



**Journal of Knowledge Information Technology and Systems**

ISSN 1975-7700

<http://www.kkits.or.kr>

---

## **Indoor Plants Image Classification Using Deep Learning and Web Application for Providing Information of Plants**

**Hong-Jae Shin, So-In Lee, Hui-Won Jeoung, Jangwoo Park\***

*Department of Information & Communication Engineering, Sunchon National University*

---

### **A B S T R A C T**

Plants have good effects such as air purification and landscaping, but they have special ingredients to protect themselves. Ingredients made to protect the plant itself can harm people or animals. There are also accidents that are mistaken for other plants with similar plant features. We have implemented a program that can identify plants and display information about each plant. Using deep learning to classify images, we created a web application that predicts plant names and displays information that matches the predicted plants. We created a data set of 61 indoor plants using Google Image Search. The positive effects and negative toxicity of indoor plants are summarized in the database. Deep learning is implemented using fast.ai, a Pytorch-based framework. Through data Augmentation, we increased the number of images to learn. Indoor plant image data were trained using ResNet50, a pretrained model using various images. The accuracy of the model was about 97.5%, which predicted most plants accurately. The web application was implemented using flask, a Python-based web framework. Using the implemented image classification deep learning model, the plant name is predicted and the information corresponding to the predicted plant name is displayed on the web page. The web application can be optimized for mobile devices and used conveniently.

© 2020 KKITS All rights reserved

---

**KEYWORDS :** Indoor plant information, Deep learning, Transfer learning, ResNet, Fast.ai

---

**ARTICLE INFO:** Received 27 November 2019, Revised 10 January 2020, Accepted 10 April 2020.

---

---

\*Corresponding author is with the Department of Information & Communication Engineering, Sunchon National University, 255, Jungang-ro, Suncheon-si,

Jeollanam-do, Republic of Korea, 57922  
*E-mail address:* [jwpark@sunchon.ac.kr](mailto:jwpark@sunchon.ac.kr)

### 1. 서론

우리 주변을 둘러보면 다양한 식물들이 있다. 야외의 식물들은 조경이나 공기 정화, 도시의 열섬 방지 등의 미적, 기능적 장점들을 가지고 있다. 가정에서 식물을 기르면 집안의 분위기를 살려 정서적 안정에 도움을 주고 최근 이슈가 되었던 미세먼지 정화 등 공기 정화에도 긍정적인 효과를 가져다준다.

하지만 식물의 특성상 다른 종의 식물 각각은 뚜렷하게 구분이 가능하지만 같은 종에 속해있는 식물들은 생김새가 비슷하여 구분하기 힘들다. 또한 같은 종에 속해있더라도 식물 각각의 특성이 다를 수 있다. 식물은 자신을 보호하기 위해 특별한 성분을 가지기도 하는데, 사람이나 동물이 해당 식물을 섭취하거나 접촉할 경우 예기치 못한 부정적인 효과가 일어날 수 있다. 자신이 평소 알던 식물이라 판단하고 섭취하여 사고로 이어지는 사례가 있고[1], 비전문가들이 화단을 조성하면서 비슷한 이름이지만 전혀 다른 성격을 가진 식물과의 착오로 논란이 되기도 하였다[2].



그림 1. 대표적인 맹독성 가로수 협죽도[3]  
Figure 1. A poisonous tree, oleander

<그림 1>에서 보이는 가로수는 제주도 관광지에서 흔하게 볼 수 있는 협죽도이다. 화려한 모습으로 관광객들의 이목을 끌지만 강한 독성으로 인해 피부에 접촉하기만 해도 피해를 입는 대표적인 맹독성 식물이다[4-6].

<그림 2>는 가정에서 키우는 실내 화초인 디펜바키아 식물이다. 이 식물은 실내 공기정화 및 습도 조절에 효과가 있다. 하지만 사람의 피부에 접촉을 하면 마비증상 및 가려움증을 유발할 수 있고 애완동물이 섭취할 경우 위염과 구토를 유발할 수 있는 독성 식물이다[7].



그림 2. 공기정화 식물로 알려진 식물인 디펜바키아[7]  
Figure 2. Defenbachia, a plant known as an air purifying plant

앞서 언급한 사고들을 방지하고 식물의 정보를 제대로 알 수 있도록 식물의 이미지를 이용하여 식물을 예측할 수 있는 딥러닝을 구현하였다. 딥러닝 시스템은 Pytorch[8]기반의 프레임워크인 Fast.ai[9]를 사용하여 구현하고 많은 이미지로

Pretrain된 ResNet50[10]모델을 사용했다. 우리는 많은 식물들이 주변에 있지만 가정에서 일어난 작은 사고부터 방지하고자 가정에서 키우는 실내 화초를 위주로 데이터베이스를 작성했다. 데이터베이스는 식물의 영문, 국문 이름과 해당 식물의 긍정적인 효과, 부정적인 효과를 정리했다. 부정적인 효과는 사람에게 미치는 효과와 가정에서 키우는 반려동물에게 미치는 효과까지 정리하여 반려동물을 키우는 가정에도 유용하게 쓰일 수 있도록 제작했다.

본 논문에서 사용한 컴퓨터 비전은 사람의 시각 영역을 컴퓨터로 구현한 것이다. 사람보다 뛰어난 성능을 보여줌으로써 의료와 환경 등 다양한 주제로 연구가 진행되고 있다. 박선 외 저자들은 Tensorflow 딥러닝 프레임워크를 사용하여 적조생물 이미지를 분류하였다[11]. 구글 이미지 분류 모델을 이용하여 이미지 분류를 하였으며 가장 높은 정확도는 89.5%를 보였다. 서영민 외 저자들은 본 논문에서 사용한 알고리즘인 CNN을 이용하여 암세포를 분류할 수 있는 연구를 진행하였고[12], 암세포를 정확하게 분류할 수 있게 임계치를 사용한 이미지 전처리를 제안하였다. 모바일 플랫폼에서 식물의 잎 이미지 검색 시스템을 구현한 논문이 있었는데[13], MPP 알고리즘을 개선하여 사용하였으며 식물의 잎을 촬영하여 식물의 이름을 알 수 있었다.

국내 포털사이트인 네이버에서 이미지 검색을 통해 식물의 이름을 검색하는 서비스를 공식 어플리케이션을 통해 제공하고 있다. 하지만 어플리케이션을 설치해야 서비스를 사용할 수 있는 번거로움이 있다. 본 논문은 사용하는 단말에 기본적으로 설치되어있는 웹 브라우저를 통해 구현한 기능을 사용할 수 있다.

본 논문의 구성은 다음과 같다. 제 2장에서는 머신러닝 프레임워크의 선택 과정, 제 3장에서는 우

리가 선택한 모델인 ResNet에 대해 서술한다. 제 4장에서 Fast.ai 프레임워크를 이용하여 모델을 직접 학습시킨 과정, 제 5장에서 학습된 모델을 이용하여 웹 어플리케이션을 구현하는 과정을 설명하고 제 6장에서는 결론을 기술한다.

## 2. 머신러닝 프레임워크 선택

머신러닝에 가장 많이 사용되는 파이썬 라이브러리는 Tensorflow[14]와 Pytorch 두가지를 꼽을 수 있다. Tensorflow는 구글에서 개발한 머신러닝 시스템이다. 주로 Python API를 제공하고 추가로 C/C++이나 자바스크립트 등의 여러가지 언어도 같이 지원한다. 그리고 Pytorch는 전신인 Lua 언어로 작성된 오픈소스 머신러닝 패키지 Torch를 기반으로 제작이 되었고 Facebook 인공지능 연구팀이 개발을 하였다. 현재는 Lua 언어로 제작이 된 Torch 패키지가 개발이 중단이 되고 파이썬 언어를 기반으로 하는 Pytorch를 지원한다.

딥러닝을 구현할 프레임워크를 선택할 때 Tensorflow와 Pytorch 두 가지를 고려하였는데, 본 논문은 Pytorch를 사용하여 딥러닝을 구현하였다. Pytorch는 간결한 문법과 빠른 구현으로 사용자가 학습하기에 편리하다. 또한 직관적인 인터페이스와 실시간으로 결과 값을 볼 수 있어 디버깅을 쉽게 할 수 있는 등의 여러 장점이 있다.

본 논문에서 사용하는 머신러닝 프레임워크인 Fast.ai는 Pytorch를 기반으로 이미지 분류 및 자연어 처리를 쉽게 구현할 수 있는 기능들을 제공한다. Fast.ai는 간단한 문법을 사용하여 데이터 분류 작업 및 모델을 생성할 수 있어 딥러닝을 구현하기에 편리하다.

## 3. 학습 모델

우리가 사용할 딥러닝 알고리즘인 CNN은 이미지나 텍스트 등의 데이터를 패턴으로 인식하여 문제를 풀이할 수 있다.

CNN알고리즘을 이용하여 이미지를 분류하는 과정은 입력된 이미지를 Convolution 과정, Max-Pooling 알고리즘 적용, Fully-connected network layer를 거치는데, 이러한 과정들을 더욱 많이 반복하여 이미지를 인식할 수 있게 만든다. 이미지를 처리하는 일련의 과정을 거친 하나의 구조를 층이라고 하는데, 층이 깊을수록 성능에 큰 영향을 미칠 수 있어 2012년에 제안된 AlexNet[15], 2014년에 제안된 VGGNet[16], 2015년에 제안된 GoogleNet[17] 등의 모델들은 네트워크 층을 깊게 구현하였다. 이 모델들은 시간이 지날수록 층을 더욱 깊게 만들어 성능을 향상시켰다.

하지만, 네트워크의 깊이를 깊게 할수록 성능이 향상이 되는 것인가에 대한 의문이 존재했다. 개발자들의 여러 번 시도 끝에 일정 층의 깊이까지 구현하면 성능이 향상되었지만 그 모델의 한계 이상으로 층을 깊게 만들 경우 Backpropagation현상으로 학습이 잘 되지 않는 문제인 Vanishing Gradient에 부딪혀 성능을 떨어뜨리는 원인이 되었다. 이것은 깊이가 깊은 모델에서 기존의 많은 학습이 진행된 경우 가중치들의 분포가 균등하지 않고 역전파시 기울기가 충분치 못해 학습을 안정적으로 할 수 없는 문제이다.

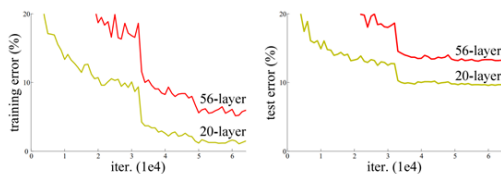


그림 3. 같은 구조의 네트워크를 20층, 56층으로 쌓아 성능을 비교한 그래프이다. [17]

Figure 3. This is a graph comparing the performance by stacking the network with 20 layers and 56 layers.

<그림 3>에서 보이는 왼쪽의 그래프는 Training error에서 20개의 층을 가진 layer보다 56개의 층을 가진 layer가 error 확률이 훨씬 높은 것을 나타낸다. 오른쪽의 그래프는 Test error에서 20개의 층을 가진 layer보다 56개의 층을 가진 layer가 error 확률이 높다. 두개의 그래프에서 20층의 layer를 가진 네트워크가 56층의 layer를 가진 네트워크보다 전체적으로 성능이 비교적 좋다는 결과를 보아 층을 깊게 만들수록 성능이 반드시 높아지지 않다는 것을 증명한다.

Resnet 모델에서 이러한 단점을 해결해 줄 수 있는 가장 핵심적인 구조는 Residual learning이다. 기존의 네트워크에 skip connection을 추가한 단순한 구조이지만 VGG 모델에 비교하면 연산량이 줄어들었고, layer를 깊게 만들어도 Vanishing Gradient 현상이 일어나지 않아 학습이 보다 원활하게 되고 좋은 성능을 보였다. Skip connection은 일종의 지름길을 만들어 Gradient가 잘 흐를 수 있도록 한다.

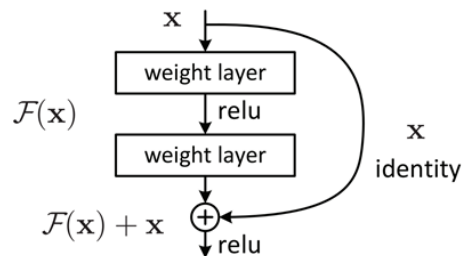


그림 4. Vanishing Gradient를 해결하기 위해 skip connection을 추가한 ResNet의 핵심 구조[17]

Figure 4. It is the core structure of ResNet that added skip connection to solve vanishing gradient.

본 논문은 ImageNet에서 pretrain된 모델인 ResNet50 모델을 사용한다. 새로운 모델을 만들기 위해서는 model parameters, layers 구조 등의 여러 가지 고려해야할 사항이 많다. 또한 문제점이 발생할 경우 이를 해결하기 쉽지 않고 처음부터 이미

지를 학습시키는데 많은 시간을 필요로 한다. 우리는 이러한 문제점을 방지하기 위해 ImageNet에서 많은 이미지를 이용하여 학습이 되었고 공개적인 결과들을 통해 안정적으로 검증이 된 모델을 사용하려고 한다.

검증이 된 ResNet모델을 사용하더라도 이미 학습시켜진 약 1000가지 이상의 이미지 데이터와 우리가 해결하고자 하는 이미지 데이터가 같지 않다. Pretrain된 모델을 우리가 해결하고자 하는 문제에 적용하기 위해 불러온 시점의 model parameters, learning rate 등을 수정한 Transfer Learning기법을 사용한다.

#### 4. 딥러닝 모델 학습

Fast.ai 프레임워크를 사용하여 모델을 학습시킨 과정은 <그림 5>와 같다.

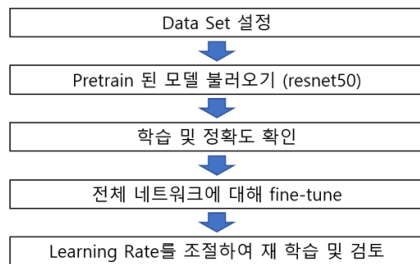


그림 5. Fast.ai를 사용한 모델 학습 과정  
Figure 5. Model training process using Fast.ai

#### 4.1 이미지 데이터 준비

실내 화초 분류를 위한 Data Set의 구성은 일반적으로 가정에서 쉽게 볼 수 있는 Aglaonema, Alocasia, Begonia, Bromeliaceae, Calathea, Cattleya, Dendrobium, Narcissus 등의 61 종류로 선정하였다. 선정된 식물들은 웹 크롤링으로 데이터

를 저장하였다. 저장된 이미지 데이터 중 사람이나 물체가 사진의 1/3이상 차지한 경우 등의 올바르지 않은 이미지는 제외하였다. <그림 6>에서 보이는 알로에 식물 데이터 세트처럼 식물과 관련되지 않은 정보를 모두 제외하고 식물의 특징이 명확하게 드러난 이미지를 사용한다. 학습 데이터는 61종의 식물, 총 27,785개의 식물 이미지를 사용했고, 각각의 이미지는 Data Augmentation과정을 거쳤다.

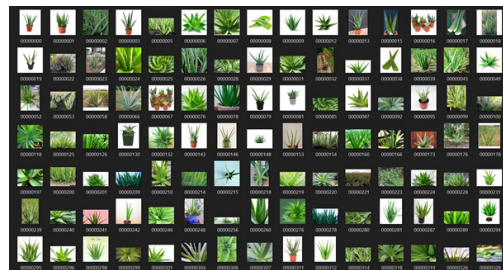


그림 6. 가정에서 키우는 대표적인 식물인 알로에 이미지 데이터  
Figure 6. Aloe image data, a plant grown at home.

Data Augmentation은 기존 데이터의 파라미터 값을 조정하여 가지고 있는 이미지 데이터를 새롭게 생성하는 기법이다. 본 논문에서는 Scaling (원본 이미지를 확대, 축소), Flipping (원본 이미지 좌우 대칭, 상하 대칭), Rotation (원본 이미지를 기울이기), Brightness(원본 이미지의 휘도를 조절)를 식물 이미지에 무작위로 적용하여 학습을 진행하였다.

#### 4.2 모델 학습

모델의 학습 과정별 정확도를 <그림 7>에서 보인다. train\_loss는 모델을 학습시키기 위해 선별한 데이터 셋의 학습 과정 진행 중 이미지를 올바르게 판단하지 못한 확률이다. valid\_loss는 모델을 검증시키기 위해 선별한 데이터 셋의 이미지를 올바르게 판단하지 못한 확률이다. error\_rate는 학습한 모델이 올바르게 예측을 하지 못한 오류 확률이다.

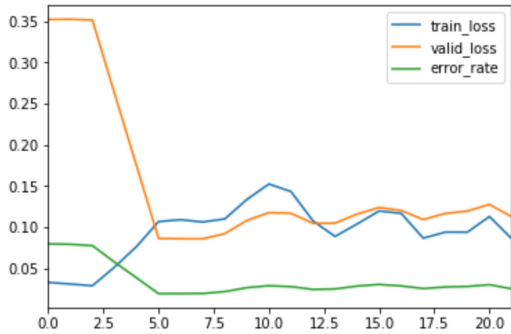


그림 7. epoch별 학습 모델 정확도 그래프. X축은 epoch를 나타내고 Y축은 각 항목별 rate를 나타낸다.

Figure 7. Graph of training model accuracy by epoch. The X axis represents the epoch and the Y axis represents the rate for each item.

모델을 학습시키는데 5번의 learning rate 조절 과정을 거쳤다. 처음 5번의 epochs는 기존 모델을 불러와 이미지 데이터를 학습시켜주는 과정이므로 train loss보다 valid loss가 훨씬 높은 것을 보인다. 이후 전체 네트워크에 대해 fine-tune을 적용하여 valid loss가 줄어드는 모습을 보이고, train loss는 잠시 늘었지만 점차 줄어드는 것을 보인다. 총 21번의 모델 학습을 진행하였고 error rate가 학습이 진행되어도 늘어나지 않은 것으로 보아 Overfitting 현상이 일어나지 않았다. Overfitting이란 모델 훈련을 위한 데이터를 너무 정확하게 학습이 되어 새로운 이미지 데이터가 개입이 될 경우 정확도가 급격하게 낮아지는 현상을 말한다.

### 5. 웹 어플리케이션 구현

웹 어플리케이션은 python의 웹 프레임워크인 Flask를 사용하여 구현하였다. 웹은 HTML5 표준으로 작성하였으며, HTML5를 지원하는 모든 브라우저로 동작함을 확인하였다.

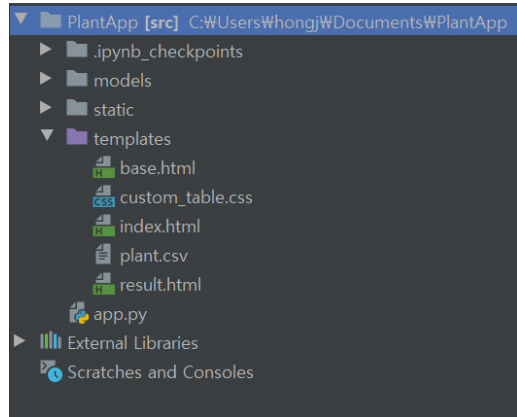


그림 8. 웹 어플리케이션 파일 구조

Figure 8. File structure of web application

<그림 8>은 웹 어플리케이션 동작을 위한 파일 구조를 나타낸다. 최상위 폴더의 app.py 파일에서 Flask 웹 어플리케이션이 동작하며, 웹은 base.html 파일을 기반으로 메인화면인 index.html, 결과를 보여주는 화면인 result.html 파일로 이루어져 있다. 서버를 실행하면 클라이언트 내부의 데이터베이스에서 식물 정보를 불러와 웹 어플리케이션이 작동한다.



그림 9. 웹 어플리케이션의 메인 화면

Figure 9. Main screen of a web application.

웹 어플리케이션을 실행하면 <그림 9>의 메인 화면이 나타난다. Browse 버튼으로 사진을 촬영하거나 이미지 파일을 업로드한다. 예측하기 버튼을 누르면 웹 어플리케이션에 업로드 된 이미지 파일을 서버로 전송한다. 4장에서 Pretrain한 Resnet50 모델이 이미지 파일로부터 식물의 이름을 예측한

다. 모델에 의해 예측된 식물 이름과 일치하는 정보를 구축된 데이터베이스에서 찾아 웹 페이지에 표시한다.

<그림 10>은 최종 결과를 보여주는 웹 페이지이다. 학습 데이터 세트에 포함이 되지 않은 디펜바키아 이미지를 넣어 정상적으로 출력됨을 확인할 수 있다. 결과 웹페이지는 예측한 식물의 이름과 입력된 이미지를 보여주고 아래에 그 식물에 대한 정보를 표시해준다. 식물의 국문명과 영문명, 긍정적인 효과와 부정적인 효과 등을 데이터베이스에서 불러와 웹 페이지에 출력한다.

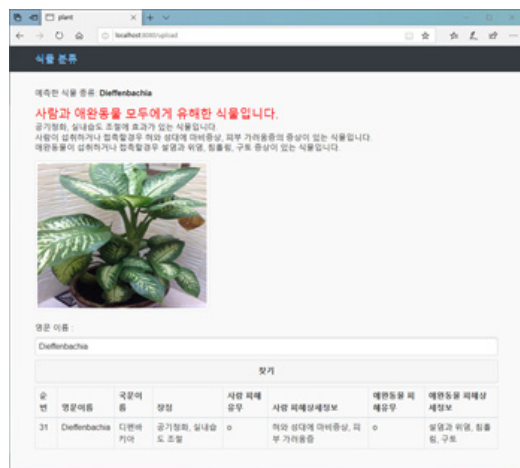


그림 10. 예측된 식물의 이름과 정보를 알려준다.

Figure 10. Provide the name and information of the predicted plant.

## 6. 결론

Pytorch기반의 Fast.ai 프레임워크와 ResNet50 모델을 기반으로 실내에서 기르는 식물을 분류할 수 있는 딥러닝을 구현하였다. ResNet50모델은 기존 학습되어진 파라미터 값과 현재 풀이하고자 하는 문제와 다르므로 전체 네트워크에 대해 Fine-tune을 진행하였다. 또한 분류한 식물의 정보를 데이터

베이스로 작성하여 사람과 애완동물에 대한 정보를 제공할 수 있게 웹 어플리케이션을 제작하였다. 번거로운 설치 절차를 거쳐야하는 단말기 어플리케이션과 달리 각 단말에 내장되어있는 웹 브라우저로 구현한 어플리케이션에 접근할 수 있다. 모바일 기기에 맞게 최적화를 거쳐 장소에 구애받지 않고 어플리케이션을 사용할 수 있는 환경을 마련하였다. 웹 어플리케이션을 이용하여 쉽게 식물의 정보를 알 수 있을 것이라 기대된다[18].

본 논문에서 구현한 식물 정보 웹 어플리케이션은 이미지 파일을 서버로 전송하면, 서버에서 식물의 이름을 예측하여 데이터베이스 내의 일치하는 정보를 출력하는 프로그램을 작성하였다. 웹 어플리케이션을 사용하여 딥러닝으로 학습한 61종의 식물을 정확하게 예측할 수 있다. 하지만 현재 구현한 어플리케이션은 처리하려는 이미지 내에 여러 종류의 식물이 있는 경우 정확성이 떨어지는 경향이 있다. 즉, 하나의 이미지에 두 종류 이상의 식물이 동시에 포함된 이미지를 처리할 경우, 한 종류의 식물 정보만 출력되거나 다른 식물로 잘못 분류하는 경우가 있다. 이러한 단점을 개선할 수 있도록 두 가지 이상의 식물을 분류할 수 있도록 수정할 예정이다. 또한 영상에서 식물을 분류할 수 있도록 프로그램을 수정하고 빠른 속도와 편리성에 중점을 두어 웹 어플리케이션의 접근성을 더욱 높일 수 있도록 개선할 예정이다.

## References

- [1] Naver news, <https://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=103&oid=047&aid=0002187631>, Apr. 2018.
- [2] Naver news, <https://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=103&oid=047&aid=0002187631>

- SD&mid=sec&sid1=102&oid=088&aid=0000448272, Jun. 2016.
- [3] Nerium image, <https://ko.wikipedia.org/wiki/%ED%98%91%EC%A3%BD%EB%8F%84>, Sep. 2019.
- [4] Naver news, <https://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=102&oid=016&aid=0001333686>, Dec. 2017.
- [5] Naver news, <https://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=102&oid=055&aid=0000598900>, Dec. 2017
- [6] Naver news, <https://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=102&oid=022&aid=0003238234>, Dec. 2017.
- [7] Dieffenbachia image, <https://en.wikipedia.org/wiki/Dieffenbachia>, Sep. 2019.
- [8] Pytorch, <https://pytorch.org/>, Sep. 2019.
- [9] Fast.ai, <https://www.fast.ai>, Sep. 2019.
- [10] K. He, X. Zhang, S. Ren and J. Sun, *Deep residual learning for image recognition*, 2015 IEEE Conference on Computer Vision and Pattern Recognition, Dec. 2016.
- [11] S. Park and J. W. Kim, *Red tide alga image classification using deep learning based open source*, Smart Media Journal, Vol. 7, No.2, Jun. 2018.
- [12] Y-M Seo and J-K Han, *Deep learning algorithm to identify cancer pictures*, Journal of Broadcast Engineering, Vol. 23, No. 5, Sep 2018.
- [13] S. C. Kim, Y. Y. Nam and I. J. Hwang, *Implementation of plant leaf image retrieval system over mobile platform*, Journal of Computing Science and Engineering, Jul. 2015.
- [14] Tensorflow, <https://www.tensorflow.org/>, Sep. 2019.
- [15] A. Krizhevsky, I. Sutskever and G. E. Hinton, *ImageNet classification with deep convolutional neural networks*, NIPS Conference 2012.
- [16] K. Simonyan, and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, International Conference on Learning Representations, Apr. 2015.
- [17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, 2015 IEEE Conference on Computer Vision and Pattern Recognition, Oct. 2015
- [18] H. J. Shin, S. I. Lee, H. W. Jeoung, J. W. Park, *House plants classification and information web using deep learning*, 2019 Autumn Conference on KOREA SOCIETY OF IT SERVICES, Nov. 2019
- 
- 딥러닝을 활용한 실내 식물 이미지 분류 및 식물 정보 제공 웹 어플리케이션**
- 신홍재<sup>1</sup>, 이소인<sup>1</sup>, 정희원<sup>1</sup>, 박장우<sup>2</sup>
- <sup>1</sup>순천대학교 정보통신공학전공 학부생  
<sup>2</sup>순천대학교 정보통신공학전공 교수
-

## 요 약

식물은 공기정화와 조경 등의 좋은 효과를 주지만, 자신을 보호하기 위해 특별한 성분을 가진다. 식물 자신을 보호하기 위해 만든 성분이 사람이나 동물에게 피해를 줄 수 있다. 또한 식물 생김새가 유사한 다른 식물로 오인하여 발생한 사고들이 있다. 우리는 식물을 구분할 수 있고 각 식물들의 정보를 보여줄 수 있는 프로그램을 구현했다. 이미지를 분류할 수 있는 딥러닝을 사용하여 식물의 이름을 예측하고, 예측한 식물에 일치하는 정보를 보여줄 수 있는 웹 어플리케이션을 제작하였다. 61가지의 실내 식물을 구글 이미지 검색을 이용하여 데이터 세트를 제작했다. 실내 식물의 긍정적인 효과와 부정적인 독성을 데이터베이스에 정리하였다. 딥러닝은 Pytorch기반의 프레임워크인 fast.ai를 사용하여 구현하였다. Data Augmentation을 거쳐 학습할 이미지 수를 증가시켰다. 다양한 이미지를 이용하여 Pretrain된 모델인 ResNet50을 사용하여 실내 식물 이미지 데이터를 학습했다. 모델의 정확도는 약 97.5%로 대부분의 식물들을 정확하게 예측할 수 있었다. Python기반의 웹 프레임워크인 flask를 활용하여 웹 어플리케이션을 구현하였다. 구현된 이미지 분류 딥러닝 모델을 이용하여 식물의 이름을 예측하고, 예측된 식물 이름에 맞는 정보를 웹 페이지에 출력해준다. 웹 어플리케이션을 모바일 기기에 최적화하여 편리하게 사용할 수 있다.



**Hong-Jae Shin** received a bachelor's degree in the Information and Communication Engineering from Suncheon National in 2020. His current research

interests include artificial intelligence, statistics, economics.

*E-mail address:* hongjae6@naver.com



**So-in Lee** is attending Suncheon National University to obtain a bachelor's degree in information and communications engineering. From 2016 to 2020, I majored in Information and Communication Engineering at Suncheon National University.

*E-mail address:* thdls122@naver.com



**Hui-won Jeung** is attending Suncheon National University to obtain a bachelor's degree in information and communications engineering. From 2016 to 2020, I majored in Information and Communication Engineering at Suncheon National University.

*E-mail address:* bliss\_1026@naver.com



**Jangwoo Park** received the B.S., M.S. and Ph.D. degrees in Electronic engineering from Hanyang University, Seoul, Korea in 1987, 1989 and 1993, respectively. In 1995, he joined the faculty member of the Suncheon National University, where he is currently a professor in the Department of Information & Communication engineering. His research focuses on deep learning and machine learning.

*E-mail address:* jwpark@sunchon.ac.kr