

# 딥러닝을 활용한 시공현장 하자 감지 및 공종별 분류 관리 시스템

2014100518 송기훈

## Automotive management of defects in construction phase using deep learning

### 요 약

현재 건설 시공현장은 하자를 직접 점검 후에 사진을 찍어 라벨링한 후, 사내 시스템에 보고하는 복잡한 시스템을 가지고 있다. 이러한 업무의 중복성 및 비효율성을 제거하기 위해 본 연구에서는 딥러닝을 활용한 시공현장 하자 감지 및 공종별 분류 관리 시스템을 제시한다. AlexNet, VGG16, VGG19, MobileNets, Inception V3 등의 5개 모델을 대상으로 총 2,290장, 23클래스로 구성된 시공 현장 이미지를 학습하였다. 실험 결과, 뛰어난 성능을 보여주지는 못했지만 하자 점검을 위한 기준을 제시하고, 시공 과정 및 업무의 효율화를 위한 아이디어를 제시한 것에 본 연구의 의의가 있다. 향후 개선 방안을 통해 정확도를 증가시켜 모델의 실현 가능성을 증가시킬 것으로 판단된다..

### 1. 서론

건설 시공은 설계도와 시방서에 따라 건설 구조물을 건설하는 일체의 활동을 말한다[1]. 일반적으로 건설 시공에서 하자란 공사상 잘못으로 인하여 균열, 침하, 파손, 들뜸, 누수 등이 발생하여 건축물 또는 시설물의 안전상, 기능상 또는 미관상의 지장을 초래할 정도의 결함을 의미한다. 크게 내력구조별 하자과 시설공사별 하자로 구분된다.

내력구조별 하자는 공동주택 구조체의 일부 또는 전부가 붕괴된 경우 또는 공동주택의 구조안전상 위험을 초래하거나, 그 위험을 초래할 우려가 있는 정도의 균열, 침하 등의 결함이 발생한 경우를 말한다. 시설공사별 하

자는 공사상의 잘못으로 인한 균열, 처짐, 비틀림, 들뜸, 침하, 파손, 붕괴, 누수, 누출, 탈락, 작동 또는 기능불량 등이 발생하여 건축물 또는 시설물의 안전상, 기능상 또는 미관상의 지장을 초래할 정도의 결함이 발생한 경우를 말한다[2]. 건설하자는 건설 과정에 있어서 안전, 공사기간, 작업지속여부 등에 영향을 주는 요인이기 때문에 신속히 하자를 발견하고 보수하는 일은 중요하다. 현재 건설현장은 대부분 관리자에 의한 하자 점검이 시행되며, 공정 사진 분류 및 하자 체크 항목이 분리되어 있어 업무의 중복이 일어나는 실정이다. 이러한 문제점을 해결하기 위한 아이디어를 딥러닝을 통해 실현하고자 한다. [Figure 1]은 하자 발생 시 처리 과정에 대한 절차

를 나타낸다.

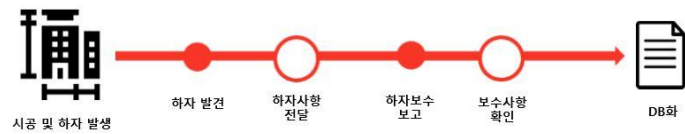


Figure 1 하자 발생 시 처리 과정

본 연구에서는 딥러닝 다중 분류(Multiple classification)를 통해 현장에서 촬영된 사진의 하자 유무 및 공종을 확인하고, 이에 따라 분류해주는 알고리즘을 도입하였다. 이는 현재 건설 현장의 비효율적인 현장 사진 분류 및 하자 점검을 기존보다 단시간 내에 이루어지도록 함으로써, 공사기간과 공사 비용의 감소를 통해 성공적인 시공의 토대를 마련할 수 있도록 한다. [Figure 2]은 본 연구의 목표를 나타낸다.

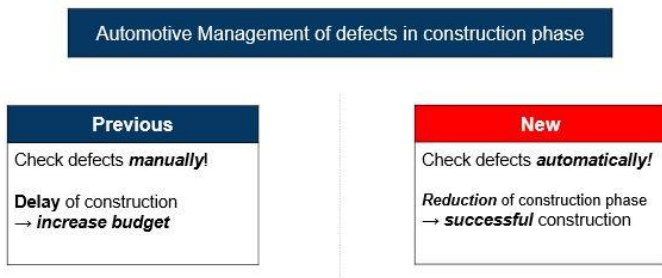


Figure 2 연구 목표

## 2. 관련 연구

### 2.1 Surface-Defect Detection

Surface-Defect Detection이란 제품 혹은 물품의 표면 부분에 생긴 균열을 딥러닝 모델을 통해 학습해 분류하는 기술이다. Domen Tabernik 등 4명은 2019년 6월, 해당 연구에 대해 KolektorSDD 데이터셋을 활용해 학습하여 99.9%의 평균 정확성(Average precision)을 달성했다 [3].

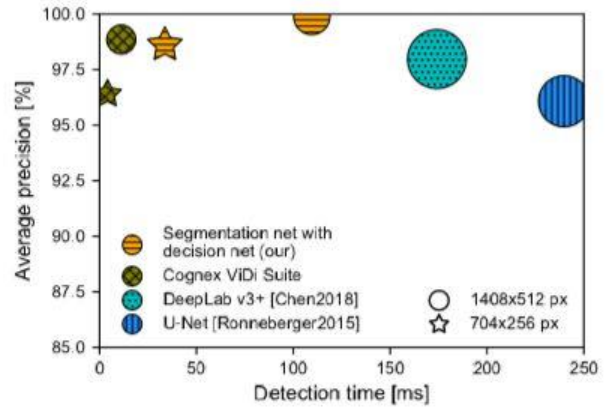


Figure 3 Surface-Defect Detection

### 2.2 Classification of Images from Construction Sites

건설현장에서 사진이란 시간이 흘러가며 변하는 현장의 시각적인 데이터를 남기는 가장 간단하고 확실한 방법이다. 현 상황은 시공자 수작업에 따라 이미지 데이터를 효율적으로 사용하지 못하고 있다. 이에 따라 해당 연구에서는 Xception, MobileNets, VGG16, VGG19 등의 딥러닝 모델을 사용해 공정 사진 분류 학습을 진행하였다. 평가지표로는 정밀도(precision), 재현율(recall), F1 점수(F1 score), 수치가 가장 높은 모델에 한해 정밀도-재현율 그래프(precision-recall graph)를 통해서 평균 정밀도(mAP: mean average precision)을 구해 정량적인 분류기의 성능을 수치화 하였다.

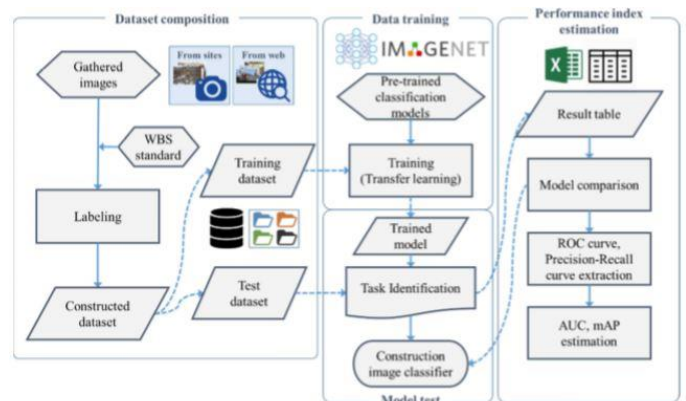


Figure 4 전체 연구 구성 및 방법

실험 결과, MobileNets 모델을 기반으로 전이 학습을 시킨 분류 모델이 정밀도 83.4%, 재현율 82.4%, F1 점수

85.8%를 보이며 가장 좋은 성능을 보였다. 해당 연구의 기여점은 딥러닝을 활용한 건설 현장 사진 분류의 첫 연구라는 점에 있다. 또한, 비교적 적은 양의 사진으로도 실제 활용 가능한 성능의 분류 모델을 학습하였다는 데 기여점을 가지고 있다.

### 3. 데이터셋 구축

#### 3.1 공종 분류 체계

국토교통부는 건설 데이터를 분류, 관리의 기준을 선정하고, 효율성 및 용이성을 증가시키기 위해 건설사업정보 운영지침으로 건설정보분류체계를 고시하였다(국토교통부 2001)[6]. 현재 최신 개정은 2015년에 이루어졌으며, 내용적으로는 크게 시설물 분류, 공간 분류, 부위 분류, 공종 분류, 자원 분류 등 5가지로 나뉜다. 본 연구에서는 세부 항목의 건축 공종으로 분류된 17개 공종 중 경계가 불분명한 건축물 부대공사(Subsidiary construction)를 제외한 16개 공종에 맞춰 데이터셋을 구축하려 한다.

데이터들은 서울특별시에서 제공하는 공공 건설 사진 “건설알리미”[7] 를 참조하였고, 추가적으로 인터넷에서 검색하여 확보하였다. 각 공종별 명칭 및 해당 1차 데이터셋은 아래 표에 도시되어 있다.

Table 1 1차 데이터셋

공종 분류	전체 data (장)	학습 data (장)	실험 data (장)
Temporary (가설공사)	92	70	22
Excavation (토공사)	133	100	33
Foundation (기초공사)	107	82	25
Reinforced Concrete (철근 콘크리트공사)	110	83	27
Steel framework (철골공사)	105	80	25
Masonry (조적공사)	105	80	25

Plastering (미장공사)	81	61	20
Waterproof (방수공사)	88	66	22
Carpentry (목공사)	98	74	24
Metal works (금속공사)	98	74	24
Roof (지붕 및 홀통공사)	101	76	25
Windows (창문 및 유리공사)	100	75	25
Tiling (타일 및 돌공사)	108	91	27
Finishing (수장공사)	92	70	22
Landscaping (조경공사)	88	66	22
Painting (도장공사)	89	67	22
계	1,595	1,205	390

#### 3.2 하자 분류 체계

건설공사에서 발생하는 하자의 유형은 공사 공종별로 분류 가능하며, 가장 많이 상용하는 공사 종류별 분류로서 크게 건축, 설비, 전기, 토목공사의 하자로 분류한다 [2]. 건축공사 하부의 세부 공사로는 조적공사, 내장목공사, 창호공사, 방수공사, 유리공사, 가구공사, 잡공사, 미장공사, 수장공사, 도장공사, 도배공사, 거실바닥공사, 타일공사 등이 있다.

Table 2 하자 분류 체계

구 분	공 종
건축공사	조적공사, 내장목공사, 창호공사, 방수공사, 유리공사, 가구공사, 잡공사
마감공사	미장공사, 수장공사, 도장공사, 도배공사, 거실바닥공사, 타일공사
설비공사	옥외급수위생관공, 난방/환기/공조조화공사, 급배수위생설비공사, 가스/소화설비공사
전기공사	전기/전력공사, 통신/신호설비공사
토목공사	대지조성공사, 조경공사

세부 13개 공사 항목 중 하자 발생율이 2% 이상이면 동시에 국토교통부의 건설정보분류체계 공종별 분류에 포함된 7개 항목들(조적공사, 미장공사, 방수공사,

유리공사와 창호공사(창문 및 유리공사), 타일공사, 수장공사, 도장공사)에 대한 현장 사진을 수집해 데이터셋을 확보하려 한다.

**Table 3 공종별 하자 유형**

공종	하자 종류
조적공사	균열
미장공사	균열, 탈락, 미경화, 미장면 불균질
방수공사	천정 누수, 옥상 바닥 배수 불량
창호공사	래핑지 불량, 연귀틈새, 찌힘, 코킹불량, 크리센트 불량, 피스커버 불량, 누수
타일공사	타일균열, 들뜸
수장공사	벽체의 미장면 균열
도장공사	바탕면의 곰보, 도장상태불량, 먼불량

**Table 4 하자 데이터셋**

공종별 하자 분류	전체 data (장)	학습 data (장)	실험 data (장)
Masonry (조적공사)	42	32	10
Plastering (미장공사)	49	37	12
Waterproof (방수공사)	119	90	29
Windows (창문 및 유리공사)	108	81	27
Tiling (타일 및 돌공사)	87	66	21
Finishing (수장공사)	34	26	8
Painting (도장공사)	46	35	11
계	485	367	118

### 3.3 최종 데이터셋

하자 종류별 데이터셋을 살펴보면, 조적공사, 미장공사, 수장공사, 도장공사에서는 충분한 개수의 데이터가 확보되지 못한 것을 알 수 있다. 따라서 본 연구에서는 2.1에서 언급한 “KolektorSDD” 데이터셋을 조적공사, 미장공사, 수장공사 균열 하자 부분에 70장씩 추가시켜 학습을 진행하였다. 균열이미지를 통한 학습이 이루어지는지

차후에 분석할 예정이다. 최종 데이터셋은 다음과 같다.

**Table 5 최종 데이터셋**

	공종 분류	전체 data (장)	학습 data (장)	실험 data (장)
1	Temporary (가설공사)	92	70	22
2	Excavation (토공사)	133	100	33
3	Foundation (기초공사)	107	82	25
4	Reinforced Concrete (철근 콘크리트공사)	110	83	27
5	Steel framework (철골공사)	105	80	25
6	Masonry (조적공사)	105	80	25
7	Masonry Defect (조적공사 하자)	112	84	28
8	Plastering (미장공사)	81	61	20
9	Plastering Defect (미장공사 하자)	119	90	29
10	Waterproof (방수공사)	88	66	22
11	Waterproof Defect (방수공사 하자)	119	90	29
12	Carpentry (목공사)	98	74	24
13	Metal works (금속공사)	98	74	24
14	Roof (지붕 및 흙통공사)	101	76	25
15	Windows (창문 및 유리공사)	100	75	25
16	Windows Defect (창문 및 유리공사 하자)	108	81	27
17	Tiling (타일 및 돌공사)	108	91	27
18	Tiling Defect (타일 및 돌공사 하자)	87	66	21
19	Finishing (수장공사)	92	70	22
20	Finishing Defect (수장공사 하자)	104	78	26
21	Landscaping (조경공사)	88	66	22

22	Painting (도장공사)	89	67	22
23	Painting Defect (도장공사 하자)	46	35	11
계		2,290	1,729	561

#### 4. 이미지 분류 모델 학습

##### 4.1 이미지 학습 및 성능평가

본 연구에서는 현장 사진을 기반으로 학습한 딥러닝 모델에서 새로운 현장 사진 데이터셋에 대한 전이 학습 성능을 볼 것이다. Tensorflow keras API를 활용해 AlexNet, VGG16, VGG19, MobileNets, Inception v3 등 총 5개의 모델들을 활용해 학습하고, 각 모델의 성능을 정확도(accuracy), 정밀도(precision), 재현율(recall), F1 점수(F1 score)로 수치화하여 표현하고 이를 비교 평가할 것이다. 학습에 사용된 중요 변수들은 다음과 같다.

Table 6 학습 변수

API	Tensorflow Keras
Batch size	128
Input shape	(224,224,3)
Result	Softmax
Loss function	Categorical_crossentropy(다중 분류)
Optimizer	Adam
Learning rate	0.01
Evaluation	Accuracy, precision, recall, F1 score

추가적으로 본 연구에서는 비교적 적은 데이터셋의 한 계로 인해 데이터 전처리 과정을 실시했다. Keras 내의 ImageDataGenerator 클래스를 통해 데이터를 변형시켜 새로운 학습 데이터를 만들어주었다. 이를 통해 데이터 부풀리기(Data Argumentation)가 가능하여 분류 모델 성능이 향상될 것으로 예측된다. 아래 그림은 실제 모델에 적용된 데이터 부풀리기 코드의 일부분이다.

```
ImageDataGenerator(rescale=1/255,
                   featurewise_std_normalization = True,
                   shear_range= 0.2,
                   zoom_range = 0.2,
                   rotation_range = 90)
```

Figure 5 데이터 부풀리기(Data Argumentation)

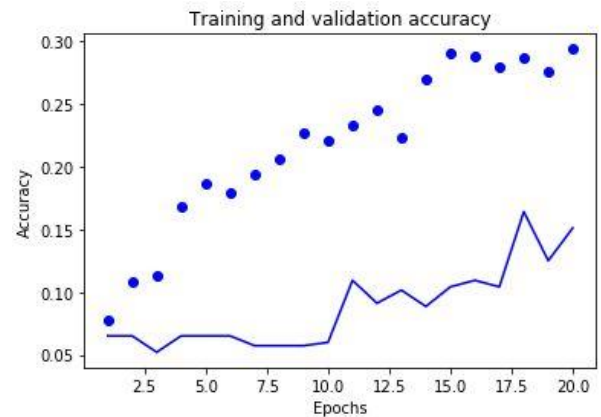
#### 4.2 공중별 분류

##### 4.2.1 AlexNet (20 epochs)

AlexNet은 2012년 개최된 ILSVRC(ImageNet Large Scale Visual Recognition Challenge) 대회의 우승을 차지한 CNN 구조다[8]. AlexNet은 5개의 Convolution 레이어와 3개의 FC 레이어로 구성되어 있고, 기본 입력이미지는 224\*224\*3 이미지다. 본 연구에서는 입력 값으로 이미지를 224\*224의 형태로 변환 후, 학습하였다.

공중별 분류에서 AlexNet은 정확도 15.10%, 정밀도 15.55%, 재현율 13.54%, F1 점수 14.47%를 기록했다. 아래의 그래프에서 점선은 Train accuracy, 실선은 Validation accuracy를 의미한다.

	Accuracy	Precision	Recall	F1 score
Train	29.37%	57.49%	11.85%	19.55%
Test	15.10%	15.55%	13.54%	14.47%

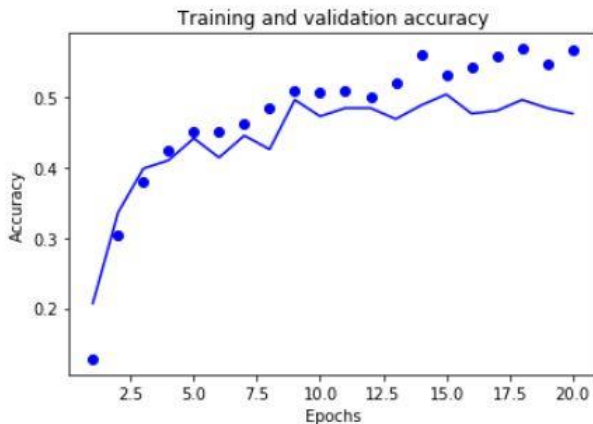


#### 4.2.2 VGG16 (20 epochs)

VGGNet은 2014년 ILSVRC에서 소개된 Oxford 대학교의 연구팀에 의해 개발된 모델로써, 2014년 이미지넷 (Imagenet) 인식 대회에서 준우승을 한 모델이다[9]. VGGNet 모델은 다양한 크기의 필터가 병렬적으로 배치되어있는 GoogleNet의 Inception 모듈에 비해 구조가 단순한 직렬배치로 되어 있기 때문에 실험적 모델로 보다 자주 활용된다. 16개의 레이어를 가진 모델은 VGG16, 19개의 레이어를 가진 모델은 VGG19로 불린다.

공종별 분류에서 VGG16은 정확도 47.66%, 정밀도 78.82%, 재현율 23.44%, F1 점수 36.09%를 기록했다. 아래의 그래프에서 점선은 Train accuracy, 실선은 Validation accuracy를 의미한다.

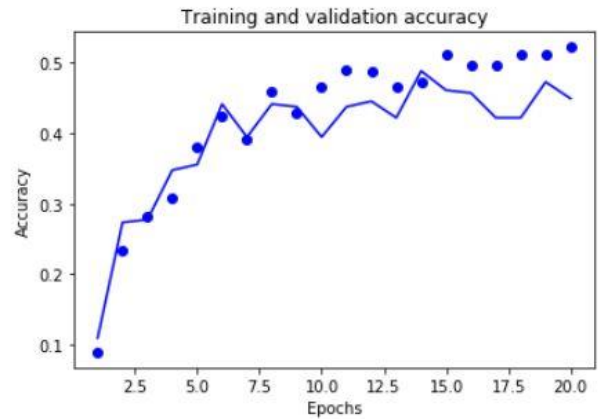
	Accuracy	Precision	Recall	F1 score
Train	56.63%	82.26%	30.05%	43.99%
Test	47.66%	78.82%	23.44%	36.09%



#### 4.2.3 VGG19 (20 epochs)

공종별 분류에서 VGG19은 정확도 44.92%, 정밀도 75.61%, 재현율 24.22%, F1 점수 36.69%를 기록했다. 아래의 그래프에서 점선은 Train accuracy, 실선은 Validation accuracy를 의미한다.

	Accuracy	Precision	Recall	F1 score
Train	52.21%	82.89%	19.69%	31.70%
Test	44.92%	75.61%	24.22%	36.69%

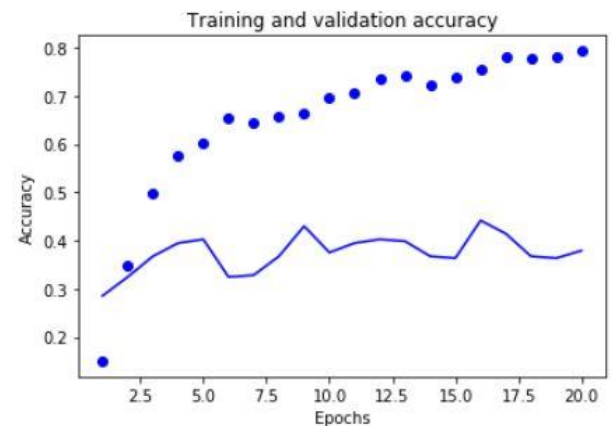


#### 4.2.4 MobileNets (20 epochs)

Google에 의해 제안된 MobileNets은 휴대용 기기에서 어플리케이션 형태로 변형되어 적용 가능하도록 개발된 모델이다[10]. MobileNets은 휴대용 기기에 맞춰 계산량을 줄이고 빠른 속도를 내는 것에 초점이 맞춰진 모델이다. 입력 값은 224\*224 크기의 이미지를 사용하였다.

공종별 분류에서 MobileNets은 정확도 37.89%, 정밀도 38.36%, 재현율 33.59%, F1 점수 35.82%를 기록했다. 아래의 그래프에서 점선은 Train accuracy, 실선은 Validation accuracy를 의미한다.

	Accuracy	Precision	Recall	F1 score
Train	79.28%	85.64%	72.73%	78.64%
Test	37.89%	38.36%	33.59%	35.82%



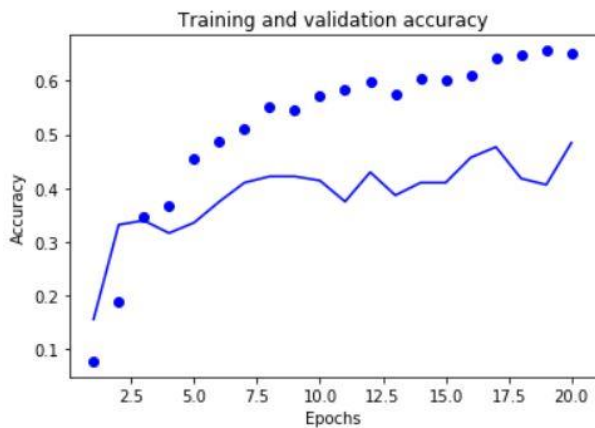
#### 4.2.5 Inception V3

GoogleNet은 ILSVRC 2014에서 우승을 차지한 모델이다 [11]. ILSVRC 대회 이후 Inception module을 사용하여,

Inception V1, Inception V3, Inception V4 등의 모델을 생성했다. 특히 Inception V3는 초기의 Inception 모듈에서 매개변수를 줄이고 실험을 통해 도출한 더 효율적인 모듈을 구성한 모델이다.

공중별 분류에서 Inception V3은 정확도 48.44%, 정밀도 51.94%, 재현율 46.48%, F1 점수 49.06%를 기록했다. 아래의 그래프에서 점선은 Train accuracy, 실선은 Validation accuracy를 의미한다. 아래의 그래프에서 점선은 Train accuracy, 실선은 Validation accuracy를 의미한다.

	Accuracy	Precision	Recall	F1 score
Train	64.92%	79.09%	53.50%	63.75%
Test	48.44%	51.94%	46.48%	49.06%

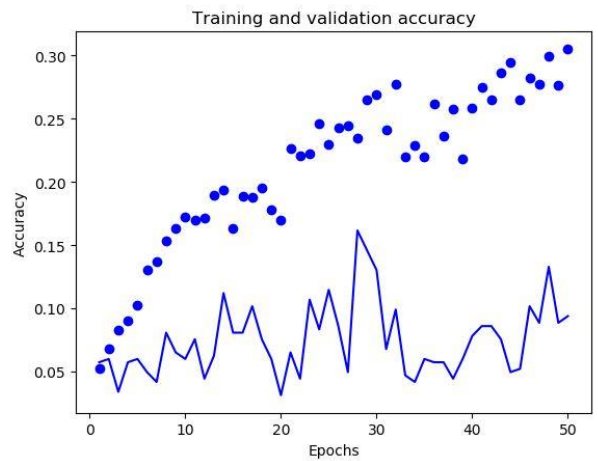


### 4.3 하자 점검 및 공중별 분류

#### 4.3.1 AlexNet (50 epochs)

하자 점검 및 공중별 분류에서 AlexNet은 정확도 9.38%, 정밀도 12.54%, 재현율 8.33%, F1 점수 10.01%를 기록했다. 아래의 그래프에서 점선은 Train accuracy, 실선은 Validation accuracy를 의미한다.

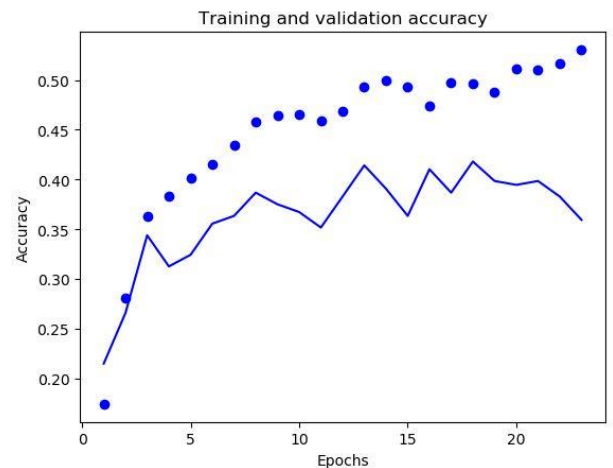
	Accuracy	Precision	Recall	F1 score
Train	30.52%	60.49%	9.08%	15.73%
Test	9.38%	12.54%	8.33%	10.01%



#### 4.3.2 VGG16

하자 점검 및 공중별 분류에서 VGG16은 정확도 35.94%, 정밀도 60.33%, 재현율 18.36%, F1 점수 28.15%를 기록했다. 아래의 그래프에서 점선은 Train accuracy, 실선은 Validation accuracy를 의미한다.

	Accuracy	Precision	Recall	F1 score
Train	53.05%	80.24%	25.04%	38.06%
Test	35.94%	60.33%	18.36%	28.15%



#### 4.3.3 VGG19

하자 점검 및 공중별 분류에서 VGG19은 정확도 32.03%, 정밀도 66.06%, 재현율 7.81%, F1 점수 13.95%를 기록했다. 아래의 그래프에서 점선은 Train accuracy, 실선은 Validation accuracy를 의미한다.

	Accuracy	Precision	Recall	F1 score
Train	40.23%	80.99%	7.11%	13.02%

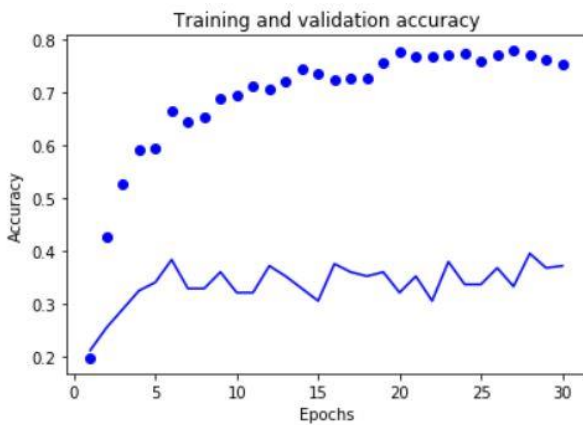
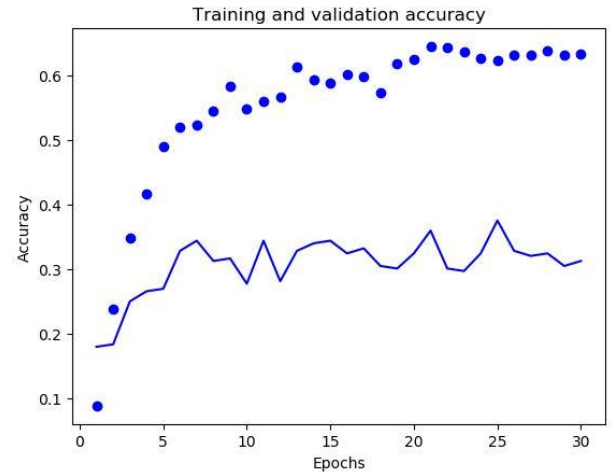


Test 32.03% 66.06% 7.81% 13.95%

#### 4.3.4 MobileNets

하자 점검 및 공종별 분류에서 MobileNets은 정확도 37.11%, 정밀도 41.48%, 재현율 34.77%, F1 점수 37.82%를 기록했다. 아래의 그래프에서 점선은 Train accuracy, 실선은 Validation accuracy을 의미한다.

	Accuracy	Precision	Recall	F1 score
Train	75.45%	78.34%	71.24%	74.60%
Test	37.11%	41.48%	34.77%	37.82%



#### 4.3.5 Inception V3

하자 점검 및 공종별 분류에서 Inception V3은 정확도 31.25%, 정밀도 33.77%, 재현율 29.30%, F1 점수 31.38%를 기록했다. 아래의 그래프에서 점선은 Train accuracy, 실선은 Validation accuracy을 의미한다.

	Accuracy	Precision	Recall	F1 score
Train	63.22%	72.84%	55.47%	62.94%
Test	31.25%	33.77%	29.30%	31.38%

### 5. 분류 성능 비교

#### 5.1 모델별 분류 성능 비교

##### (1) 공종별 분류

AlexNet, VGG16, VGG19, MobileNets, Inception-V3의 모델을 사용해 총 1,595장(Training set: 1,205장, Test set: 390장)의 건설 현장 사진을 5가지 모델을 사용하여 학습했다. 각 평가지표에 대한 설명은 다음과 같다.

정밀도(Precision)란 모델이 True라고 분류한 결과 중에 실제 True인 결과의 비율이다. 흔히 PPV(Positive Predictive Value)라고도 불린다.

$$(Precision) = \frac{TP}{TP + FP}$$

재현율(Recall)이란 실제 True인 결과 중에서 모델이 True라고 예측한 것의 비율이다.

$$(Recall) = \frac{TP}{TP + FN}$$

정확도(Accuracy)란 False를 False로 예측한 경우도 고려하는 지표다.

$$(Accuracy) = \frac{TP + TN}{TP + FN + FP + TN}$$



F1 score는 precision과 recall의 조화평균으로써, 데이터 label이 불균형 구조일 때, 모델의 성능을 정확하게 평가할 수 있으며, 성능을 하나의 숫자로 표현할 수 있다.

$$(F1-score) = 2 \times \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

결과를 살펴보면 전체적으로 F1 score가 상당히 낮다고 할 수 있는데, 이는 추가적인 연구를 통해 데이터셋을 보완하고, 모델과 특징 벡터의 수정이 필요함을 의미한다. 결과적으로 공종별 분류에서는 GoogleNet의 Inception V3 model이 가장 높은 F1 점수를 나타내고 있다.

	Accuracy	Precision	Recall	F1 score
AlexNet	15.10%	15.55%	13.54%	14.47%
VGG16	47.66%	78.82%	23.44%	36.09%
VGG19	44.92%	75.61%	24.22%	36.69%
MobileNets	37.89%	38.36%	33.59%	35.82%
Inception V3	48.44%	51.94%	46.48%	49.06%

## (2) 하자 점검 및 공종별 분류

AlexNet, VGG16, VGG19, MobileNets, Inception-V3의 모델을 사용해 총 2,290장(Training set: 1,729장, Test set: 561장)의 건설 현장 사진을 학습해 분류를 한 결과는 다음과 같다.

하자 점검 및 공종별 분류에서는 MobileNets이 37.82% 가장 높은 F1 점수를 나타냈고, Inception V3, VGG16가 각각 31.38%, 28.15%로 뒤를 이었다. 전체적으로 F1 점수가 낮은 것은 추가적인 데이터셋과 모델 보완의 필요성을 제기한다.

	Accuracy	Precision	Recall	F1 score
AlexNet	9.38%	12.54%	8.33%	10.01%
VGG16	35.94%	60.33%	18.36%	28.15%
VGG19	32.03%	66.06%	7.81%	13.95%
MobileNets	37.11%	41.48%	34.77%	37.82%
Inception V3	31.25%	33.77%	29.30%	31.38%

## 6. 결론

### 6.1 한계점

#### (1) 도메인 전이 학습 미적용

데이터셋 구성 시 조적공사, 미장공사, 수장공사, 도장공사 분야의 하자 이미지가 부족하여 "KolektorSDD" dataset을 토대로 모델 학습을 실시하였다. KolektorSDD Image는 단순한 균열 이미지인 반면에, 실제 건설 사진은 기둥, 벽, 바닥, 천정 등의 균열 이미지이다. 따라서 학습 사진과 타겟 사진과의 차이가 존재한다. 이 차이를 최소화하여 오차를 줄이기 위해 도메인 전이 학습을 실시하는 것이 좋다. 하지만 본 연구에서는 도메인 전이 학습을 적용하지 않았고, 이는 오류를 발생시키는 요인일 것이라고 추정된다.

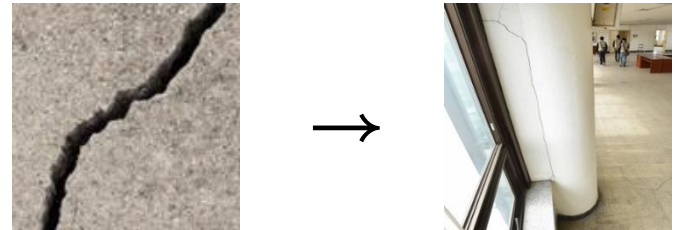


Figure 6 학습 사진과 타겟 사진

#### (2) 데이터셋의 제한

데이터셋은 1차적으로 서울특별시 건설사업관리단에서 운영하고 있는 서울특별시 건설알리미에서 건설 현장 사진을 획득했다. 해당 이미지들은 실제 건설 현장 사진과 유사한 이미지를 제공하지만, 학습을 위한 절대적인 이미지 갯수가 부족하다. 따라서 추가적인 사진은 인터넷 검색을 통해 획득했다. 인터넷을 통해 획득한 사진들은 실제 건설 현장 사진과 유사한 것처럼 보이지만, 차이점들이 존재한다. 이러한 차이점들이 학습을 하는 데 있어 부정적 요인으로 작용한 것으로 추정된다.

#### (3) 연속적인 건축공사 공정

일반적으로 건축공사는 같은 시기에 여러 가지 공정이

동시에 이루어지는 경우가 있다. 예를 들어, 아래 사진은 철근 콘크리트 공사로 구분되지만, 가설공사로도 구분될 수 있다. 동시에 공사가 이루어져 정확히 분류하기 애매한 특성이 모델의 학습 능력을 저해시킨 것으로 판단된다.



철근 콘크리트 공사



가설 공사

## 6.2 향후 개선 방안

### (1) 도메인 전이 적용

도메인 전이 학습을 적용하여 단순 균열 이미지를 통해 조적공사, 미장공사, 수장공사, 도장공사의 균열 이미지 부분에 대한 학습이 잘 이루어지도록 적용할 것이다.

### (2) 다양한 과잉적합(overfitting) 규제 방법 적용

실험 결과를 살펴보면, 몇 개의 모델에서 overfitting 현상이 일어나고 있다. 이를 해결하기 위해 데이터셋의 양 증가, 여러 가지 규제 기법 적용, 모델의 복잡도 감소 등을 통해 과잉적합이 발생하지 않도록 훈련해야 한다.

## 6.3 기여점 및 최종 결론

본 연구는 세 가지의 기여점을 가지고 있다. 첫째, 기존의 딥러닝을 활용한 건설 공종별 사진 분류를 활용해 건설 현장에서 발생하는 하자의 자동적인 분류를 시도

했다는 점에서 기여점을 가지고 있다. 해당 시도는 건설 분야에 딥러닝 기술을 통해 공사의 효율을 증가시켜 성공적인 건설 과정이 이루어지도록 기여한다.

둘째, 본 연구는 기존 건설 현장에 내재되어 있는 업무의 중복성 제거 및 자동화를 통해 업무의 효율성을 극대화할 수 있다. 이를 통해 건설 안전 관리 분야에 비교적 많은 시간을 투자하게 되어 건축물의 품질이 향상될 수 있다.

셋째, 본 연구는 건설 하자 점검 및 공종별 분류에 따른 데이터셋을 구축하는 기준을 제시했다는 점에서 기여점을 가지고 있다. 데이터셋을 구축하기 위한 23개의 명확한 하자, 공종별 분류체계 기준을 제시하였다.

## 참고문헌

- [1] 이미연, 이학주, 김형근(SH공사 도시연구소), "공동주택 하자유형 분석 및 사전예방을 위한 연구"
- [2] 조달청 시설사업국 공사관리팀, "하자발생사례 분석"
- [3] Domen Tabernik, Samo Sela, Jure Skvarc, Danijel skoca, "Segmentation-Based Deep-Learning Approach for Surface-Defect Detection", 11 June 2019, arXiv:1903.08536v3 [cs, CV]
- [4] Gil Dae-Young, Jeon Ka-Hyun, Lee Ghang, "Classification of Images from Construction Sites Using a Deep-learning Algorithm", October 2017, ARCHITECTURAL INSTITUTE OF KOREA
- [5] 길대영, "딥 러닝 기반의 공종 별 건설현장사진 자동 분류 방법"
- [6] 국토교통부(기술정책과), "건설사업정보 운용지침", 국토교통부고시 제2015-850호, 2015. 11. 20. 일부개정
- [7] 서울시 건설사업관리단, 서울특별시 건설알리미, [https://cis.seoul.go.kr/TotalAlimi\\_new/Main.action?cmd=main](https://cis.seoul.go.kr/TotalAlimi_new/Main.action?cmd=main)

[8] Alex Krizhevsky, Ilya Sutskever, Geoffrey E.Hinton, "ImageNet Classification with Deep Convolutional Neural Networks"

[9] Karen Simonyan, Andrew Zisserman, Visual Geometry Group, Department of Engineering Science, University of Oxford, "Very Deep Convolutional Networks For Large-Scale Image Recognition", ICLR 2015, arXiv:1409.1556v6 [cs.CV] 10 Apr 2015.

[10] Andrew G.Howard, Google Inc, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", arXiv:1704.04861v1 [cs.CV] 17 Apr 2017.

[11] Christian Szegedy<sup>1</sup>, Wei Liu<sup>2</sup>, Yangqing Jia<sup>1</sup>, Pierre Sermanet<sup>1</sup>, Scott Reed<sup>3</sup>, Dragomir Anguelov<sup>1</sup>, Dumitru Erhan<sup>1</sup>, Vincent Vanhoucke<sup>1</sup>, Andrew Rabinovich<sup>4</sup>  
<sup>1</sup>Google Inc. <sup>2</sup>University of North Carolina, Chapel Hill  
<sup>3</sup>University of Michigan, Ann Arbor <sup>4</sup>Magic Leap Inc,  
"Going Deeper with Convolutions"