

# 합성곱 신경망 기반의 꽃 분류기

김해린, 노윤아, 박미희, 엄단경, 이세미  
세종대학교 지능기전공학부

[linil634@naver.com](mailto:linil634@naver.com) / [eksrud9403@naver.com](mailto:eksrud9403@naver.com) / [mihee78952@naver.com](mailto:mihee78952@naver.com) / [yoona028@naver.com](mailto:yoona028@naver.com) / [98sammy@naver.com](mailto:98sammy@naver.com)

## 요 약

딥러닝 기법 중 합성곱 신경망을 활용한 꽃 분류기를 설계하고 웹을 이용하여 사용자 인터페이스를 구축한다. 기존의 꽃 분류기 어플리케이션과는 차별점으로 각 꽃마다 축제 정보를 연결하여 기술을 대중적으로 이용할 수 있도록 한다. CNN의 네트워크 중, VGG16, Resnet34, \*참고자료의 모델을 사용하여 정확도를 비교한다. 각각의 모델에 Epoch, learning rate, batch를 조절하고 정확도가 가장 높게 나오는 모델을 선정하여 적용한다. 모델 체크포인트 제작 후, Visual Studio Code를 이용하여 CNN 모델과 html 코드를 연동하여 꽃 정보를 출력하는 웹을 제작한다.

## 1. 서론

현재 시중에 나와 있는 꽃 분류기는 앱 형태로 스마트폰 카메라로 인식된 꽃 사진을 분석하여 이름과 각종 정보들을 제공한다. 합성곱 신경망 모델로는 최신 네트워크 구조를 바탕으로 다양한 딥러닝 기술을 활용해서 새로운 꽃 검색 알고리즘을 지속적으로 개발하고 있다고 한다. 본 aicoco팀은 합성곱 신경망의 3가지 모델을 사용하여 정확도를 분석하고 성능을 높이기 위해 데이터 수와 클래스 수에 변화를 주었다. 사용자 인터페이스로는 웹페이지를 이용하고, 꽃을 인식한 후에는 꽃과 관련된 축제 정보를 제공하여 기술의 효용성을 높이려고 계획하였다. 본 aicoco팀에서 사용한 vgg모델은 기존 vgg16에서 마지막 512개의 채널 수를 이용하는 convolution layer 3개를 제외한 13-layer로 이루어진 것을 이용하였다. flower classification참고 모델은 총 layer가 7개이고 filter 개수가 4개로 처음 32, 64, 128, 256, 128의 5개 convolution layer를 거치고 Fully connected layer로 2048, 1024의 2개 layer를 이용했다. resnet34모델은 기존 모델과 구성은 같으나 마지막 fully connected layer에서 1000이 아닌 512개를 사용하였다.

## 2. 합성곱 신경망 모델을 이용한 꽃 분류

### 2.1 모델 성능 향상 과정

학습 클래스는 개나리, 메밀꽃, 코스모스, 해바라기, 백합, 무궁화, 벚꽃, 튜립, 장미, 진달래로 총 10개이다. 학습은 클래스 당 1000장의 데이터로 총 10000장을 사용하였다. 그리고 이 10000장의 데이터를 90도 회전, 180도 회전, 270도 회전, 색 반

전, 흑백변환으로 가공하여 총 60000장의 데이터를 얻었다. 우리는 flower classification<sup>\*1</sup>, VGG16, Resnet34<sup>\*2</sup> 세 가지모델을 사용하여 학습을 진행하였고 learning rate는 0.0001로 고정하였다. 위의 초기조건으로 실험을 진행하였을 때, VGG16 모델의 batch size를 50, epoch를 100으로 설정하니 72%의 정확도가 나왔다. overfitting의 가능성을 염두에 두고 dropout을 0.6으로 주었으나 정확도는 72.9%로 주기 전과 큰 차이가 없었다. VGG16모델과 같은 조건에서 Resnet34 모델로 실험을 진행하였을 때에는 54%의 정확도가 나왔다. 마찬가지로 같은 조건에서 flower classification 모델로 실험을 진행하였을 때에는 50%의 정확도가 나왔다. 이 때, flower classification 모델에서 drop out을 사용하여 보니 45%정도의 정확도가 나왔다.

표 1. 데이터수가 60000 장인 경우 모델 별 가장 높은 정확도

	Accuracy
Flower classification	50%
Vgg16	72%
Resnet34	54%

이를 통해서 데이터가 random하게 학습되는 것이 정확도를 높이는 데에 무조건 좋은 영향을 미치는 못함을 확인하고 drop out을 주지 않고 이후의 실험을 진행하였다.

그후 클래스별 정확도를 뽑아보니 장미의 정확도가 0.041%로 매우 낮은 것을 발견하였다. 그로 인해 장미의 데이터를 확인해보니 장미의 색깔이 7

가지인 것을 확인하고 데이터의 다양한 색깔이 학습에 영향을 준 것이라고 추측하였다. 그래서 빨간색 장미로만 데이터를 다시 모아 가공하였다. 그 후 VGG 모델의 batch size를 50, epoch을 50으로 설정하여 학습을 진행하니 72%의 정확도가 나왔다. 같은 조건에서 데이터를 바꾸기 전에는 71%의 정확도가 나왔었으므로 미세하게 정확도가 오른 것을 확인할 수 있었다. 그리고 장미의 클래스별 정확도만 뽑아보니 17%로 올랐지만 아직도 정확도가 매우 낮음을 확인하였다. 이 과정에서 우리가 학습을 grey channel로 진행하기 때문에 장미의 색깔이 주원인이 아닐 수 있겠다는 추측을 하게 되었다. 이 추측을 이유로 하여 전체 데이터를 확인하여 보니 풍경 위주의 꽃밭 데이터가 생각보다 많음을 발견하였다. 그리하여 데이터의 일관성에 문제가 있을 수 있다고 판단을 하여, 데이터를 가까이에서 명확하게 찍힌 사진으로만 클래스당 250장씩 간추린 후 다시 가공하였다. 즉, 기존에는 클래스 당 10000장씩 총 60000장의 꽃 데이터를 사용하였지만 이 과정을 거치며 클래스당 250장씩 12500장의 꽃 데이터를 사용하게 되었고 결과적으로 데이터의 양이 대폭 축소되었다.

이러한 변화 후에 VGG16 모델에서 batch size를 50, epoch을 100으로 학습을 진행하였을 때, 74%의 정확도가 나왔다. 같은 조건에서 Resnet34 모델로 실험을 진행하였을 때에는 61%의 정확도가 나왔다. flower classification 모델에서 batch size를 50, epoch을 200으로 설정하여 학습을 진행하였을 때에는 60%의 정확도가 나왔다.

표 2. 데이터수가 10000 장인 경우 모델 별 가장 높은 정확도

	Accuracy
Flower classification	60%
Vgg16	74%
Resnet34	61%

정확도가 오르기는 했지만 클래스별 정확도를 뽑아봤을 때 아직도 정확도가 50%를 넘지 못하는 클래스들이 몇 있다는 것을 확인하였다.

```
[[124 14 14 72 11 8 3 0 4 0]
 [ 9 123 17 13 9 6 13 12 7 41]
 [ 4 10 184 13 0 16 0 6 6 11]
 [ 5 0 2 235 7 0 0 0 0 1]
 [ 5 0 0 16 218 0 11 0 0 0]
 [ 0 8 16 0 0 213 0 7 0 6]
 [ 4 3 4 14 29 0 190 0 1 5]
 [ 2 11 37 0 0 43 1 147 4 5]
 [ 0 2 16 4 0 6 0 0 220 2]
 [ 7 6 12 8 4 3 4 6 3 197]]
```

그림 1. 각 2500 장의 10 개 클래스의 confusion matrix

그림 1의 confusion matrix에서 정확도가 낮은 순으로 벚꽃, 메밀이 나왔다.

```
[[161 5 4 39 34 0 7 0 0 0]
 [ 11 117 10 21 15 9 22 7 3 35]
 [ 7 18 153 10 4 18 2 8 18 12]
 [ 8 0 1 210 23 0 7 0 0 1]
 [ 1 0 0 4 207 0 38 0 0 0]
 [ 7 7 23 2 0 180 0 18 0 13]
 [ 0 1 0 4 30 0 212 1 0 2]
 [ 3 10 38 0 1 26 1 159 3 9]
 [ 0 1 7 4 1 10 0 1 224 2]
 [ 9 5 8 0 2 5 15 3 0 203]]
```

그림 2. 각 2500 장의 10 개 클래스의 confusion matrix

그림 2의 confusion matrix에서 정확도가 낮은 순으로 벚꽃, 코스모스가 나왔다.

클래스별 정확도가 낮은 벚꽃, 메밀, 코스모스의 클래스를 제외하면 정확도가 오를 것이라고 추측이 되어, 이러한 변화를 적용한 후 다시 실험을 진행하였다. 그 결과, vgg16 모델의 batch size가 25, epoch이 200인 경우 86%의 정확도가 나왔다.

그리고 같은 데이터로 flower classification 모델의 batch size를 50, epoch을 200으로 설정하여 학습을 진행하니 85%의 정확도가 나왔다.

표 3. 데이터수가 7000 장인 경우 모델 별 가장 높은 정확도

	Accuracy
Flower classification	85%
Vgg16	86%

위에서 제외한 3개의 클래스들은 모두 꽃밭 풍경의 데이터들이 많은 경우였다. 우리는 이를 통해 꽃들이 다발로 뭉쳐 있는 경우에 정확도가 낮게 나온다는 것을 발견하였다.

## 2.2 축제 정보 연결과 웹 제작

Visual studio code에서 html과 python을 이용하여 웹을 제작하였다. 이미 완성된 분류기 웹 코드\*3를 참고하여 이미지 분류하는 함수 부분을 변경하여 실행하였다. 본 코드 원본은 inception\_v3를 이용하여 이미지 분류를 하였으나 정확도가 낮아서 본 프로젝트에서 앞서 작성한 CNN코드로 바꾸어 사용하였다. visual studio code를 사용하여 server.py 파일을 js, css, html, json 파일과 연동하여 설계하였다.

server.py내에서 어떤 꽃인지 예측하는 predict함수를 작성하기 위해서 가장 높은 정확도를 가진 Flower classification 체크포인트 모델을 이용하였다. Flower classification 모델에서 꽃 이름과 확률을 받아온 후 server.py

부분에서 새로운 배열로 만들어 엮어 페이지에서 출력되도록 설계하였다. system.html 파일에서는 웹 페이지에서 'Class Name' 과 'Score' 부분의 결과가 간략하게 출력 될 수 있도록 수정하였다. 또한 웹의 첫번째 페이지와 그림3의 이미지 분류 페이지에서의 배경, 문구를 프로젝트의 효용성에 맞게 수정하였다. 페이지 하단에 'Powered By - AICOCO' 를 클릭하면 프로젝트의 초반 기획안부터 진행 과정, 최종 결과물까지 정리되어 있는 팀의 오픈소스 Github 페이지에 연결되도록 설정하였다.

결과적으로 웹에서 이미지 주소를 입력하면 이미지가 뜨고 분류된 꽃 이름과 확률이 뜨도록 웹을 설계하였다.

본 계획은 웹에 축제 정보를 연동하여 꽃 이름과 함께 출력하는 것이었다. system.html에서 <head> 부분에 새로운 cell innerHTML변수를 추가해서 festival정보 연결을 시도하였으나 새로운 cell을 생성하지 못하였다. 웹 축제 정보 연동에 대한 대안으로 코랩을 이용해서 엑셀 파일과 Flower classification 분류 코드를 연동하여 분류된 꽃 종류와 연동된 축제 정보를 출력하였다.

## AICOCO



그림 3. Visual Studio Code로 구현한 웹페이지 출력 결과물

## 3. 실험 결과 및 분석

꽃 사진을 분류할 때 멀리서 찍힌 데이터의 경우 정확도가 낮은 것으로 보아 멀리서 꽃밭풍경으로 흐드러지게 찍힌 데이터들보다 가까이서 특징이 명확하게 드러난 데이터들이 학습에 유리한 것을 확인하였다. 비슷한 맥락으로 전반적으로 데이터가 정제되지 않고 양이 많은 경우보다 양질의 정제된 데이터를 사용하는 것이 성능을 높이는 데에 효과가 있었다. 또한 drop out 역시 데이터가 깔끔하지 않은 경우에는 오히려 악영향을 줄 수 있고, 원래 drop out을 사용하지 않는 resnet모델의 경우에는 drop out을 사용할 경우 정확도가 훨씬 떨어지는 것을 알 수 있었다. 그리고 현재 resnet34가 vgg16보다 예러율이 낮은 것으로 발표가 났지만 꽃 이미지 인식의 경우 vgg16이 resnet34보다 좋은 성능을 가지는 것을 발견하였다.

## 4. 결론

처음 60000장의 데이터에서 3가지의 각 모델로 72%의 최고 성능을 뽑은 후, 더 이상 현재의 모델을 이용하여 성능을 높이는 데에 한계가 있겠다고 판단하여 데이터를 전처리 하는데 주력하였다. 각 클래스당 데이터에서 가까이 찍힌 명확한 사진들로만 250장씩 간추려 총 2500장의 데이터로 정제하였다. 또한 test데이터에서 각 클래스의 데이터가 고르게 분포되도록 하기 위하여 전체데이터를 shuffle 시킨 후 train과 test로 나누어 사용하는 방식 대신 클래스 당 train 200개, test 50개로 나눠 데이터를 구성한 후에 shuffle하는 방식을 사용하였다. 하지만 예상 외로 정확도가 2% 밖에 오르지 않았다. 그리하여 다시 클래스별 정확도를 확인하여 가장 정확도가 낮은 세 클래스를 제외했다. 그 후 다시 학습을 진행하니 flower classification은 85%, vgg16은 86%로 resnet34를 제외하고 두 모델이 모두 높은 정확도가 나오는 것을 확인할 수 있었다.

이후 웹을 통해 이미지를 테스트해 보면서 정확도가 낮은 것을 확인하였다. 이미지 사이즈가 28\*28로 굉장히 작았고 기존 이미지에 비해 가로 세로 비율이 달라져 실제 이미지로 테스트할 때 정확도가 더 떨어진 것 같다고 추측해 본다.

## 감사의 글

2 년 6 개월가량의 학부과정을 거치고 해당 논문을 제출하게 되었습니다. 4 월부터 약 3 개월간 최유경 교수님의 지도하에 수많은 피드백을 나누며 상당한 도움을 얻었습니다. 이에 진심으로 감사드리며 끝까지 희망을 놓지 않고 열심히 해준 조원들에게도 감사의 인사를 전합니다.

## 참고문헌

- [1] [https://github.com/Vibashan/Flower\\_Classification](https://github.com/Vibashan/Flower_Classification)
- [2] Kaiming He Xiangyu Zhang Shaoqing Ren Jian "Deep Residual Learning for Image Recognition" Sun Microsoft Research
- [3] <https://github.com/akshaybhatia10/DeepClassifyML>