

프로젝트 #1 결과 발표

[CNN을 이용한 불량 검출]

2022. 06. 15.

충북대학교 산업인공지능학과

[20-3조] 최원희, 손의걸

수행방법 및 기여도

수행방법

- 과제 수행을 위해 자유롭게 프로젝트 구현 및 실습
- 인터넷 논문 샘플 모델링 공유 및 Github 예시 자료 등 참조
- 프로그래밍 실습 이후 결과 값 비교 및 자료 합치는 방식으로 진행

업무분장 및 기여도

이름	비중	수행내용	비고
최 원 희	60%	<ul style="list-style-type: none">• 자료 검색 및 취합 정리• 데이터 증량 / 코딩 / 학습• 주제발표 및 결과발표	프로그래밍 관련 현업 해당 없음
손 의 걸	40%	<ul style="list-style-type: none">• 자료 검색• 코딩 / 학습• 중간 발표	코딩 능력 미숙

데이터 셋

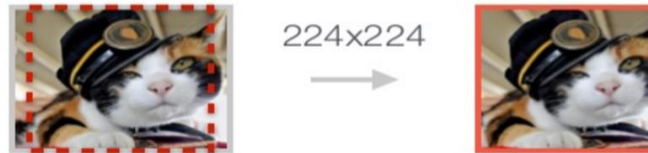
Data augmentation / 전처리

- 방법 : 1) 원본 이미지에서 적절한 변경을 가해서 새로운 데이터를 만들어 내는 방식
2) 원본 이미지를 상하좌우 방향으로 이동하거나, 여러가지 변환을 조합해 이미지 데이터 개수를 증가시킴

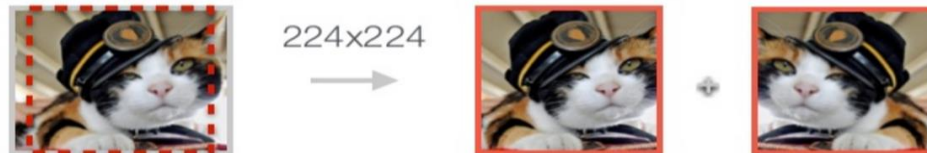
- 결과 : 이미지 편집 등

- 실행 : 이미지 데이터

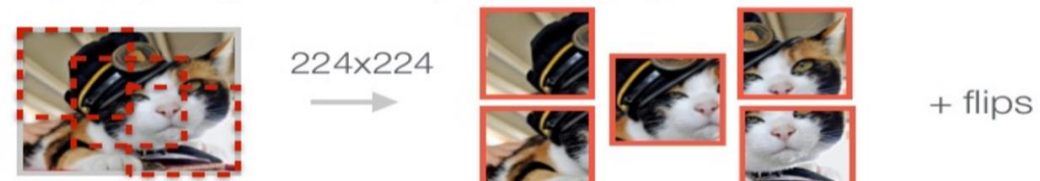
a. No augmentation (= 1 image)



b. Flip augmentation (= 2 images)

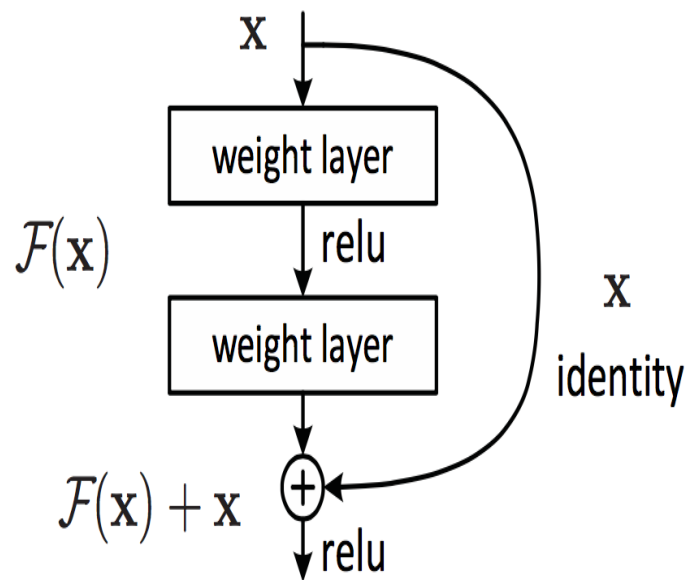
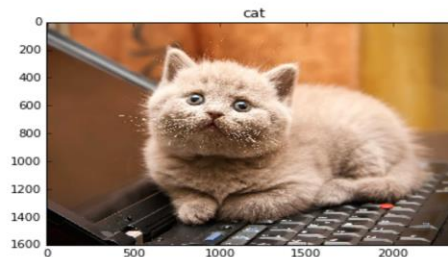
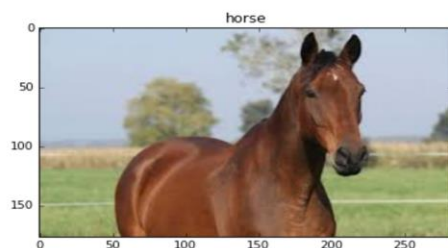
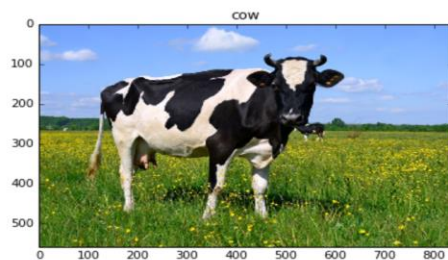
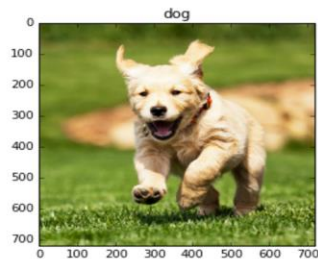
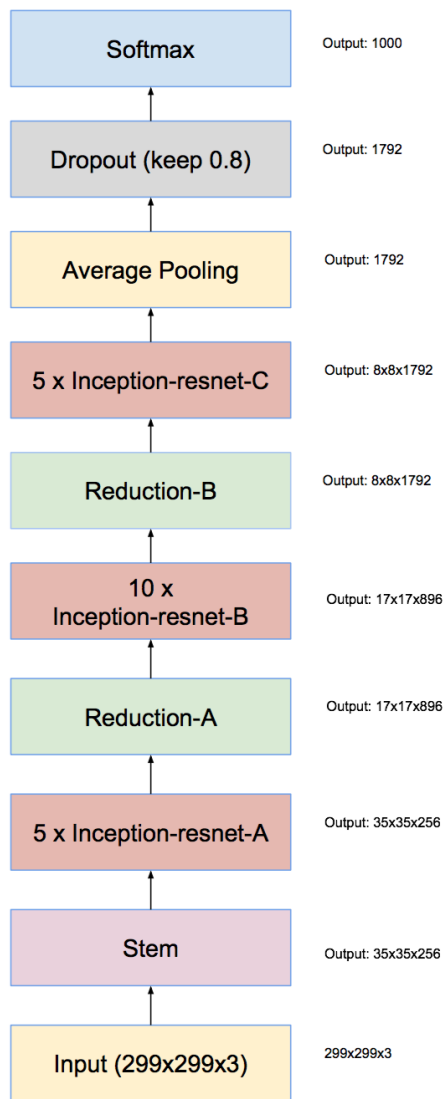


c. Crop+Flip augmentation (= 10 images)



- 장점 : 1) 수집된 데이터가 적을 경우 다양한 모형으로 변형이 가능함
2) 모델을 더욱 견고하게 만들 수 있어 실제 데이터를 가지고 더 높은 성능 기대

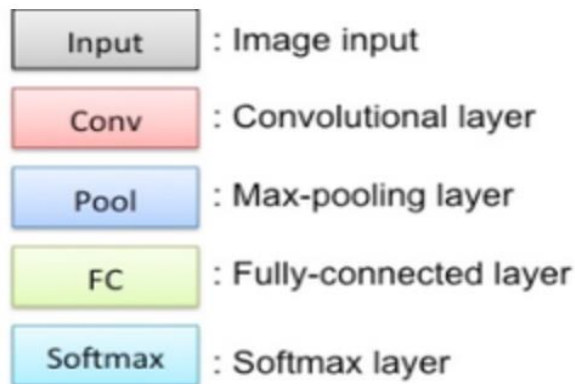
데이터 흐름도



데이터 구성

데이터 구성

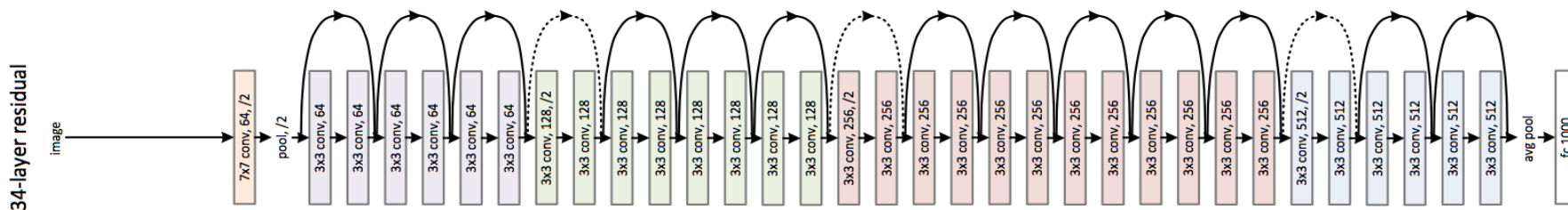
VGG



VGGNet



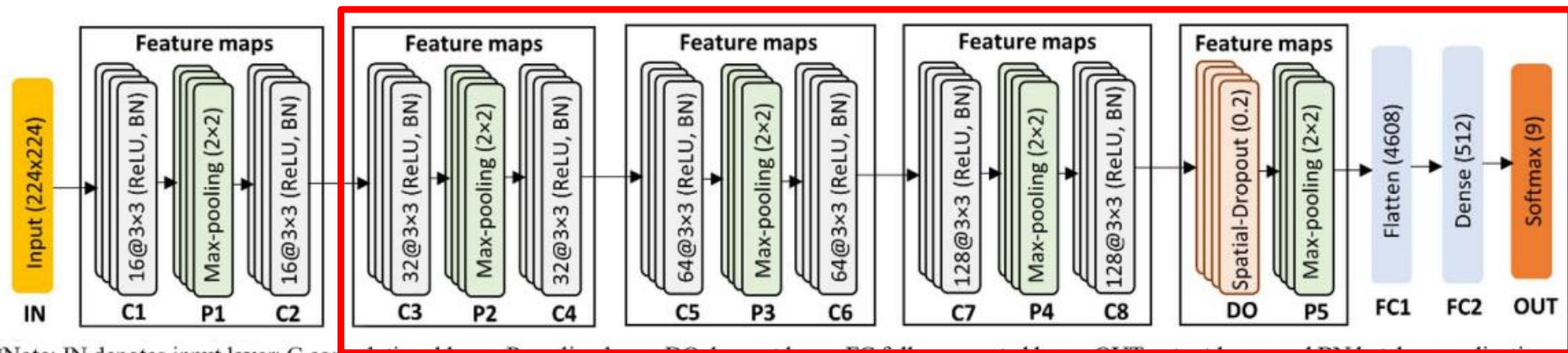
ResNet



CNN 구조

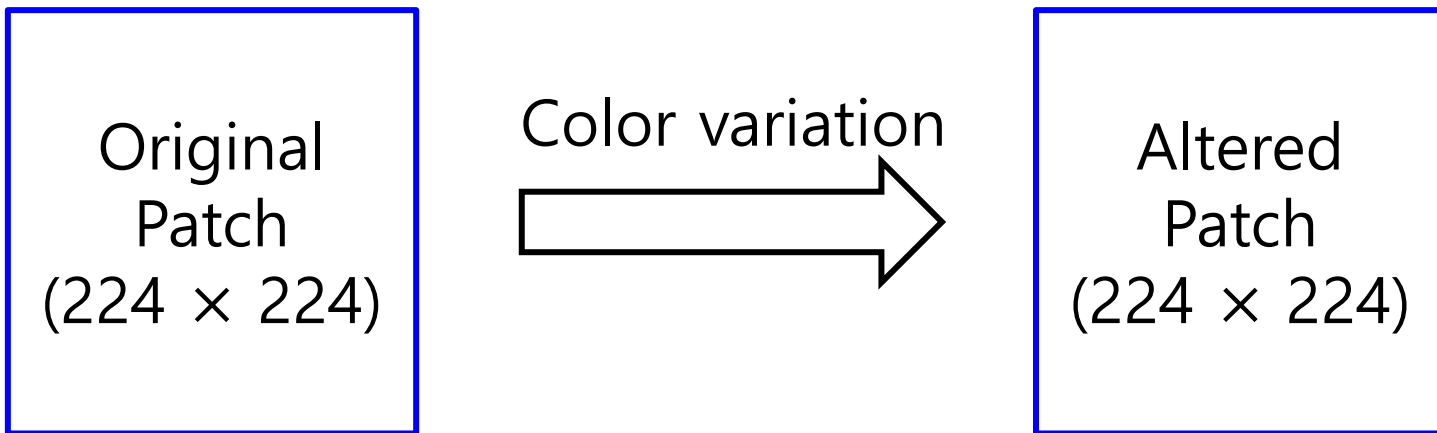
CNN 구조

- 동일 조건의 예제와 시험 대상이 아니므로 결과 값에 대한 구조 내용 상이



*Note: IN denotes input layer; C convolutional layer; P pooling layer; D0 dropout layer; FC fully connected layer; OUT output layer, and BN batch normalization

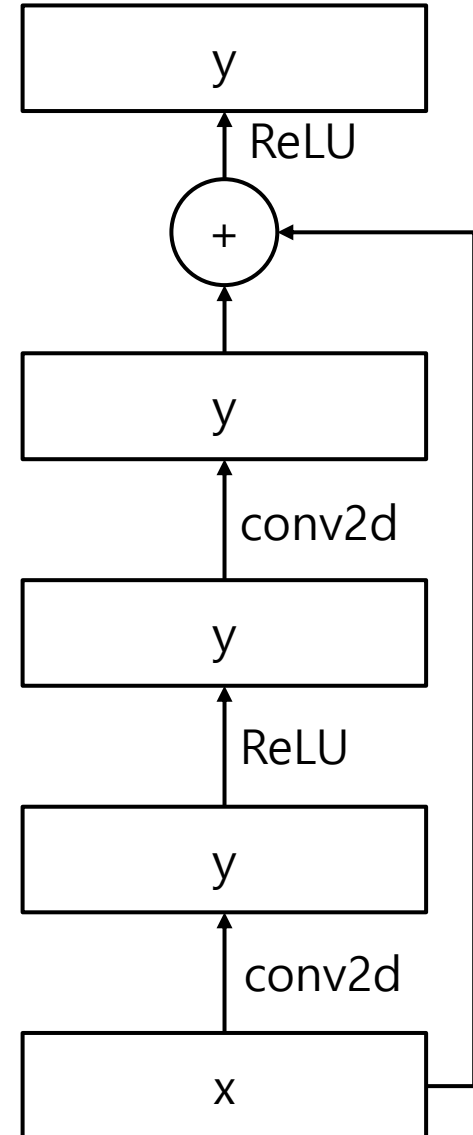
과적합을 방지하기 위한 규제화(regulation)



CNN 실습

주요 코드 및 실행 결과

```
1 # FIRST, SEE WHAT'S INSIDE THE FOLDER
2 cwd = os.getcwd()
3 path = cwd + "/img/"
4 flist = os.listdir(path)
5 print ("[%d] FILE ARE IN [%s]" % (len(flist), path))
6
7 # LIST UP FILES INSIDE THE FOLDER
8 for i, f in enumerate(flist):
9     print ("[%d] TH FILE IS [%s]" % (i, f))
10
11 # LOAD IMAGES WITH APPROPRIATE EXTENSIONS
12 valid_exts = [".jpg", ".gif", ".png", ".tga", ".jpeg"]
13 imgs = []
14 names = []
15 for f in flist:
16     ext = os.path.splitext(f)[1]
17     if ext.lower() not in valid_exts:
18         continue
19     # LOAD AND APPEND TO IMGS LIST
20     fullpath = os.path.join(path, f)
21     imgs.append(imread(fullpath))
22     # APPEND NAMES
23     names.append(os.path.splitext(f)[0])
24
25 # PLOT LOADED IMAGE
26 for img, name in zip(imgs, names):
27     plt.imshow(img)
28     plt.title(name)
29     plt.show()
```



학습 방법

딥러닝 학습 조건

- (HW) PC 사양

CPU : [AMD] 라이젠5 버미어 5600X (6코어/스레드/3.7GHz)

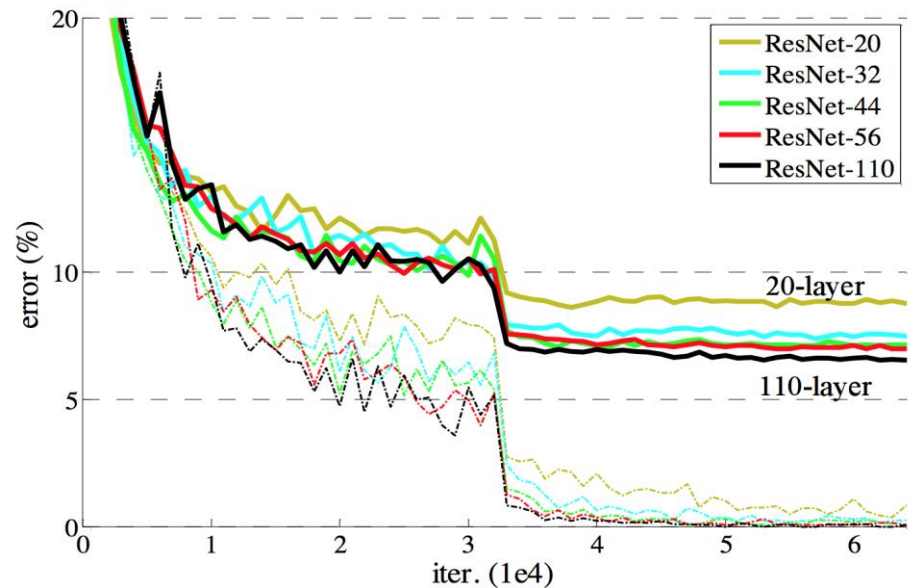
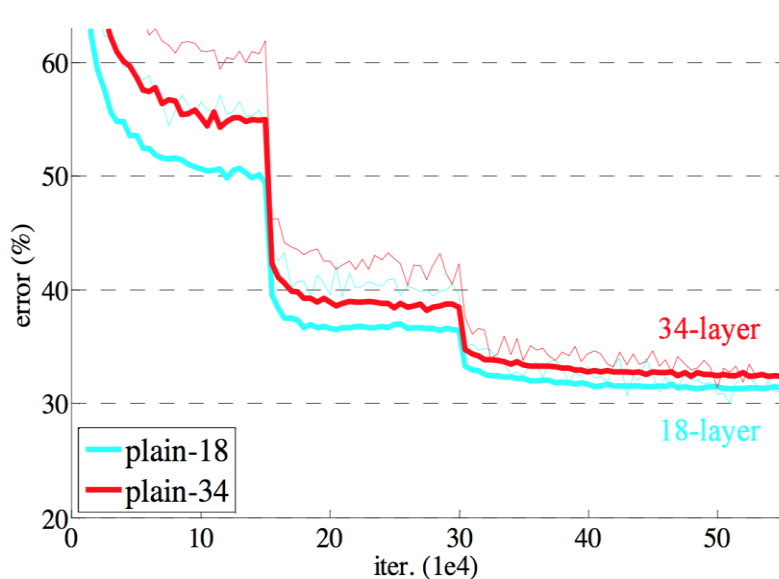
RAM : [Team Group] DDR4 32GB PC4-25600 Elite

GPU : [NVIDIA] Quadro RTX A6000 D6 48GB

- 파라미터 및 연산량 : 6.000개 / 1만개

- 학습시간 : 약 4시간

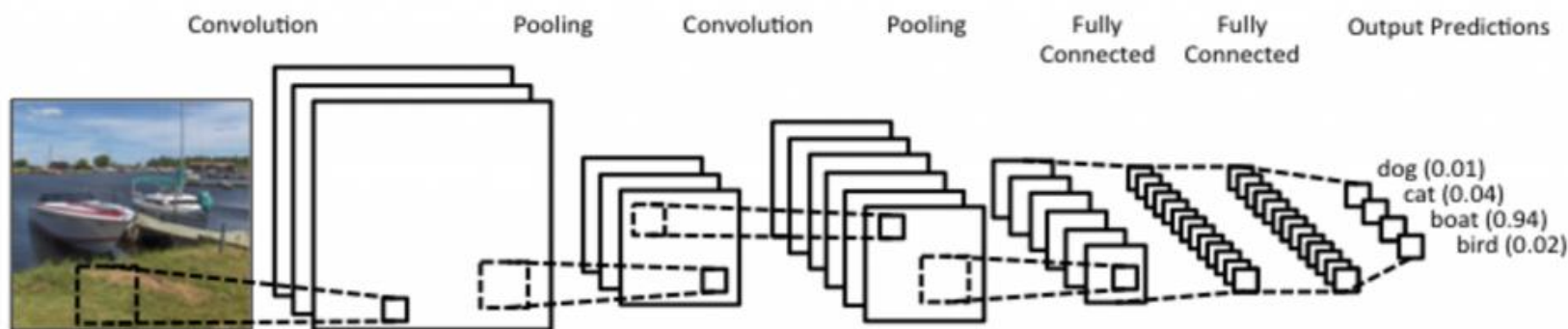
- 최적화 : ReLU 활성화 함수 (하이퍼 파라미터)



결과 내용

분류 성능

- 기존 논문 자료에 대한 비교 데이터 자료 산출 불가 (프로젝트 산출 진행 데이터가 상이함)
- 미흡 하나 다른 예제 진행 및 직접 실습(코딩)을 통해 CNN 관련 수학 및 이해에 만족



OVERALL PERFORMANCE COMPARISON OF VARIOUS CLASSIFIERS (%)

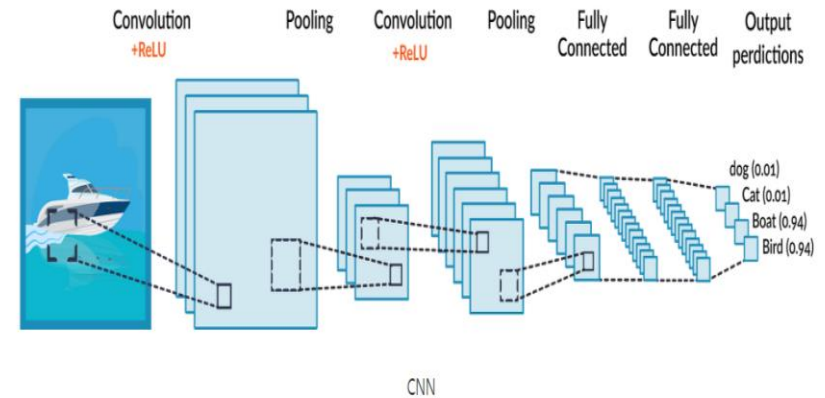
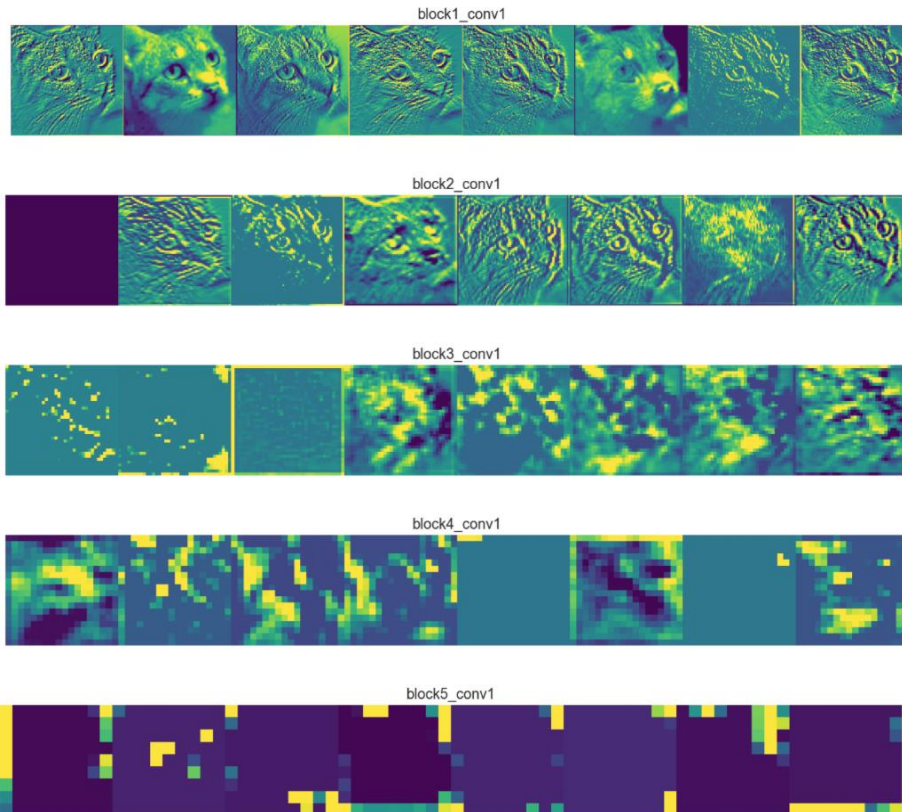
Classifier	Training Acc	Validation Acc	Testing Acc	Precision	Recall	F1-Score
CNN-WDI	98.9	96.4	96.2	96.2	96.2	96.2
CNN-D	97.6	95.5	95.2	95.2	95.2	95.2
CNN-BN	99.4	95.6	95.6	95.6	95.6	95.6
CNN-SD	98.6	94.7	94.8	94.8	94.8	94.8
VGG-16	82.3	80.0	80.1	80.3	80.1	79.9
ANN	95.9	95.9	72.0	95.2	95.9	95.4
SVM	91.3	91.0	32.6	87.5	91.0	88.0

Note: Boldface numbers denote the highest values of different performance measures and Acc accuracy

결과 고찰

토의 및 개선점

- CNN은 데이터의 특징을 추출하여 특징들의 패턴을 파악하는 구조
- 주로 정보 추출, 문장 분류, 얼굴 인식 등의 분야에 널리 사용되고 있음
- 다양한 데이터 자료 확인 및 실습을 통해 CNN에 대한 정의와 특징에 대해 알게 되었음



감사합니다