999821001457a90e8f3b5.jpg: 320x320 1 36, 2 45s, 8.0ms
image 13/133 /content/drive/MyDrive/darknet/test/003_00023_02_c031.rf.c6c03b353cd5536997abdcf929b4be0e.jpg: 320x320 2 33s, 8.1ms
image 14/133 /content/drive/MyDrive/darknet/test/003_00024_09_c031.rf.alb7adca2dfcd39b4b6f5008b6a69dcd.jpg: 320x320 (no detections), 7.9ms
image 15/133 /content/drive/MyDrive/darknet/test/003_00024_12_c031.rf.4e80df83509bc2f29329972f4efab088.jpg: 320x320 1 37, 1 38, 8.5ms
image 16/133 /content/drive/MyDrive/darknet/test/003_00026_03_c031.rf.7f93310439c477c422f31422df88baf.jpg: 320x320 (no detections), 7.8ms
image 17/133 /content/drive/MyDrive/darknet/test/003_00027_03_c031.rf.0bb419f2d60f188e792b328489976aeb0.jpg: 320x320 (no detections), 8.0ms
image 18/133 /content/drive/MyDrive/darknet/test/003_00027_07_c031.rf.7c3ff254a4060f5b0eld80248a95c0db.jpg: 320x320 (no detections), 7.5ms
image 19/133 /content/drive/MyDrive/darknet/test/003_00027_08_c031.rf.5f8bb5f77991f40f56a40e9c0171d136.jpg: 320x320 1 24, 7.6ms
image 20/133 /content/drive/MyDrive/darknet/test/003_00028_11_c031.rf.c8c69d51a83973854a1afb7885984d34.jpg: 320x320 1 33, 2 36s, 2 37s, 1 38, 1 42, 1 45, 1 48, 7.9ms
image 21/133 /content/drive/MyDrive/darknet/test/003_00029_02_c031.rf.8bf444355ea6b9901939905c294c47eb.jpg: 320x320 2 33s, 1 42, 2 43s, 1 45, 7.4ms
image 22/133 /content/drive/MyDrive/darknet/test/003_00029_07_c031.rf.5995f431141a8bedeb92aafde4d39b2d.jpg: 320x320 (no detections), 7.6ms
image 23/133 /content/drive/MyDrive/darknet/test/003_00030_01_c031.rf.1aa0027f87d80125d17fca265eeal467.jpg: 320x320 (no detections), 11.3ms
image 24/133 /content/drive/MyDrive/darknet/test/003_00030_10_c031.rf.3c00c6e02ad2367efe38414b655c8c87.jpg: 320x320 1 45, 1 48, 8.1ms
image 25/133 /content/drive/MyDrive/darknet/test/003_00031_04_c031.rf.f0fc3a10468dba9d0e5c7e0aaf4ed2d6.jpg: 320x320 1 33, 1 36, 1 37, 1 38, 1 41, 1 43, 9.9ms
image 26/133 /content/drive/MyDrive/darknet/test/003_00031_06_c031.rf.b7c14c5ace56b3d3875242045a878156.jpg: 320x320 1 42, 2 48s, 7.7ms
image 27/133 /content/drive/MyDrive/darknet/test/003_00032_01_c031.rf.08b07766c0f9eef3f773404e55e4aa5e.jpg: 320x320 (no detections), 7.9ms
image 28/133 /content/drive/MyDrive/darknet/test/003_00032_02_c031.rf.10ab374e7049acc8aaa9dc1247025ec9.jpg: 320x320 (no detections), 8.4ms
image 29/133 /content/drive/MyDrive/darknet/test/003_00033_06_c031.rf.8a206df773db6d5f532e8c555f5a495.jpg: 320x320 1 41, 1 48, 1 none, 8.2ms
image 30/133 /content/drive/MyDrive/darknet/test/003_00033_12_c031.rf.51efd7c89a07b8bb04f476901c27428b.jpg: 320x320 1 37, 1 38, 1 41, 1 none, 7.7ms
image 31/133 /content/drive/MyDrive/darknet/test/003_00034_03_c031.rf.2falb588ebdd5cbcc0192c291b733f1.jpg: 320x320 (no detections), 7.8ms
image 32/133 /content/drive/MyDrive/darknet/test/003_00034_04_c031.rf.0eaa32b03997efae27e574eed5aaf4a6.jpg: 320x320 1 38, 1 41, 8.6ms
```

코드

그렇다면 어떤 코드를 사용할까?
Precision & Recall rate

```
import glob
from IPython.display import Image, display

for imageName in glob.glob('/content/drive/MyDrive/yolov5-master/runs/detect/exp/*.jpg'):
    display(Image(filename=imageName))
    print("\n")
```



```
import glob
from IPython.display import Image, display

for imageName in glob.glob('/content/drive/MyDrive/yolov5-master/runs/detect/exp/*.jpg'):
    display(Image(filename=imageName))
    print("\n")
```



```
import glob
from IPython.display import Image, display

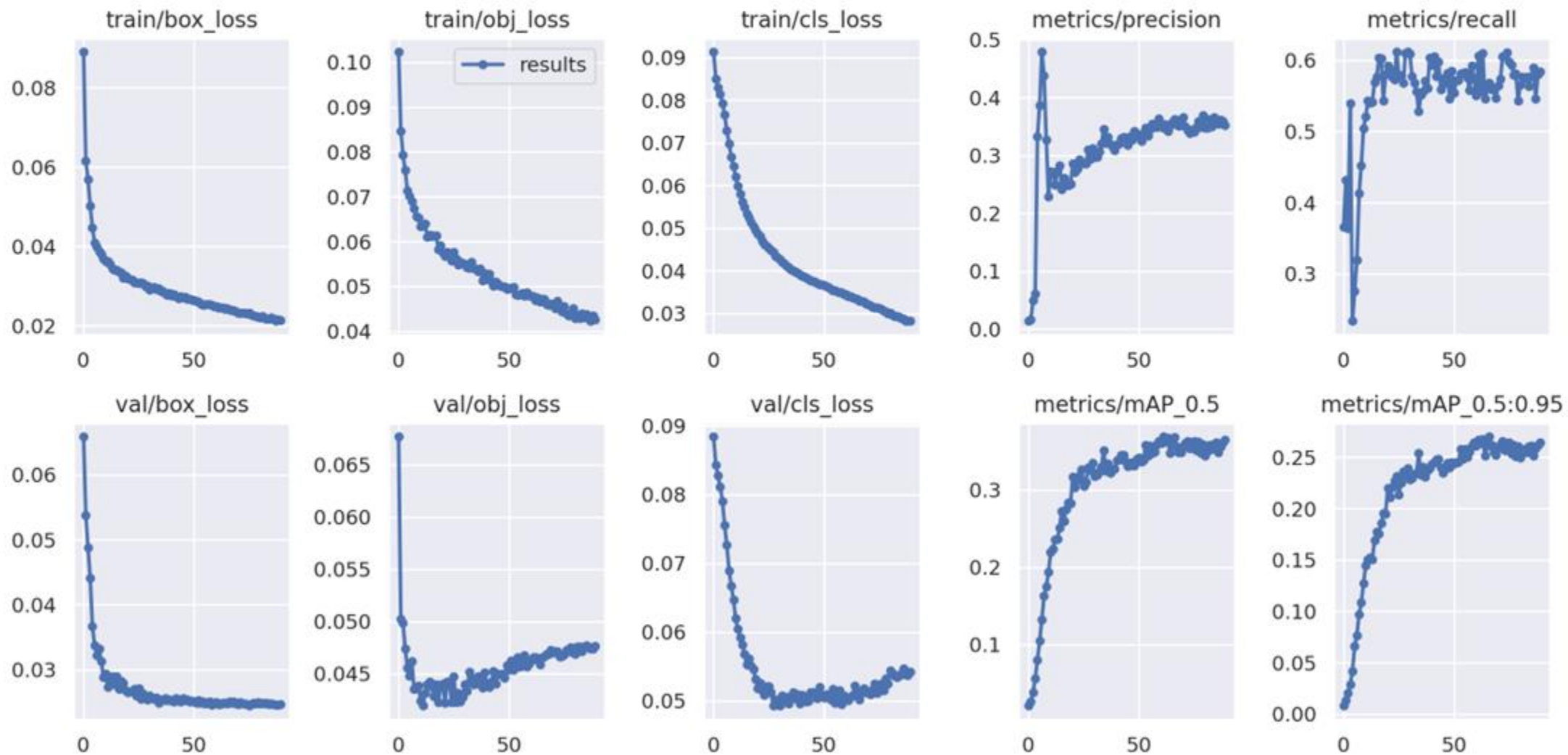
for imageName in glob.glob('/content/drive/MyDrive/yolov5-master/runs/detect/exp2/*.jpg'):
    display(Image(filename=imageName))
    print("\n")
```





코드

Results



프로젝트를 활용한 서비스 확장 가능성



딥러닝과 빅데이터를 활용한
치주질환 관리 **플랫폼 기술 개발**



치의학 의료 데이터
인공지능 의료기기 개발 가능



증상, 질병 이미지로 전문 자료를
검색 및 환자에게 가장 **적합한 치료법 제시** 가능



다양한 **이미지 인식 및 영상 인식** 등을 이용한
새로운 형태의 서비스 출시

● ● ● ●
전망

서비스 사용으로 인해
얻을 수 있는 긍정적인 효과

▶ 서비스

인건비 절약

정확도 상승

높아진
영상데이터질

어플 사용률 ▲

▶ 고령화 사회



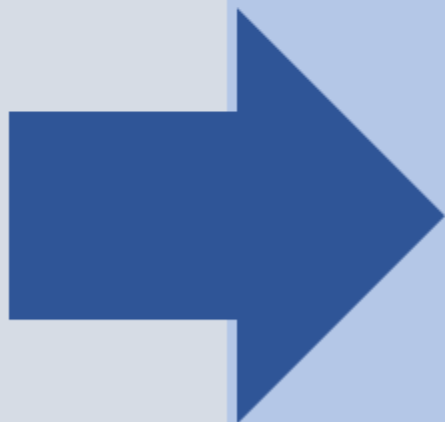
질병 완화

● ● ● ●
전망

부정적이었던 병원 인식에 대한 긍정적 변화

부정

과잉진료 시설
평판 인지도
높은 진료 비용



긍정

쾌적함 꼼꼼한 진료
진료 효과/증상 개선
저렴한 진료 비용

최종적으로 서비스를 이용하였을 때 얻을 수 있는 기대효과

다양한 치의학 의료영상데이터에서 활용 가능

- 교정 치료 시 필요한 해부학적 랜드마크 자동 주석, 치아 자동 분할
- 치아우식, 치주염, 매복치, 가성낭종 등 자동으로 병소 검출
- 이전 치료내역 포함 자동 차팅
- 교정 시뮬레이션 및 성장 단계 모니터링
- 자동 보철물 디자인을 통한 이상적인 3D 수복물 자동 디자인 및 마진 표시, 환자 맞춤 스마일 라인 디자인
- 보험사기 의심 케이스 자동 체크를 통한 보험사기 예방

최종적으로 서비스를 이용하였을 때 얻을 수 있는 기대효과

교정치료, 부정교합 치료 판단 정확성 증가

- ANN 기술로 임상에서 의사 결정 과정에 도움을 극대화
- 환자로부터 예측 가능한 최상의 결과를 얻어 교정치료 가능
- 돌이킬 수 없는 치료를 진행하기 전에 이 방법이 최선인지, 대체할 치료방법은 없는지 신중하게 판단 가능

AI 임상 적용 사례 증가

대표적으로 교정, 치주, 근관 치료, 구강병리학 등 치과에서 AI를 활용한 다양한 임상 및 진료 효율성 향상 사례 증가

의사와 환자 간의 관계 강화

- AI 지원 이미지를 통해 치과 의사들은 간단하고 쉽게 이해 가능한 바운딩 박스로 관심영역을 제시
- 환자들에게 엑스레이에서 일어나고 있는 일을 정확히 알려주면서 훨씬 더 명확한 정보를 제공하는 동시에 치료 계획이 합리적인 기반에 입각하도록 보장



감사합니다