
효율적 선체 균열 탐지를 위한 데이터 증강 기법과 도메인 적응 방법 간 최적의 적용 방안 연구

A66050 여윤기

Contents

1. 연구 배경 및 필요성
2. 이론적 배경
3. 데이터 준비 및 전처리
4. Baseline 모델 선정 및 학습
5. Domain Adaptation 적용결과
6. Data Augmentation 적용결과
7. 동시 적용 결과
8. 해석 및 결론

1. 연구 배경 및 필요성

- 오늘날 운항하는 선박은 지속적인 승선인원의 감소 등으로 인해 최소한의 선원 또는 무인 선박으로 운항중에 있으며, 현장에서 선체 표면 균열 탐지와 진단은 법적으로 규정된 정기적인 수리기간 중 상가를 통해 검사원의 육안 검사가 주를 이루고 있다.
- 따라서 인공지능을 활용한 선체 균열을 탐지하는 것은 인구 감소 등으로 인한 운항 무인화 시대에 선박의 안전운항을 위해 매우 중요함.
- 하지만 공개된 선체 표면 균열 탐지를 위한 이미지 데이터셋 구축은 매우 부족한 것이 현실



선체 표면 균열 이미지를 거의 사용하지않고,
이를 효과적으로 탐지/분류할수있는 모델 개발 필요성

2. 이론적 배경

- 딥러닝 기반 이미지 분류 기법 : CNN(VGG, ResNet, MobileNet 등..)
- CAM(Class Activation Mapping)
- 도메인 적응(Domain Adaptation)
- 데이터 증강(Data Augmentation)

각 이론적 배경들에 대해 원문 참고하여 서술

3. 데이터 준비 및 전처리

- **Source Domain Dataset(TRAIN)**

부족한 선체 표면 균열 이미지 데이터셋 대신, 대중적으로 널리 공개되어 있고 접근이 쉬운 깃허브(Github) 및 캐글(Kaggle)의 콘크리트 표면 균열 이미지를 4가지 종류의 데이터셋을 Train Domain으로 활용

No. 1 : (https://github.com/khanhha/crack_segmentation#Dataset)

No. 2 : (https://digitalcommons.usu.edu/all_datasets/48/)

No. 3 : (<https://www.kaggle.com/datasets/arunrk7/surface-crack-detection>)

No. 4 : (<https://www.kaggle.com/datasets/pauldavid22/crack50020220509t090436z001>)

- **Target Domain Dataset(TEST)**

최종 연구 목표인 선체 표면 균열 탐지를 위해, AI허브에서 제공하는 "선박 도장 품질 데이터" 일부 활용

(<https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=data&dataSetSn=71447>)

3. 데이터 준비 및 전처리

- Source Domain Dataset

구분	특성
No. 1 (github)	가로*세로 448*448 pixel 포장도로 및 콘크리트 구조물의 표면 균열 이미지 및 정상 이미지 약 11,000장 용량 약 2GB
No. 2 (UtahState University)	가로*세로 256*256 pixel 교량갑판, 포장도로, 벽 표면 균열 이미지 및 정상 이미지 56,000장 용량 약 530MB
No. 3 (Kaggle)	가로*세로 227*227 pixel 다양한 콘크리트 표면의 균열 이미지 및 정상 이미지 50,000장 용량 약 500MB
No. 4 (Kaggle)	가로*세로 2000*1500 pixel 다양한 종류의 표면 균열 이미지 및 정상 이미지 3,300장 용량 약 5.7GB

- Target Domain Dataset

구분	특성
선박 도장 품질 데이터 (AI 허브)	선박 도장 표면의 균열 이미지 및 정상 이미지 약 1,000장 용량 약 200MB

3. 데이터 준비 및 전처리

- Source Domain Dataset(Sample)

Crack
(Sample)



Non-Crack
(Sample)

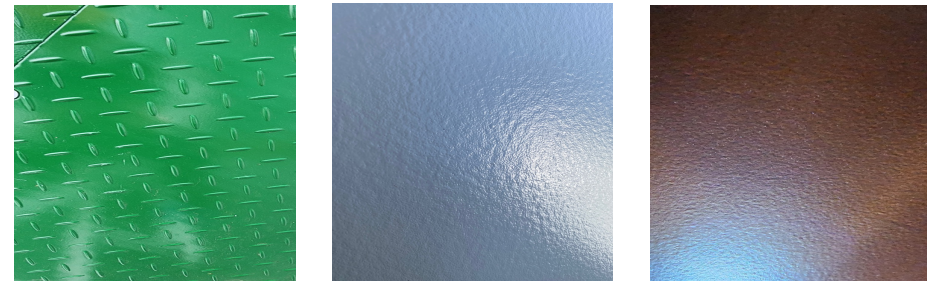


- Target Domain Dataset(Sample)

Crack
(Sample)



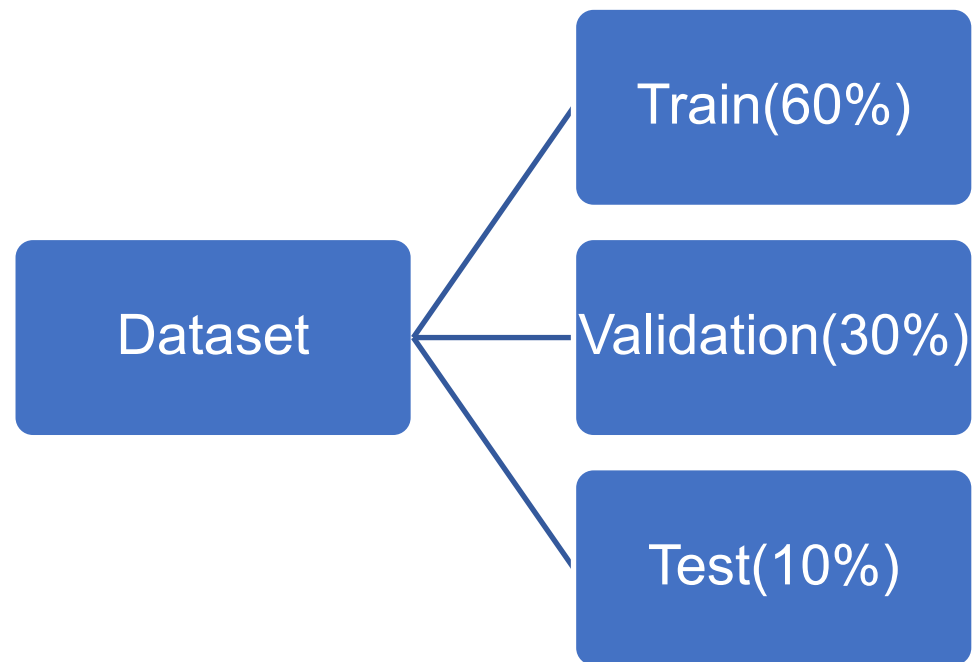
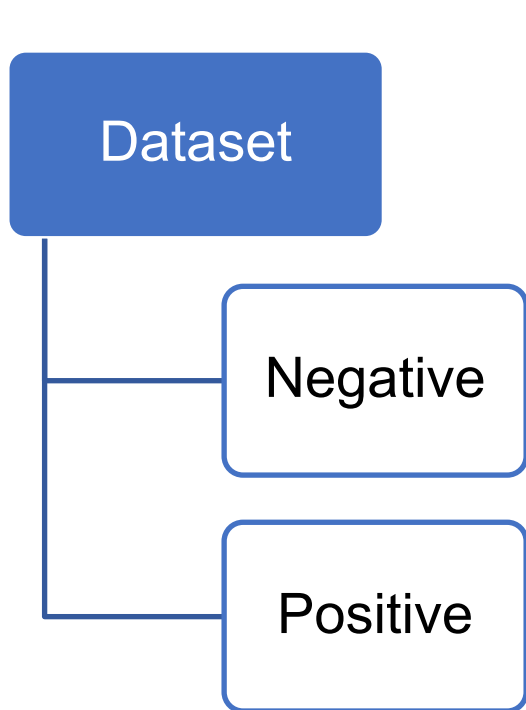
Non-Crack
(Sample)



3. 데이터 준비 및 전처리

- 데이터 전처리

1. 노이즈 제거 : 이미지에서 불필요한 노이즈를 제거하고 명암 대비, 선명도 등을 조절하여 균열 부분이 더 명확하게 드러나도록 전처리 수행
2. 라벨링 및 데이터셋 분할 : 정상 이미지(Negative_0)과 균열 이미지(Positive_1)을 라벨링하여 구분하고 전체 데이터셋을 Train / Validation / Test 데이터셋으로 6:3:1 비율로 분할.



4. Baseline 모델 학습 및 선정

- Baseline 모델 아키텍처 비교 및 선택

1. **VGG 모델** : 컨볼루션 필터를 사용하는 여러개의 layer를 단순히 반복해서 쌓아올린 간단한 CNN 모델.
2. **ResNet 모델** : ResNet 모델 특성상 더욱 깊은 layer를 쌓아올려 높은 수준의 특징 추출 및 성능 향상
3. **Asemble 모델(VGG + ResNet)** : 두 모델을 혼합한 Asemble 모델 활용

⇒ Source Domain Dataset(No. 1~4) 및 Target Domain Dataset에서의 모델학습 결과를 비교하여 가장 좋은 결과를 나타내는 모델을 Baseline 모델으로 선정

- 모델 컴파일 및 훈련

1. **Hyperparameter 설정** : 검증 데이터셋에 대한 모델 평가를 통해 손실 및 정확도 측정.

- loss Function : binary_crossentropy
- learning rate = 0.001 / batch_size = 32 / epoch = 50
- optimizer = 'adam'

2. **평가지표** : 학습 결과로 출력되는 accuracy_score, Confusion Matrix 및 성능 지표(Precision, Recall, F1-score) 활용

4. Baseline 모델 학습 및 선정

- Baseline Model 학습 평가결과 비교(Accuracy_Score(%))

Model	Dataset No.1 (%)	Dataset No.2 (%)	Dataset No.3 (%)	Dataset No.4 (%)	Combined Dataset (%)	Ship Dataset (%)
VGG16	74.3	78.2	76.9	74.5	72.4	54.7
VGG19	77.7	81.8	82.8	76.1	78.9	56.6
ResNet50	97.5	99.2	98.3	98.5	99.1	57.9
ResNet101	99.2	99.3	98.7	98.8	99.6	57.7
ResNet152	98.8	99.2	98.4	98.6	98.3	58.5
ResNet50 + VGG16	99.4	99.3	99.6	99.2	99.3	61.4

ResNet50+VGGNet16(64*64) Classification Report:

	precision	recall	f1-score	support
NEGATIVE	1.00	1.00	1.00	3978
POSITIVE	1.00	1.00	1.00	4022
accuracy			1.00	8000
macro avg	1.00	1.00	1.00	8000
weighted avg	1.00	1.00	1.00	8000

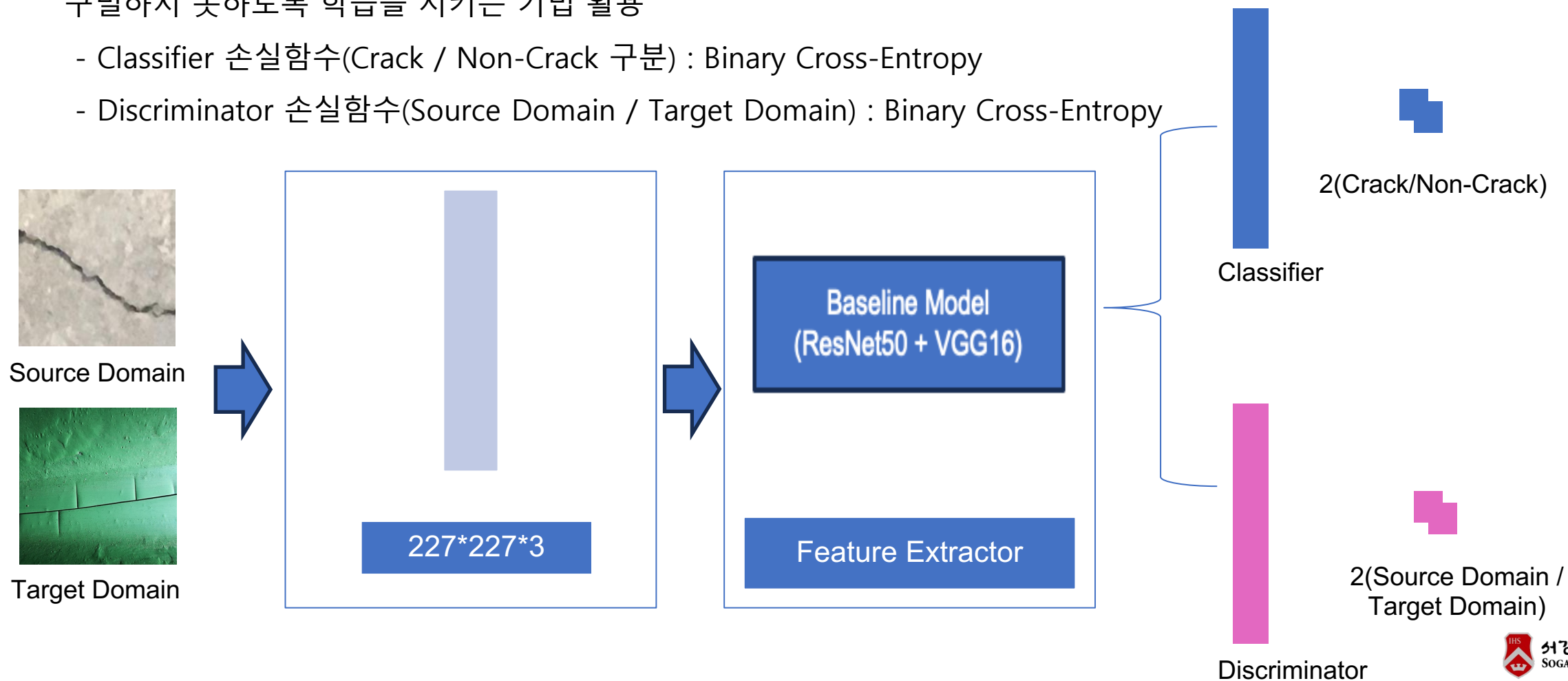
- Source Domain Dataset으로 학습시킨 모델을 도메인 영역이 다른 Target Domain Dataset(선체 표면 균열 데이터)으로 평가하였을 때, 분류 정확도 차이가 크게 발생함. 이를 극복할 수 있는 Data augmentation, Domain Adaptation 적용의 필요성 대두됨.
- Source Domain Dataset(No.1 ~ 4) 에서 ResNet50과 VGG16모델을 결합한 Asemble 모델이 최소 99.2%이상의 높은 정확도를 기록 -> Data Augmentation 및 Domain Adaptation 적용결과 비교 간 활용할 Baseline 모델로 선정

5. Domain Adaptation 적용결과

• Method

1. DANN(Domain-Adversarial Training of Neural Networks) : 신경망 모델이 Source Domain과 Target Domain을 구별하지 못하도록 학습을 시키는 기법 활용

- Classifier 손실함수(Crack / Non-Crack 구분) : Binary Cross-Entropy
- Discriminator 손실함수(Source Domain / Target Domain) : Binary Cross-Entropy



5. Domain Adaptation 적용결과

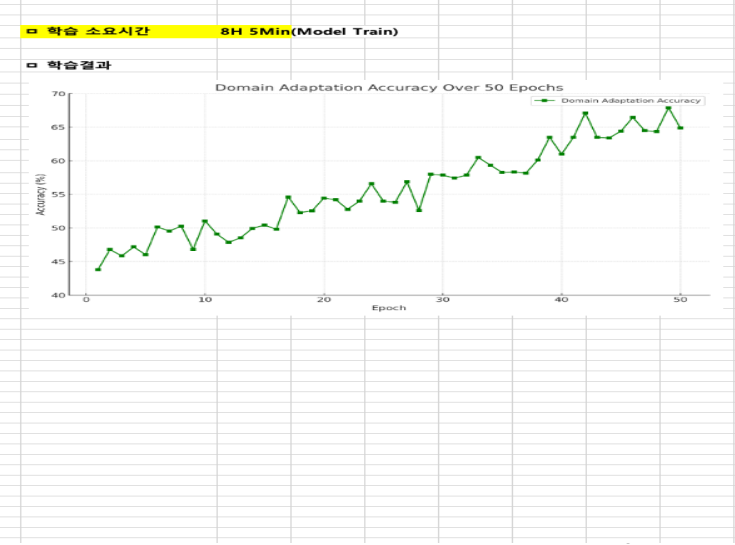
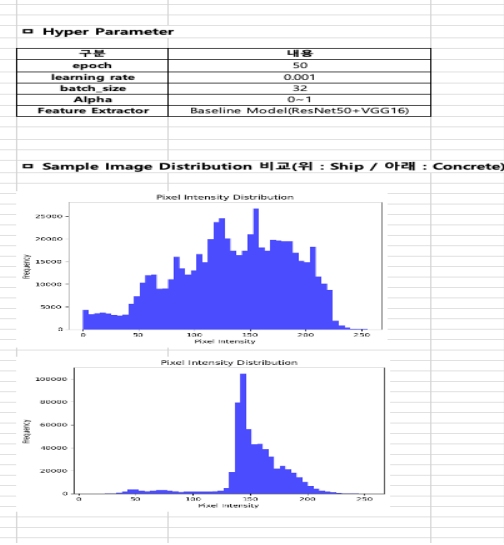
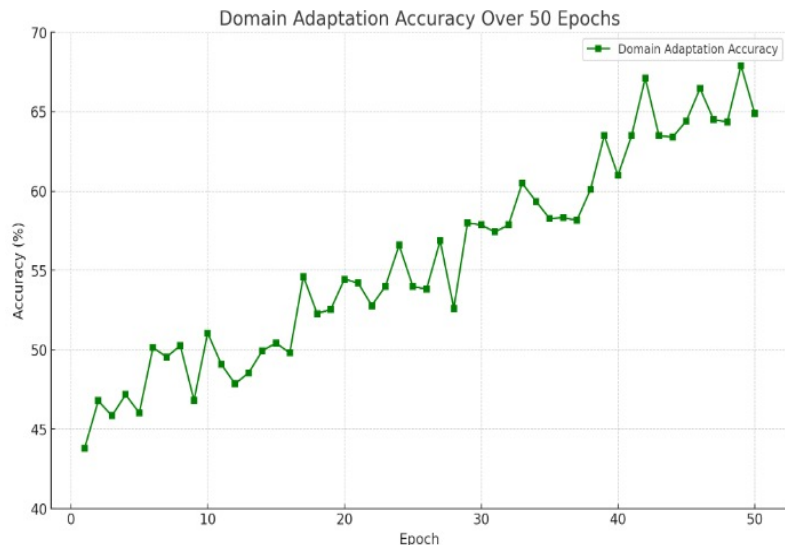
- 모델 컴파일 및 훈련

1. Hyperparameter 설정 : epoch=50, learning_rate=0.001, batch_size=32, Alpha=0~1, Feature Extractor/Discriminator=Baseline Model(ResNet50+VGG16),
loss function=binary_cross_entropy

- 모델 평가지표

1. Accuracy / 2. 학습 소요시간 / 3. Precision, Recall, F1 Score

⇒ 위 지표를 기준으로 하였을때 Baseline 모델에서 어떤 기법도 적용하지 않았을때보다 Target Domain에서의 성능이 약 5.3% 가량 향상됨.(**Target Domain Accuracy : 66.7%**)



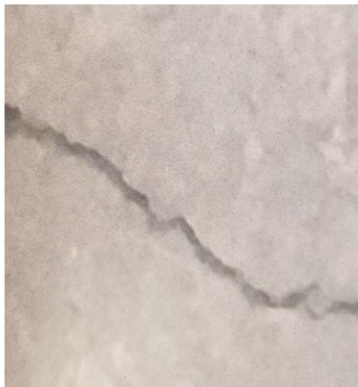
6. Data Augmentation 적용결과

- Method

1. Basic Data Augmentation : Keras 제공 ImageDataGenerator 라이브러리 활용, Source Domain Dataset에 특정 확률로 Rotate / Flip / Zoom / Brightness 등의 단순한 효과 적용
2. Mixup : Source Domain Dataset 내에서 각각의 학습용 데이터를 설정한 비율로 mixup (설정비율 1:1)
3. Cutout : Source Domain Dataset 내 각각의 이미지 일부를 설정 비율만큼 도려낸 후 학습
4. Auto Augmentation : 자주 사용되는 Data augmentation 기법들 중 최적의 조합을 찾기 위해 강화학습 기반의 PPO(Proximal Policy Optimization)로 학습
5. Cutmix : 이미지에서 제거된 영역을 다른 이미지로부터 가져온 Patch로 적용하여 학습



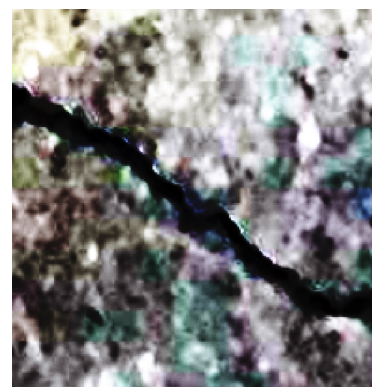
Basic Data
Augmentation



mixup



cutout



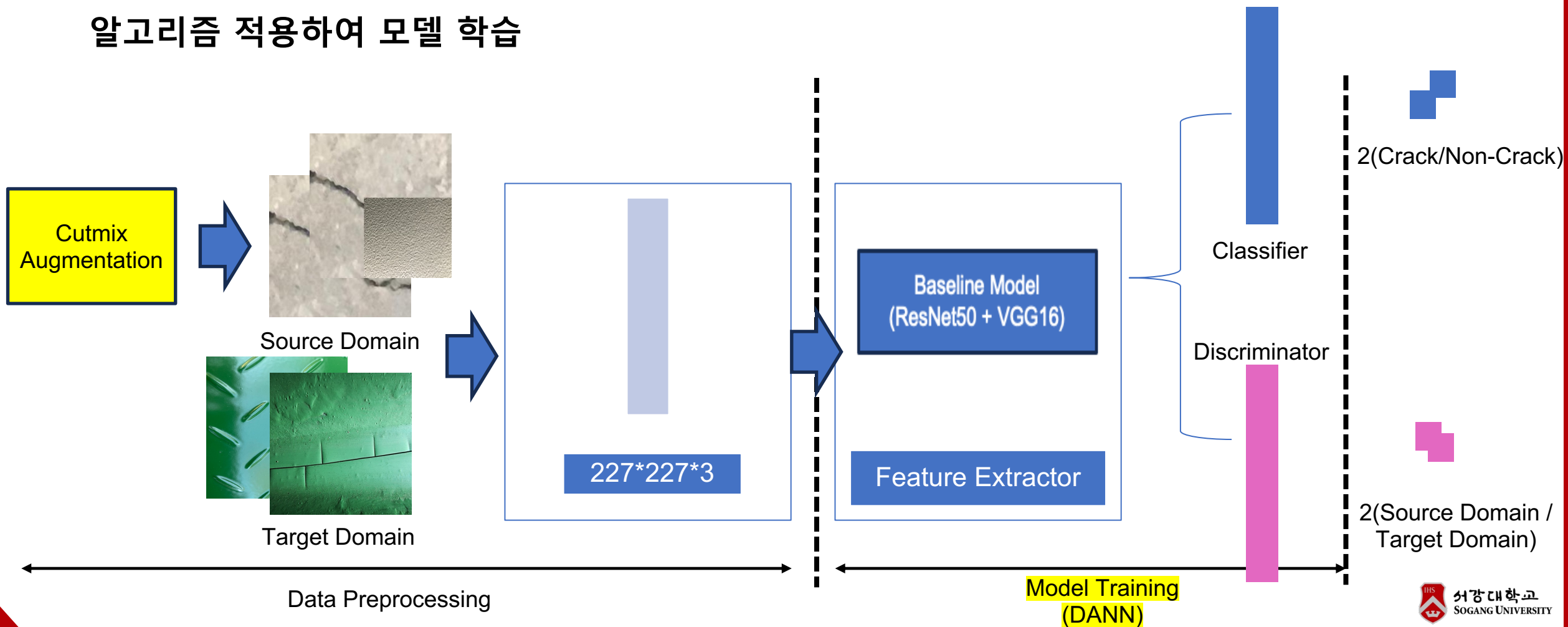
Auto
Augmentation



cutmix

7. 동시 적용 결과

- Method : 데이터 전처리 과정에서 Data Augmentation 기법 중 가장 성능이 좋게 나타난 Cutmix를 적용하여 Source Domain Dataset 증강 후, Baseline 모델 학습 과정에서 DANN 알고리즘 적용하여 모델 학습



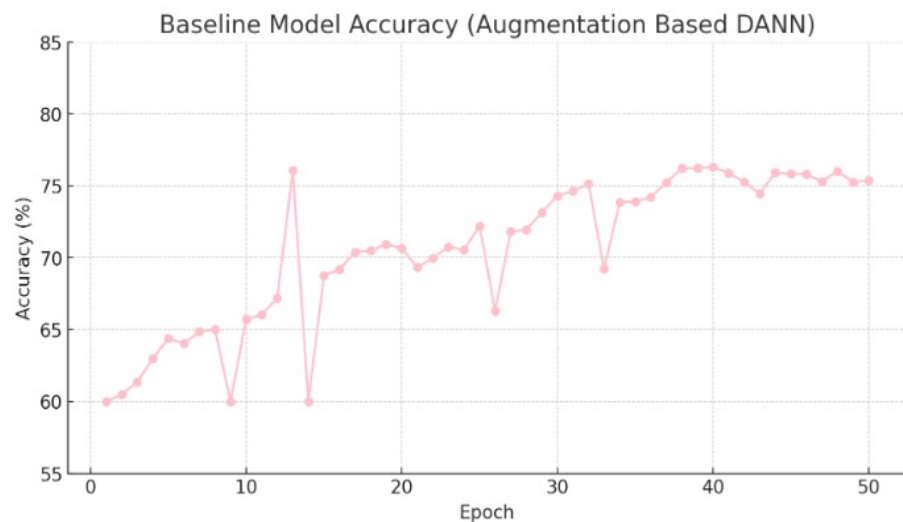
7. 동시 적용 결과

- 평가지표

1. Accuracy
2. 학습 소요시간
3. F1 Score(Precision, Recall)

- 평가결과(Target Domain Accuracy : 76.3%)

1. Source Domain으로만 훈련시켰을 때보다 **정확도 약 14.9% 향상**
2. Domain Adaptation(DANN)만 적용하였을때보다 **정확도 약 9.6% 향상**
3. Data Augmentation(Cutmix)만 적용하였을때보다 **정확도 약 2.3% 향상**



True Labels	Predicted Labels	
	Normal	Crack
Normal	382	118
Crack	119	381

8. 해석 및 결론

Method		GPU Runtime	F1 Score	Target Domain Accuracy(%)
None		8H 3min	0.62	61.4
Domain Adaptation	DANN	8H 5min	0.69	66.7
Data Augmentation	Basic	8H 40min	0.682	68.3
	Cutout	8H 42min	0.72	72.0
	mixup	8H 40min	0.735	73.0
	Auto Augmentation	10H 37min	0.745	74.5
	Cutmix	8H 38min	0.74	74.0
Domain Adaptation(DANN) + Data Augmentation(Cutmix)		8H 48min	0.79	76.3

8. 해석 및 결론

• 해석 및 결론

1. 학습을 위한 Target Domain의 데이터의 양이 충분하지 않을 경우, 유사한 Source Domain Dataset을 충분히 확보한 후 적절한 Data Augmentation 또는 Domain Adaptation 기법을 활용하여 모델 학습을 시키는 것이 Target Domain에서의 모델 성능 향상에 있어 효과적이다.
2. 단일 기법만 적용한다고 하였을 때, 증강된 데이터로 인해 학습 소요시간은 약 40분 정도 더 소요되나 모델 학습 과정에서의 Domain Adaptation 보다 데이터 전처리 과정에서의 Data Augmentation 적용이 Target Domain에서 약 7.3% 가량 모델성능 향상에 효과적이었고, 비교한 5가지 Data Augmentation 기법 중에서는 Cutmix 증강 기법이 성능향상에 가장 효과적이었다.
3. Data Augmentation + Domain Adaptation 동시적용 시, 각각의 기법을 적용하였을때 보다 Target Domain에서의 모델 성능을 약 2.3 ~ 9.6%까지 향상시키는 시너지 효과를 확인할 수 있었다.

8. 해석 및 결론

- 한계점

1. 보다 다양한 Domain Adaptation 및 Data Augmentation 기법들에 대한 비교 연구 수행이 필요함
2. 실험 환경을 보다 다양하게 구성하고, 실험 환경과 부합되는 각각의 평가 체계를 마련한 후
성능별 비교를 통해 실제 자원 소모량/연산량 등을 비교하는 등 더욱 구체적인 성능평가가 필요해 보임