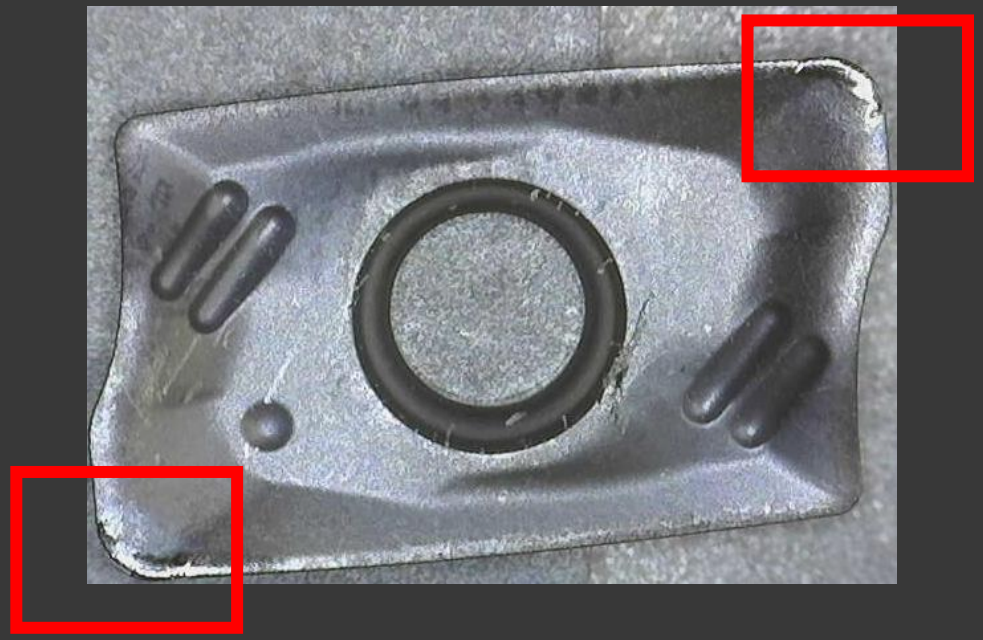
3월 20일 meeting

공구 마모 이미지를 활용한 마모 정량화

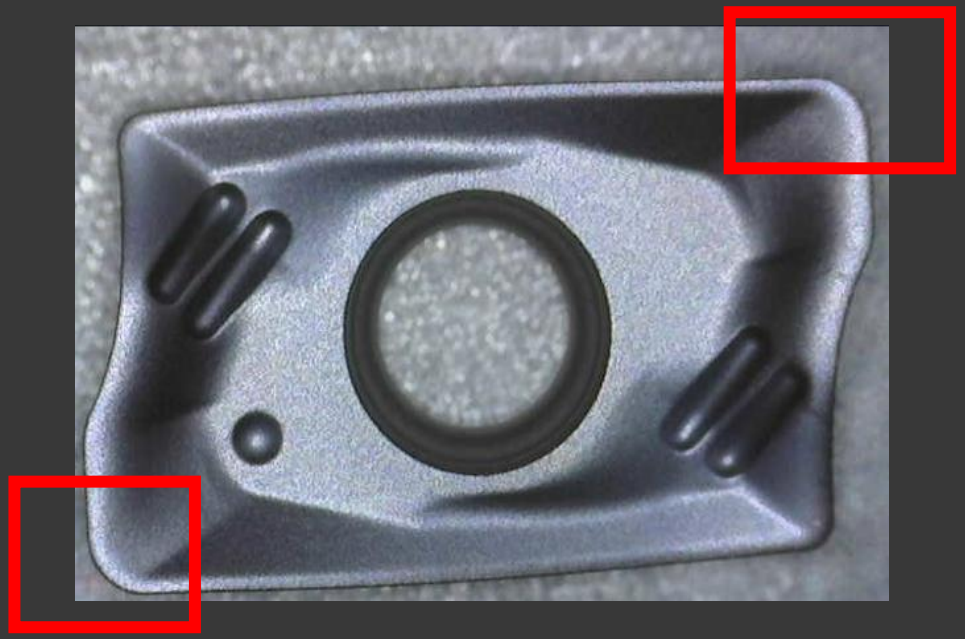
조민준

CONTENTS

데이터 수집

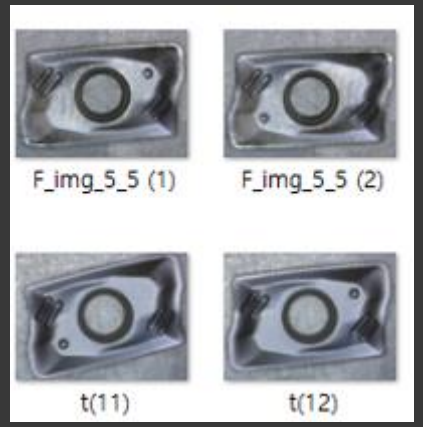


- 마모된 tool



- 정상 tool

데이터 수집




```
def label_func(fname):
  if fname[0].isupper():
    return 'fail'
  else:
    return 'normal'
```

- 데이터 총 106개
- 마모 데이터 : 84개
- 정상 데이터 : 22개
- 정상 데이터는 소문자 f, 마모 데이터는 대문자F로 라벨링후 사용자 정의 함수로 를통해 데이터를 분류하였음.

```
dls = ImageDataLoaders.from_name_func(path,files,label_func,₩
                                      item_tfms=Resize(224))
dls.show_batch(max_n=12)
```

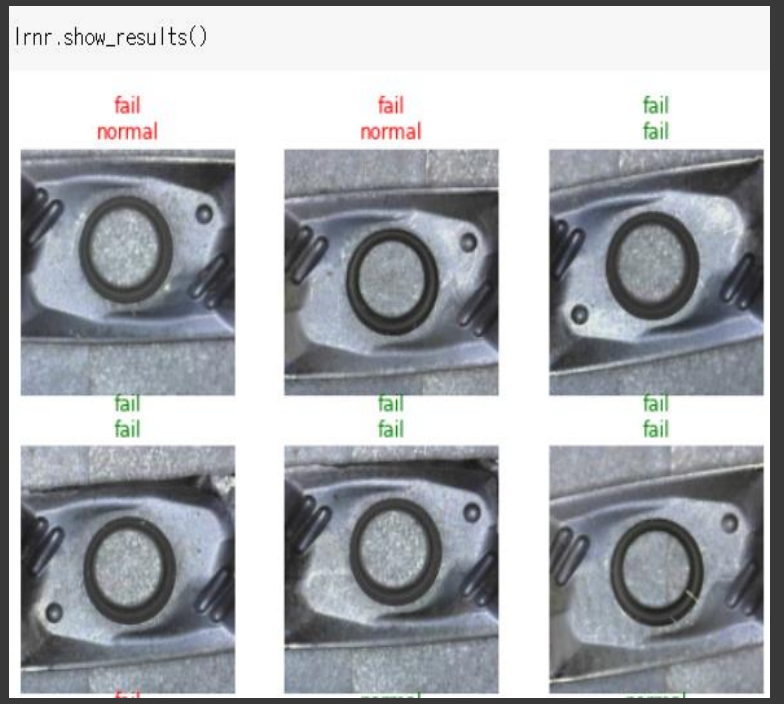
- ImageDataLoaders.from_name_func 를 사용하여 이미지 Resize하여 데이터셋을 생성하고 미니배치로 만들어 줌.

CNN Pytorch- Fastai를 통한 CNN 구현

```
lrnr = vision_learner(dls,resnet34,metrics=accuracy)
lrnr.fine_tune(6)
```

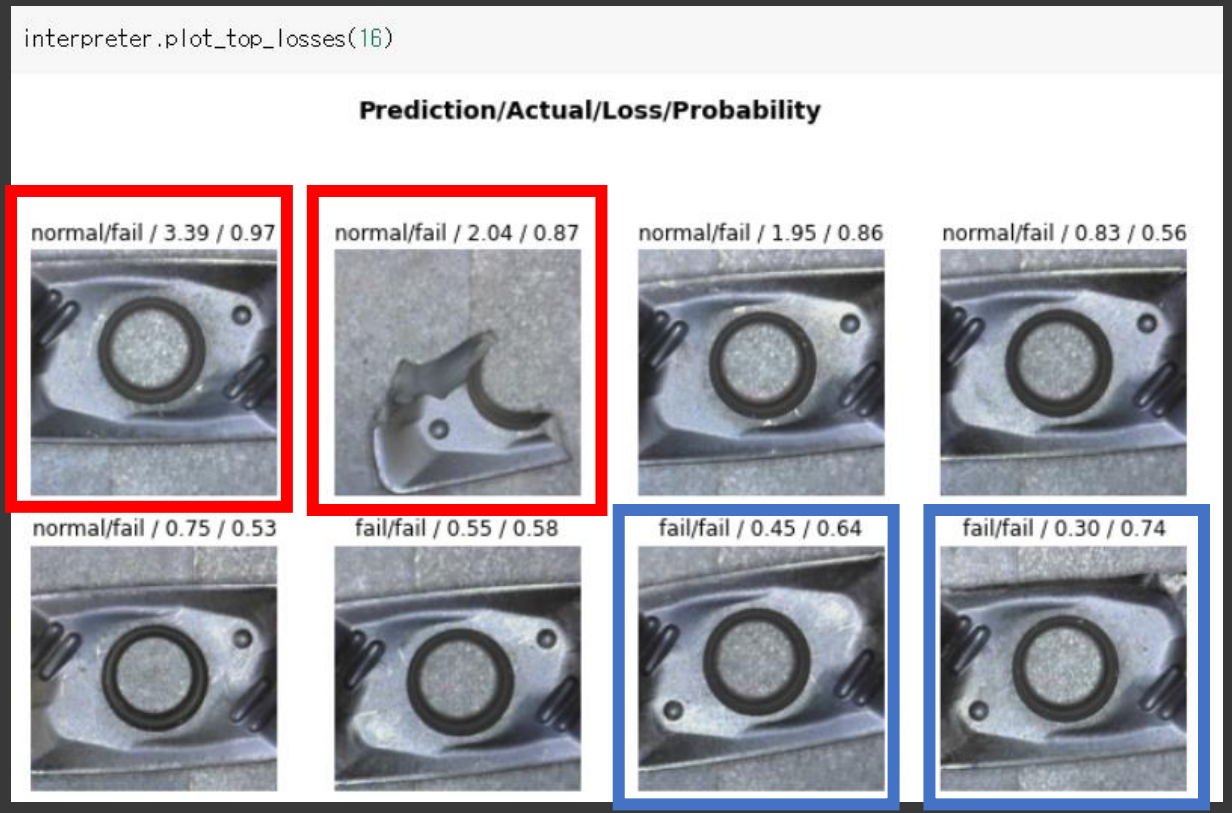
epoch	train_loss	valid_loss	accuracy	time
0	1.171375	1.860622	0.761905	00:01
epoch	train_loss	valid_loss	accuracy	time
0	0.742362	0.816778	0.761905	00:02
1	0.749170	0.118727	0.952381	00:02
2	0.553508	0.288723	0.857143	00:01
3	0.475548	0.473058	0.761905	00:01
4	0.423640	0.528403	0.809524	00:01
5	0.369312	0.531042	0.761905	00:01

```
lrnr.show_results()
```



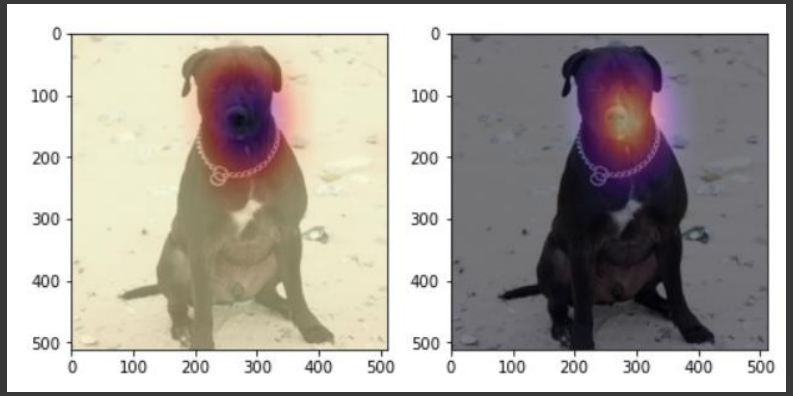
- Fastai에 구현되어 있는 resnet34 (이진분류)모델을 사용하여 분류 모델을 생성하여 학습.

CNN -



- 모델의 결과값을 분석하는 interpreter 생성하여 결과를 분석.
- 정상적으로 분류한 결과를 살펴보면 Probability가 50%~70% 로 높지 않는 것으로 확인
- 잘못 분류한 결과를 살펴보면 Probability가 80%~90%로 높은 경우 확률을 가짐.
- 또한 공구가 완전이 파손된 건 제대로 예측 못하는것으로 보여짐

앞으로의 연구방향



1. 완전히 파손된 공구에 대해서는 분류를 전혀 못하고 있기에 구도를 달리하여 데이터 재수집 할 예정이며 또한 지금 있는 데이터가 같은날에 데이터를 수집한것이 아니기에 다시 한번 데이터를 수집할 예정.

2. 현재 이미지의 노이즈를 제거하지 않고 모델에 학습시켰지만 노이즈를 제거하는 방식인 autoencoder 방식을 통해 이미지의 노이즈를 제거하고 모델에 학습시킬 예정.

3. 현재는 resnet모델을 사용하여 학습시키는 중이지만 yolov5, googlenet등 이미 성능이 증명된 뛰어난 모델을 사용하여 학습시킬 예정.

4. 딥러닝 모델이 이미지를 분류할 때 어떤 부분을 주로 고려하는지 이해하는데 도움을 주는 알고리즘인 CAM을 통해 결과를 분석할 예정

감사합니다.