

연구 포트폴리오

조 윤 상

프로젝트 1. 변분 오토인코더(VAE)와 k-means 알고리즘을 활용한 웨이퍼 결함패턴 클러스터링 알고리즘

○ 프로젝트 설명

- 본 프로젝트는 반도체 웨이퍼 데이터 셋(WM-811K)을 결함패턴 별로 군집화(clustering)하여 반도체 결함 식별 전문가가 분류해 좋은 데이터 셋에서 결함패턴을 보다 세분화하여 유사한 그룹별로 구분하기 위한 프로젝트입니다.

○ 담당 업무

- 알고리즘 전체 flowchart를 도출하고, 논문 리뷰 및 Keras를 이용한 파이썬 스크립트 코드 구현과 실험하는 역할을 담당했습니다.

○ 사용기술

- Tensorflow, Keras, Python

○ 참고문헌

- Tulala, P., Mahyar, H., Ghalebi, E., & Grosu, R. (2018, July). Unsupervised wafermap patterns clustering via variational autoencoders. In: *2018 International Joint Conference on Neural Networks(IJCNN)*. (pp.1-8). IEEE

프로젝트 1. 변분 오토인코더(VAE)와 k-means 알고리즘을 활용한 웨이퍼 결함패턴 클러스터링 알고리즘

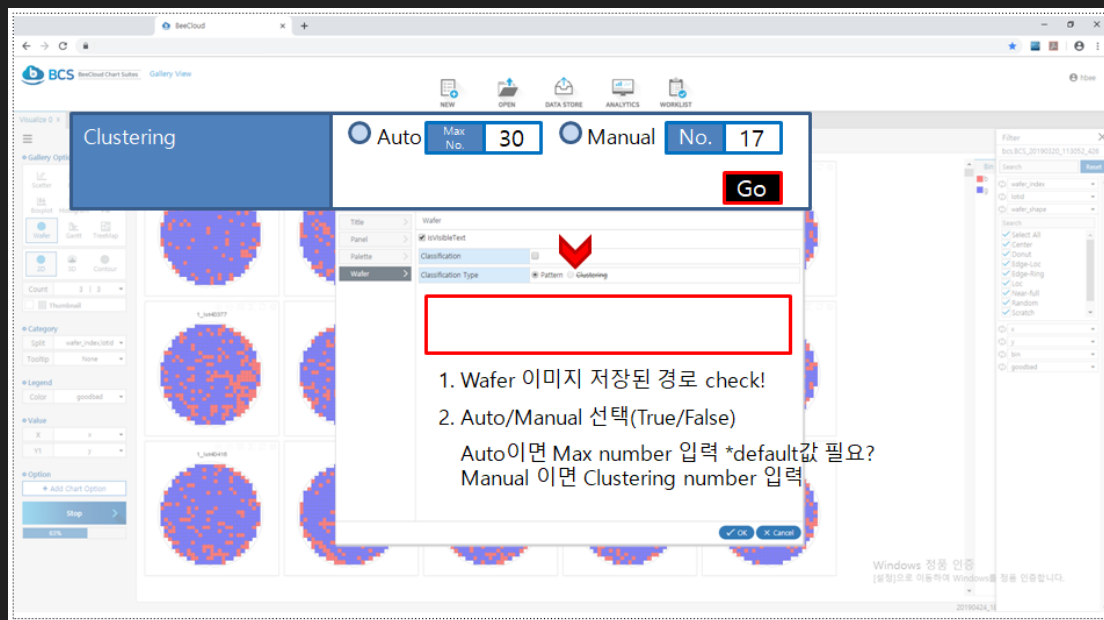


그림 1. 군집화 동작 알고리즘 실행 방법에 대한 예

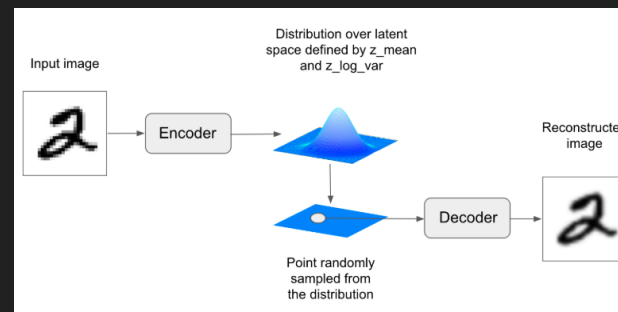


그림 2. VAE

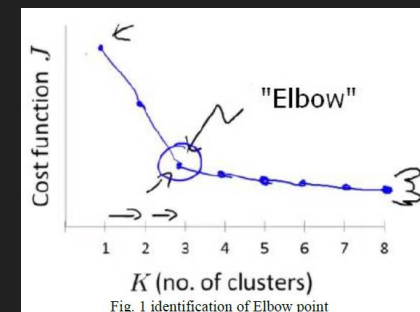


그림 3. Elbow method

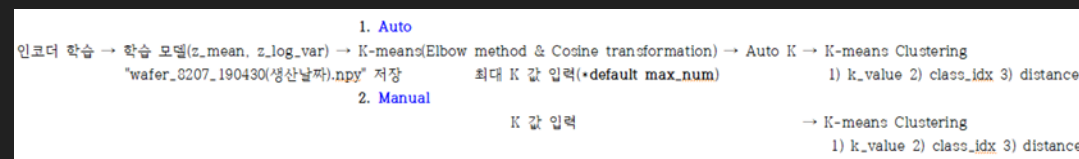


그림 4. Flowchart

프로젝트 2. 다중 사용자 정의 클래스 동시 학습 알고리즘

○ 프로젝트 설명

- 본 프로젝트는 이전 프로젝트에서 진행했던 웨이퍼 데이터 셋 군집화 결과를 이용하여 다중 사용자가 정의한 클래스를 학습 데이터 셋으로 사용하여 동시에 신경망 학습을 진행할 때, 클라우드 리소스(메모리, 소요시간)를 사용량을 측정하고 적용하는 프로젝트입니다.

○ 담당 업무

- Keras를 이용한 파이썬 스크립트 코드 구현, 실험하는 역할을 담당했습니다.

○ 사용기술

- Keras, Python

프로젝트 2. 다중 사용자 정의 클래스 동시 학습 알고리즘

3. 사용자 정의 클래스 training

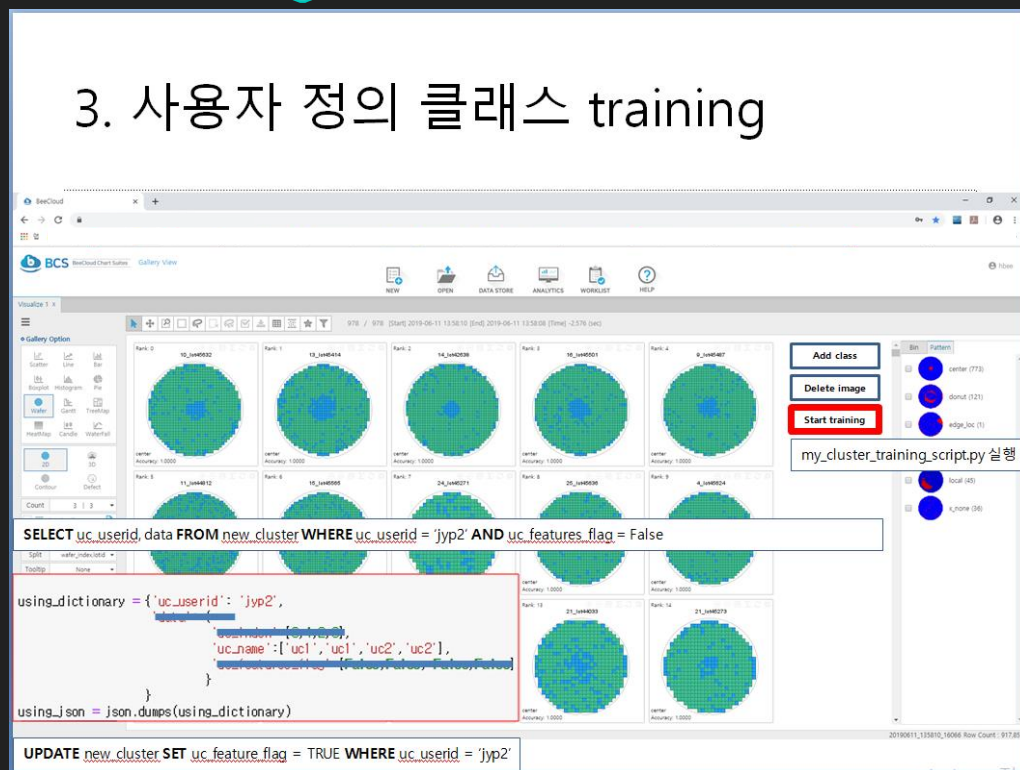


그림 1. 사용자 정의 클래스 실행 방법에 대한 예

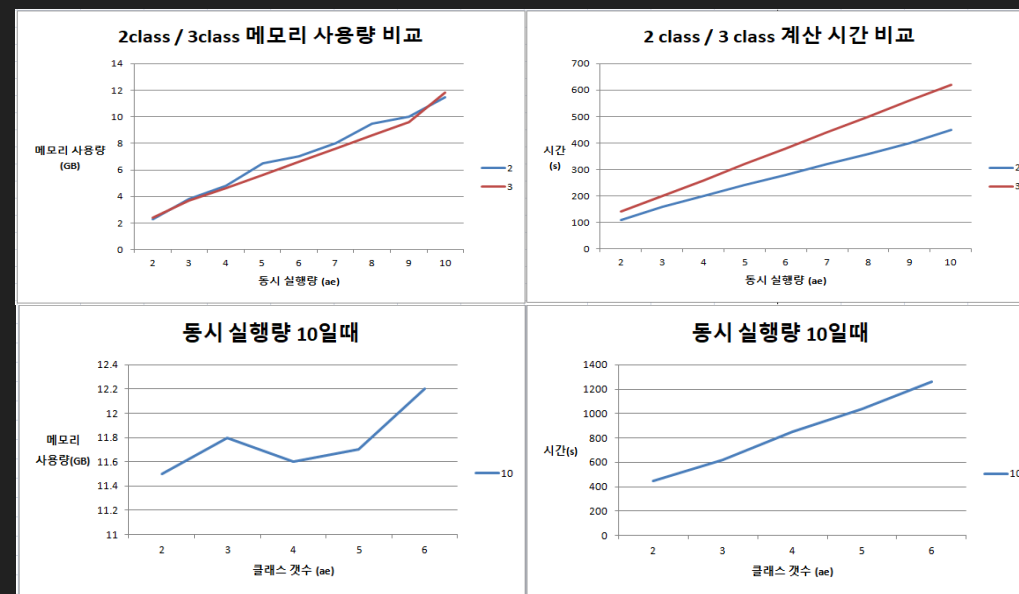


그림 2. 동시 학습에 따른 리소스 사용량 비교 I

		2개 동시	3개 동시	4개 동시	5개 동시	6개 동시	7개 동시	8개 동시	9개 동시	10개 동시
2 Class	메모리	2.3 G	3.8 G	4.8 G	6.5 G	7.5	8.5	9.5 G	10.5	11.5 G
	시간	110 s	160 s	200 s	240 s	280	320	360 s	400	450 s
3 Class	메모리	2.4 G	3.7 G	4.6	5.6	6.6	7.6	8.6	9.6	11.8 G
	시간	140 s	200 s	260	320	380	440	500	560	620 s

그림 3. 동시 학습에 따른 리소스 사용량 비교 II

프로젝트 3. 심층 합성곱 생성적 적대 신경망(DCGAN)기반 웨이퍼 결함 패턴 생성 및 평가

○ 프로젝트 설명

- 본 프로젝트는 분류 모델의 학습에 사용할 데이터 셋에서 클래스 불균형 문제를 해결하고 테스트 이미지의 예측 정확도를 높이기 위해 DCGAN 알고리즘을 사용하여 학습 웨이퍼 결함패턴 이미지를 생성하는 자기지도학습 프로젝트입니다.

○ 담당 업무

- UI/UX 구상, 논문 리뷰 및 Keras를 이용한 파이썬 코드 구현 및 유효성 평가를 실험하는 역할을 담당했습니다.

○ 사용기술

- Keras, Python

○ 참고문헌

- Nero, matthew, et al. Discovering Interesting Plots in Production Yield Data Analytics. *arXiv preprint arXiv: 1807. 03920*, 2018.

프로젝트 3. 심층 합성곱 생성적 적대 신경망(DCGAN)기반 웨이퍼 결함 패턴 생성 및 평가

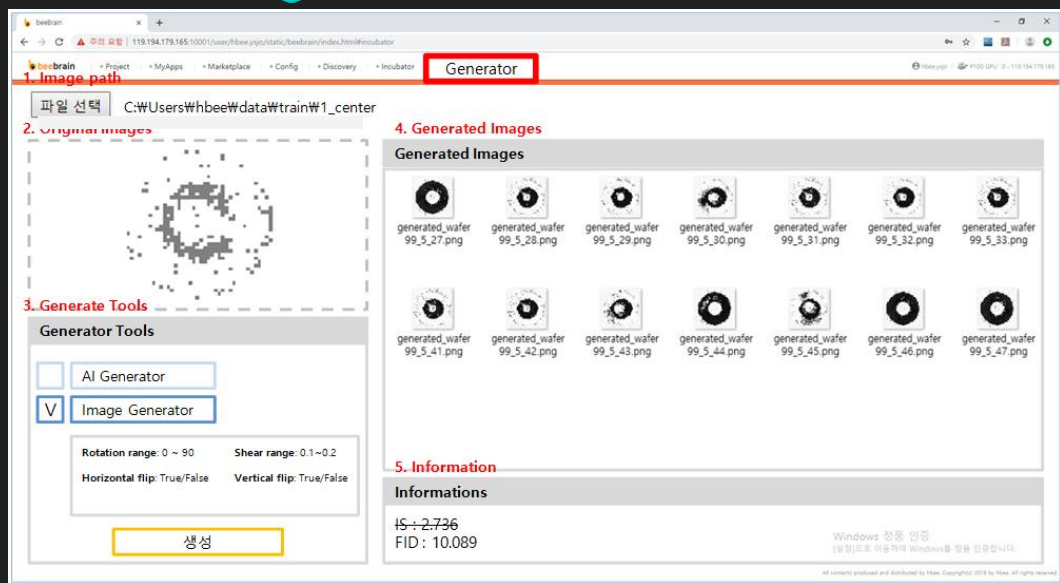


그림 1. 웨이퍼 결함 패턴 생성 화면

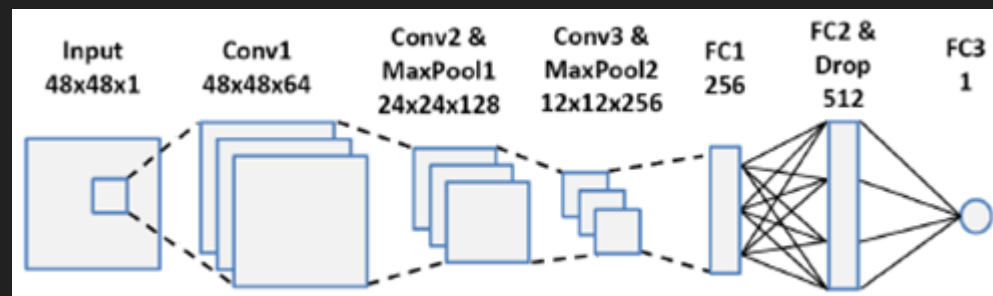


그림 2. CNN architecture for the discriminator

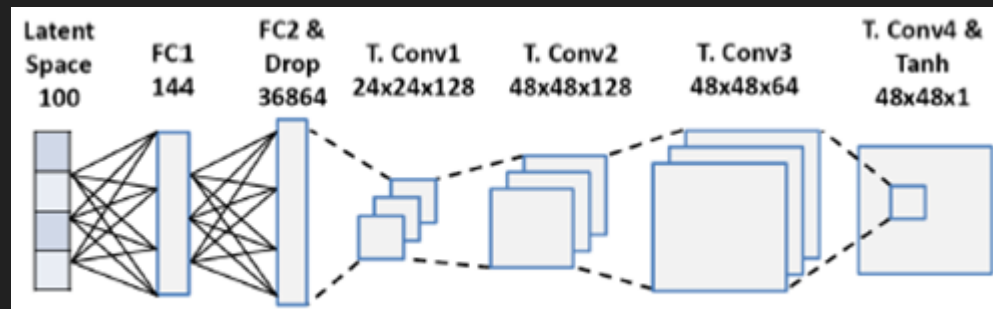


그림 3. CNN architecture for the generator

프로젝트 3. 심층 합성곱 생성적 적대 신경망(DCGAN)기반 웨이퍼 결함 패턴 생성 및 평가



그림 4. 웨이퍼 데이터 셋 생성 및 원본 이미지

$$FID = \|m - m_w\|_2^2 + Tr(C + C_w - 2(CC_w)^{1/2}),$$

그림 5. FID 계산



그림 6. 이진화 임계값에 따른 생성 이미지의 변화

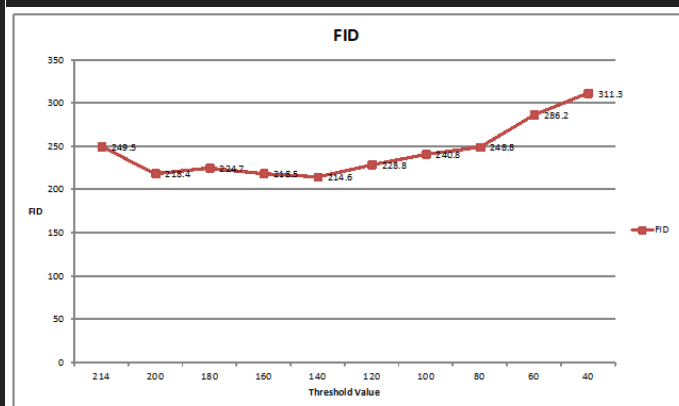


그림 7. 이진화 임계값 변화에 따른 생성 이미지 FID 측정값 변화

프로젝트 3. 심층 합성곱 생성적 적대 신경망(DCGAN)기반 웨이퍼 결함 패턴 생성 및 평가

original images 60 장 기준 / 1class

generated images: 400 장 기준

실행 위치: P100

소요 시간			
GAN		ImageDataGenerator	
패키지 로드: 1.9s			
1. down size	0.06s	1. increasing	1.2s
2. increasing	0.4s		
3. generated (150 epochs)	195s		
4. binarize	0.8s		
5. up size	0.3s		
increasing image for compare: 1.2s (FID 구할 때 사용)			
SI: 3.5s			
FID: 약 58s			
합계	약 261s (4분 21초)	합계	약 66s (1분 6초)

그림 8. 데이터 생성 시 실행에 소요되는 시간

original images 60 장 기준 / 1 class

generated images: 400 장 기준

저장 공간			
GAN		ImageDataGenerator	
original_image: 140KB			
1. down size	26KB	1. increasing	1.64MB
2. increasing	1.64MB		
3. generated	1.62MB		
4. binarize	169KB		
5. up size	1.37MB		
increasing_image: 1.65MB			
inception model: 169MB			
모델 제외할 경우	6MB	모델 제외할 경우	3MB
합계	약 175MB	합계	약 172MB

그림 9. 데이터 생성 시 필요한 저장 공간

original images 60 장 기준 / 1class
generated images: 400 장 기준
실행 위치: P100

소요 메모리			
GAN		ImageDataGenerator	
사용량	3GB	사용량	3GB

* FID값 계산할 때 44개의 core에서 CPU 점유율 100%까지 올라감.

그림 10. 데이터 생성 시 소요 메모리

프로젝트 4. 의사결정트리(Decision Tree) 알고리즘 기반 데이터 시각화 추천(Visualization recommendation)

○ 프로젝트 설명

- 본 프로젝트는 임의의 데이터 셋(ex. 'abc.csv')으로부터 알고리즘으로부터 도출된 결과값으로 시각화 방법을 추천하여 시각화 솔루션 사용자에게 데이터에서 유의미한 차트를 스스로 만들어 제공하는 프로젝트입니다.

○ 담당 업무

- UI/UX 구상, 논문 리뷰 및 파이썬 코드 실험 및 구현하는 역할을 담당했습니다.

○ 사용기술

- Python

○ 참고문헌

- Luo, Yuyu, et al. DeepEye: towards automatic data visualization. In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 2018. p. 101-112.

프로젝트 4. 의사결정트리(Decision Tree) 알고리즘 기반 데이터 시각화 추천(Visualization recommendation)

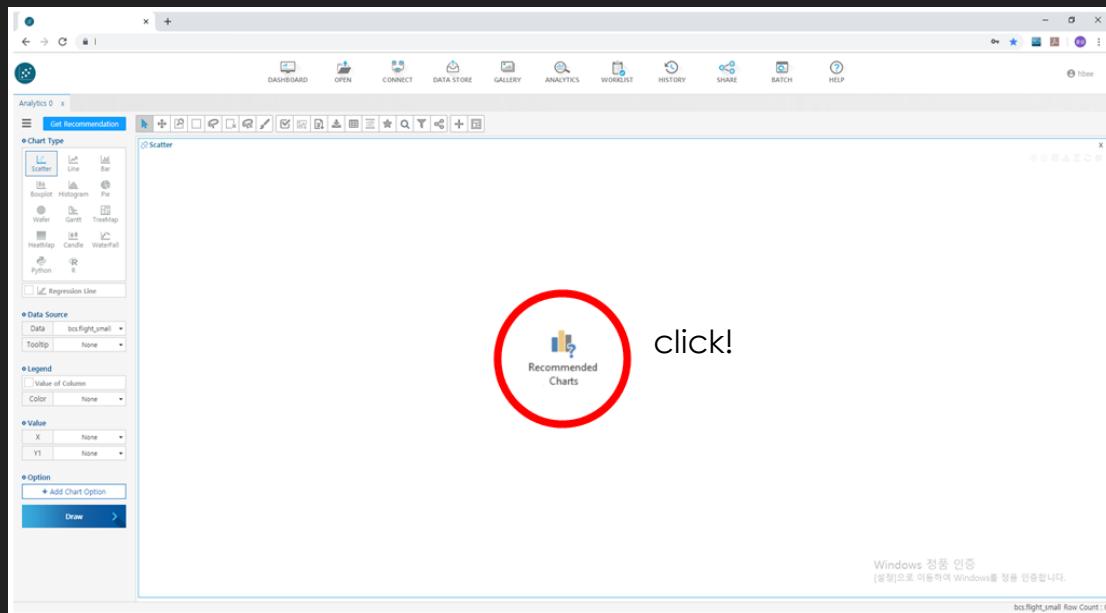


그림 1. 메인 화면 차트 추천 아이콘 예



그림 2. 추천 방법 및 컬럼 선택에 따른 차트 추천

프로젝트 5. 클라우드 기반 사전훈련모델(Pre-trained Model)을 이용한 웨이퍼 결함 패턴 분류 알고리즘

○ 프로젝트 설명

- 본 프로젝트는 로컬 사용자의 리소스를 사용하는 구글 Teachable machine을 벤치마크한 프로젝트로 반도체 웨이퍼 데이터 셋 분류에 클라우드 리소스(GPU V100)를 사용하여 제품의 양불을 판단하는 전문가를 구하기 쉽지 않은 환경에 제공하기 위한 프로젝트입니다.

○ 담당 업무

- UI/UX 구상, Keras를 이용한 파이썬 코드 실험 및 구현하는 역할을 담당했습니다.

○ 사용기술

- Keras, Python, Tensorflow serving

○ 참고문헌

- <https://teachablemachine.withgoogle.com/>

프로젝트 5. 클라우드 기반 사전훈련모델(Pre-trained Model)을 이용한 웨이퍼 결함 패턴 분류 알고리즘

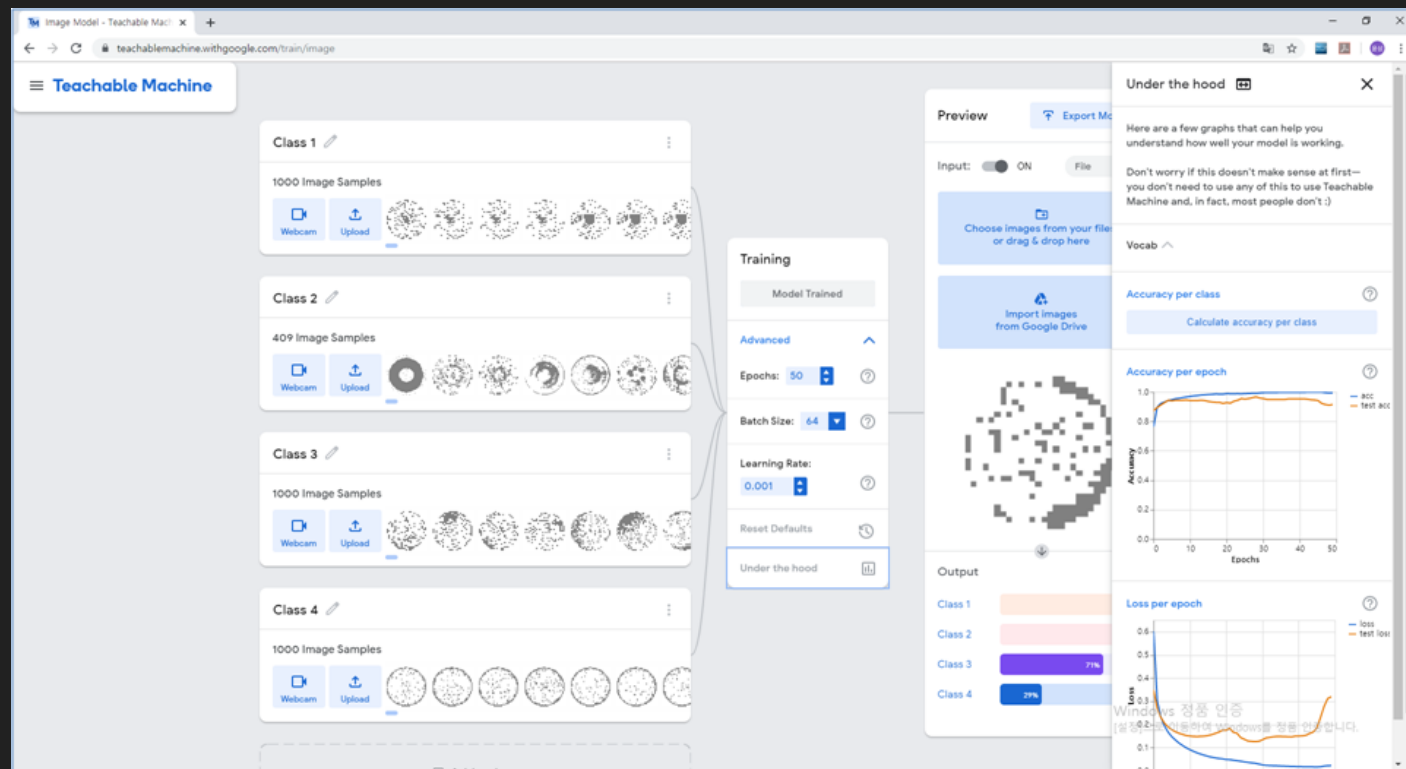
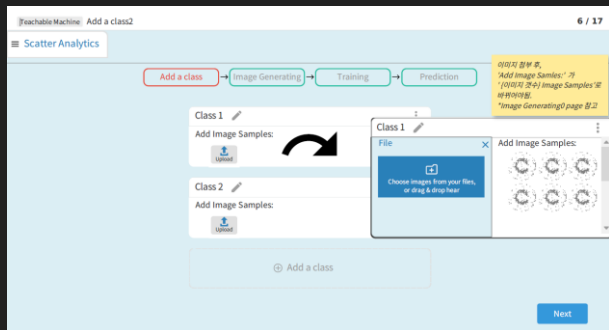
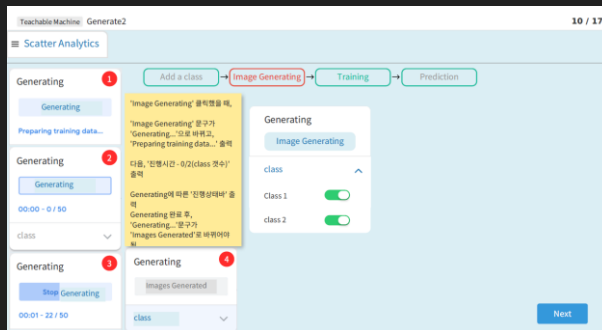


그림 1. Google teachable machine

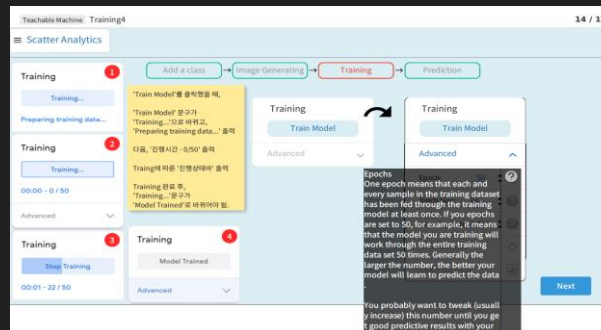
프로젝트 5. 클라우드 기반 사전훈련모델(Pre-trained Model)을 이용한 웨이퍼 결함 패턴 분류 알고리즘



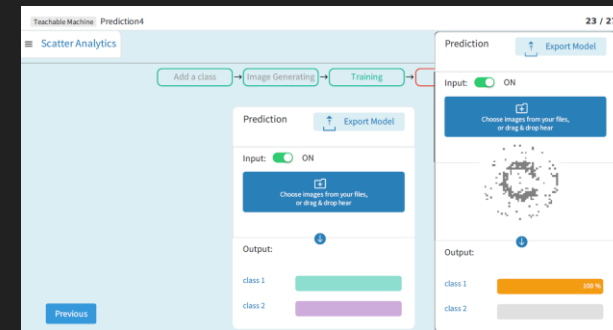
a. 클래스 추가



b. 이미지 생성(증식)



c. 데이터 셋 훈련



d. 테스트 이미지 예측

그림 2. 클라우드 기반 사전훈련모델을 이용한 웨이퍼 결함 패턴 분류

프로젝트 6. 억 규모의 이미지 분류를 위한 준지도 학습(semi-supervised learning) 알고리즘

○ 프로젝트 설명

- 본 프로젝트는 준지도학습(semi-supervised learning) 논문을 리뷰하고 웨이퍼 이미지 데이터 셋에 적용하여 적은 데이터 셋으로부터 억 단위의 테스트 데이터 셋으로부터 정확도가 높은 분류 모델을 만드는 프로젝트입니다.

○ 담당 업무

- Keras를 이용한 파이썬 코드 실험 및 구현하는 역할을 담당했습니다.

○ 사용기술

- Keras, Python

○ 참고문헌

- Yalniz, I. Zeki, et al. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

프로젝트 6. 억 규모의 이미지 분류를 위한 준지도 학습(semi-supervised learning) 알고리즘

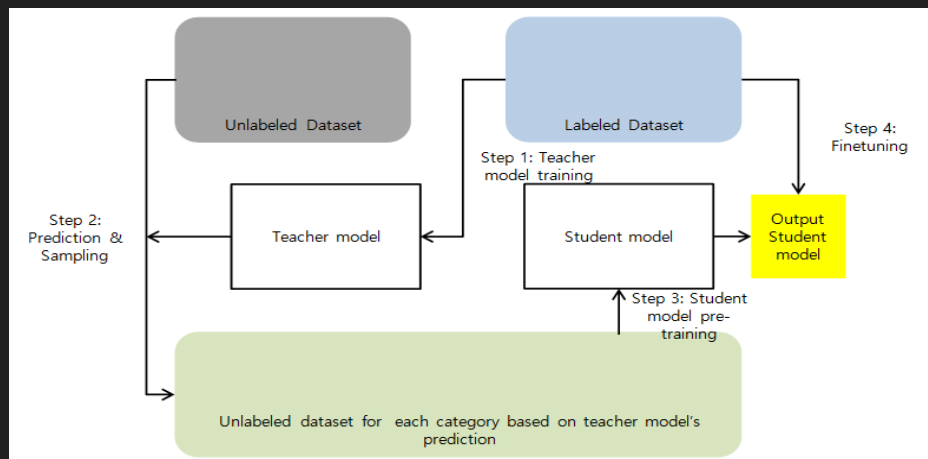


그림 1. Semi-supervised learning flowchart

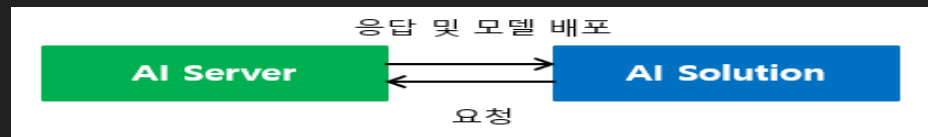


그림 2. 학습 모델 배포 및 응답 예

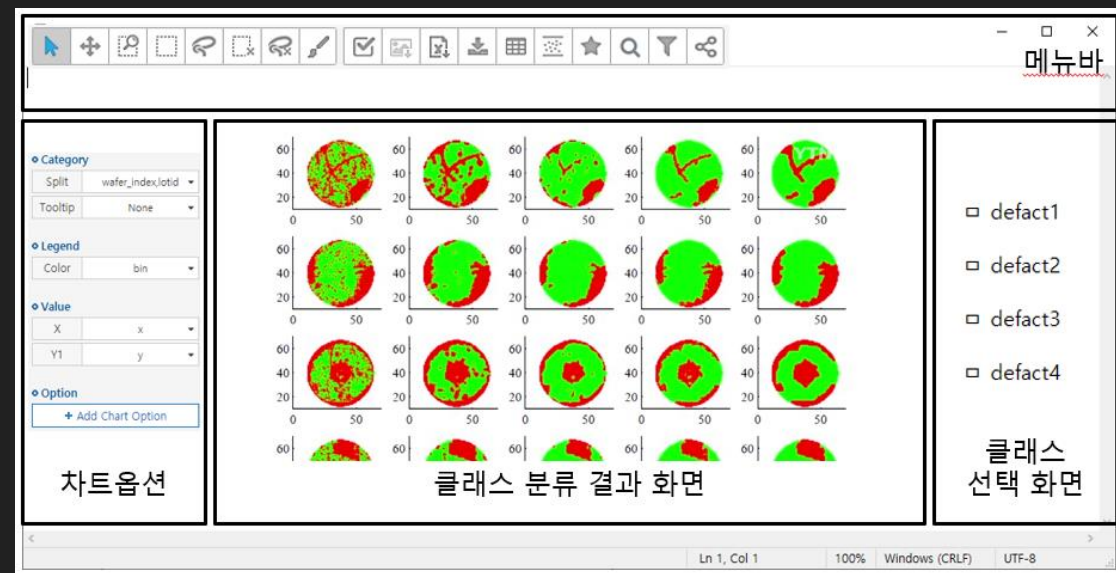


그림 3. 준지도 학습 배포 모델 예측 결과 화면 예

감사합니다!