

## 샤인머스캣과 청포도 자동 분류 딥러닝 시스템 개발

김명진<sup>1</sup> · 한영석<sup>2</sup> · 김승우<sup>3</sup> · 김희숙<sup>4\*</sup><sup>1,2,3</sup>한국폴리텍V대학 AI융합과(하이테크과정)<sup>4\*</sup>전남대학교 공과대학 소프트웨어공학과

## Development of Automatic Classification Deep Learning System for Shine-muscat and Green-grape

Myung-Jin Kim<sup>1</sup> · Young-Seok Han<sup>2</sup> · Seung-Woo Kim<sup>3</sup> · Hye-Suk Kim<sup>4\*</sup><sup>1,2,3</sup>Undergraduate course, Department of AI Convergence(high-tech course), Korea Polytechnic V, Haseo-ro 85, Korea<sup>4\*</sup>Department of Software Engineering, Chonnam National University, 77 Yongbong-ro, Gwangju, Korea

### [요약]

본 논문에서는 유사한 특징을 가진 샤인머스캣과 청포도의 자동 분류를 위한 인공지능 시스템을 개발하기 위해서 두 과일의 이미지 데이터셋과 특징 기반의 수치 데이터셋을 구축하였다. 구축된 이미지 데이터셋을 대상으로 2가지 CNN 모델을 적용하고 검증 데이터를 활용하여 자동 분류의 정확도를 산출하였다. 그 결과 VGGNET 모델은 25%, 과적합을 최소화하기 위해서 제안된 CNN 딥러닝 모델은 84%의 정확도를 나타냈다. 그리고 특징 기반의 수치 데이터셋을 대상으로 선형회귀 딥러닝 모델을 적용한 경우에는 자동 분류의 정확도가 94%로 나타났다. 본 논문에서 제안한 데이터셋 구축 방식과 딥러닝 방식을 서로 다른 품종의 과일이나 농산물 자동 분류에 적용하여 인공지능 기반의 스마트 팜 시스템 구현에 응용할 수 있을 것으로 기대한다.

### [Abstract]

In this paper, in order to develop an artificial intelligence system for automatic classification of shine muscats and green grapes with similar characteristics, an image dataset of the two fruits and a feature-based numerical dataset were constructed. Two CNN models were applied to the constructed image dataset and the accuracy of automatic classification was calculated using verification data. As a result, the VGGNET model showed 25% accuracy, and the CNN deep learning model proposed to minimize overfitting showed 84% accuracy. In addition, when the linear regression deep learning model was applied to feature-based numerical datasets, the accuracy of automatic classification was 94%. It is expected that the dataset construction method and deep learning method proposed in this paper can be applied to the implementation of artificial intelligence-based smart farm systems by automatically classifying fruits or agricultural products of different varieties.

**색인어** : CNN, 분류, 딥러닝, 인공지능, 스마트 팜, 샤인머스캣, 청포도**Keyword** : CNN, Classification, Deep Learning, Artificial Intelligence, Smart Farm, Shine-muscat, Green-grape<http://dx.doi.org/10.9728/dcs.2021.22.10.1637>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 18 August 2021; Revised 07 September 2021

Accepted 27 September 2021

\*Corresponding Author; Hye-Suk Kim

Tel: +82-62-530-0147

E-mail: idishskim@naver.com

## I. 서론

1993년 12월 15일 우루과이 라운드(UR:Uruguay Round) 협상이 타결됨으로써 한국의 글로벌 농수산물 무역이 본격화되었다. 쌀 개방으로 시작하여 2004년에는 한국과 칠레 자유 무역 협정(FTA; Free Trade Agreement)이 확대되었고 이후 농산물 수입개방은 견잡을 수 없는 상황으로 확장되었다. 또한 코로나바이러스감염증-19(COVID-19) 직전까지 전 세계적인 무역 및 관광, 기술 등 다양한 산업 분야의 활발한 교류 속에서 전 세계의 농수산물 무역 또한 급진적인 발전을 이룰 수 있었다. 포스트 코로나 시대에 직면하면서 바이러스의 영향으로 사람 대 사람 간의 교류는 현저히 줄었지만, 농수산물 및 상품과 기술에 대한 무역은 여전히 왕성하게 진행되고 있다. 이렇게 전 세계의 다양한 토종에서 자랐지만 비슷하게 생긴 농수산물이 무작위로 국내에 들어오면서 국내의 토종 제품들과의 경쟁이 심해지고 있으며, 소비자로서도 국산과 외국산의 비교가 육안으로는 불가능한 상황에 이르렀다. 생김새가 비슷한 다양한 품종들에 대한 정확한 데이터셋을 확보해 나가는 것이 국내 농수산물 품질 향상과 기존의 농수산물을 보호하기 위해서도 필요할 것으로 판단된다.

포스트 코로나 시대에 인간의 야외 활동은 제약받게 되었으며 본격적인 ‘언택트 시대’가 열리게 되었고 이로 인해 사람의 손을 거치지 않고 해결할 수 있는 기계 중심, 컴퓨터 중심 업무의 중요성이 다시 한번 강조되었다. 그 결과, 기계학습(Machine Learning) 및 딥러닝(Deep Learning) 분야로 이목이 다시금 집중되고 있다[1]. 딥러닝은 컴퓨터가 스스로 데이터로부터 학습하는 네트워크 모델을 기반으로 텍스트, 이미지 등을 분류하고 축적된 데이터를 바탕으로 자동으로 규칙을 찾고 학습을 하고 있다. 이러한 딥러닝은 21세기로 들어서서 컴퓨터 사양과 인터넷 통신의 발달이 만들어낸 무한대의 빅데이터 그리고 고성능 GPU(Graphic Processing Unit)가 등장함에 따라 자동 분류 시스템 분야에서 성능이 더욱 향상되었다.

본 연구에서는 과일 중에서 특징이 유사한 샤인머스캣과 청포도의 자동 분류를 위해서 이미지 데이터셋과 특징 기반의 수치 데이터셋을 구축하고, 구축된 데이터셋을 기반으로 두 종류의 과일 특징을 비교 분석한 후 분석 결과를 중심으로 다중 분류(Multi Classification)[2-7] 딥러닝을 실행하여 인공지능 기반의 자동 분류 시스템을 개발하고자 한다.

본 논문의 2장에서는 이미지 분류에 활용된 기존의 딥러닝 방식을 살펴보고, 3장에서는 샤인머스캣과 청포도의 자동 분류를 위해서 본 연구에서 제안하는 이미지 데이터셋 구축 방식과 딥러닝 방식을 소개한다. 이어서 4장에서는 샤인머스캣과 청포도의 특징을 수치 데이터셋으로 구축하여 딥러닝을 수행한 방식을 소개한다. 5장에서는 3장에서 소개한 이미지 데이터셋을 기반으로 실행한 딥러닝 모델과 4장에서 소개한 수치 데이터셋 기반의 딥러닝 모델의 자동 분류에 관한 손실(Loss)과 정확도(Accuracy)를 비교하고 분석한다. 마지막으로 6장에서는 본 연구의 결론과 향후 연구 방향에 관해 서술한다.

## II. 관련 연구

### 2-1 CNN

1998년 개발된 CNN(Convolutional Neural Network) 방식은 각각의 레이블로 선행 분류된 대량의 이미지 데이터를 이용하여 다양한 형태의 객체 특징을 학습할 수 있으며, 학습된 인공신경망은 또다시 전송 학습(Transfer Learning)을 통해 새로운 인식 작업을 재학습할 수 있다. 최근 학습 결과의 정확성이 더욱 높아지고 있어서 이미지 인식 분야에 널리 이용되고 있다[1].

CNN 이전의 신경망은 이미지들을 인식할 때 해당 이미지들의 픽셀값을 그대로 가져와 분류하고 인식하기 때문에 이미지를 조금만 변형시켜도 이미지를 인식하는 데 어려움이 있었다. 반면, CNN은 동물의 시각 처리 과정을 모방하여 이미지 패턴의 크기, 위치가 바뀌어도 인식할 수 있는 장점이 있어서 이미지 인식 분야에서 좋은 성능을 나타내고 있다[1][2][8].

그림 1은 컴퓨터 비전 분야에서 활용되고 있는 다양한 예지 검출 알고리즘을 이용하여 이미지의 특징을 예지로 표현한 것이다[8]. CNN의 경우, 합성곱(Convolution) 연산을 통해 특징 맵(Feature Maps)을 구성하여 그림 1의 예지에 해당하는 이미지의 특징을 추출하고 이를 이미지 인식 및 분류에 활용하고 있다.

CNN은 그림 2와 같이 이미지의 특징을 추출하는 부분과 지도 학습을 수행하여 이미지를 인식하는 부분으로 나누어 설계된다. 이미지의 특징을 추출하는 과정에서 이미지 특징 추출을 위한 가중치 필터를 적용하여 합성곱 연산이 적용된 후, 활성화 함수(Activation Function)를 적용한다. 이 경우, 다양한 가중치 필터를 일정 간격으로 이동해가며 입력 데이터에 적용하면서 최적화된 딥러닝 모델을 찾아야 한다.

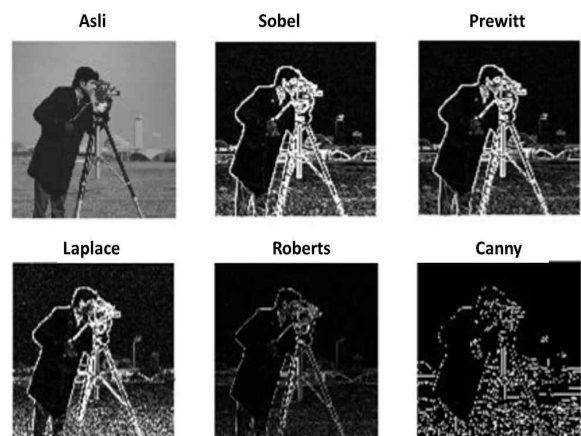


그림 1. 다양한 필터를 적용한 이미지 예시  
Fig. 1. Example of various image filter

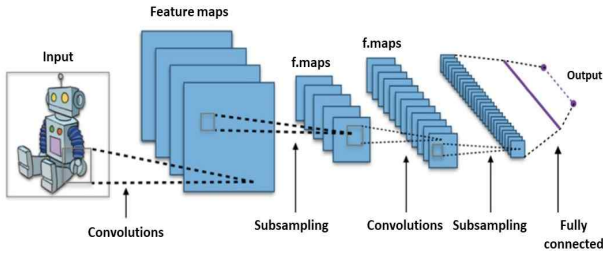


그림 2. 합성곱 신경망 모델의 구조  
Fig. 2. Model structure of Convolutional Neural Network

합성곱 연산으로 얻어진 Convolution Layer의 데이터 크기를 축소하기 위해서 Pooling Layer를 사용하고, Fully Connected Layer를 통해서 1차원 벡터 형식의 데이터로 변환한다. 변환된 1차원 벡터값들은 Softmax 등의 다중 분류 함수를 이용하여 이미지 인식 및 분류를 수행하게 된다.

### Ⅲ. 이미지 데이터셋 기반의 자동 분류 딥러닝

#### 3-1 이미지 데이터셋 구축

본 연구에서는 CNN 딥러닝 알고리즘을 이용하여 샤인머스캣과 청포도 이미지를 자동 분류하기 위해서 직접 촬영한 이미지와 웹상에서 수집한 이미지를 모두 포함하여 이미지 데이터셋을 구축하였다.

직접 촬영한 이미지의 경우, 샤인머스캣은 국내에서 생산되는 김천의 샤인머스캣을 활용하였고 청포도는 씨 없는 청포도인 톨슨 시들리스 품종을 구매하여 다양한 각도로 촬영을 진행하여 샤인머스캣과 청포도 이미지를 각각 500장씩 확보하였다. 확보된 이미지를 기반으로 CNN 딥러닝 수행이 가능한지 확인하기 위해서 그림 1에서 제시한 에지 검출 알고리즘을 이용하여 샤인머스캣과 청포도 이미지의 특징 추출을 시도하였다. 그림 3은 샤인머스캣 이미지의 특징을 추출한 결과이며 매우 선명하게 이미지의 특징이 잘 드러나고 있음이 확인되었다.

웹상에서 샤인머스캣과 청포도 이미지 수집은 크롬 드라이버를 사용하여 구글 및 네이버 포털을 대상으로 크롤링을 수행하여 각각 26장씩 수집하였다. 수집된 26장의 이미지는 시계방향 및 반시계방향으로 회전 또는 좌우 대칭 등의 이미지 재구성 방법[9-12]을 적용하여 각각 260장으로 확장되었다.

그림 4는 이미지 재구성 방법으로 확장된 샤인머스캣 이미지의 일부분을 보여준다. 본 연구에서는 이처럼 촬영된 이미지, 수집된 이미지 그리고 재구성된 이미지를 모두 포함하여 딥러닝에 활용할 수 있는 이미지 데이터셋을 구성하였다.

#### 3-2 기존 연구의 CNN 모델을 이용한 방식

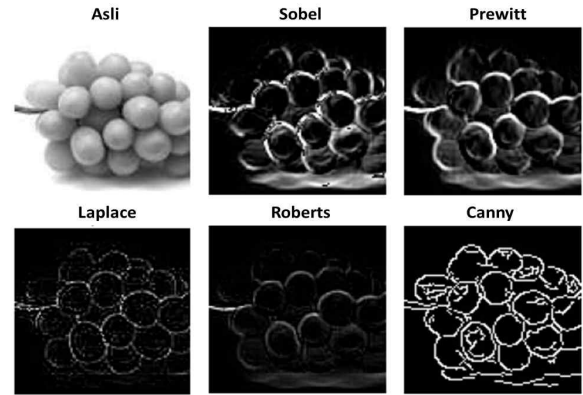


그림 3. 샤인머스캣 이미지의 에지 검출 결과  
Fig. 3. Edge detection result of Shine-muscat image

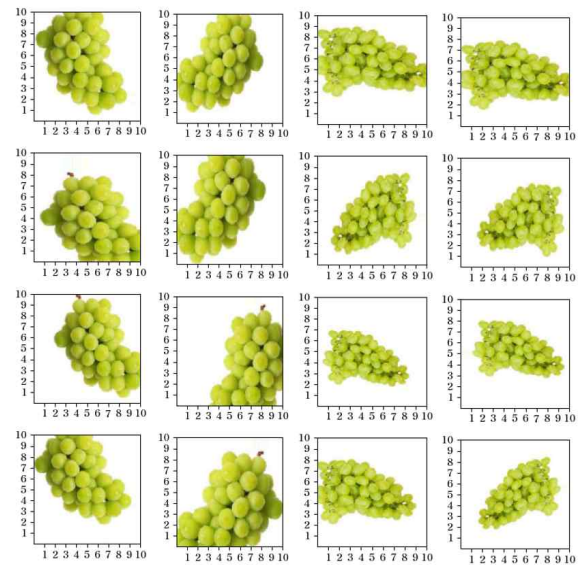


그림 4. 재구성으로 확장된 샤인머스캣 이미지 데이터셋  
Fig. 4. Shine-muscat image data set extended by reconstruction

먼저 본 논문에서는 오이 등급 자동 분류를 위한 CNN 기반의 기존 연구[13]에서 제안한 VGGNET 딥러닝 모델 구조를 샤인머스캣과 청포도 자동 분류를 위한 딥러닝 모델로 적용하였다. 그림 5는 본 논문에 적용된 VGGNET 딥러닝 모델 구조를 보여준다.

표 1과 그림 6은 기존 연구[13]에서 제안된 VGGNET 모델로 딥러닝을 실행하여 산출된 손실과 정확도에 대한 결과를 나타낸다. 오이 등급 분류를 위해 제시한 기존 연구[13]의 VGGNET 모델을 이용하여 샤인머스캣과 청포도의 이미지 분류를 시도해 보았으나, 학습 데이터의 경우 58%, 검증 데이터의 경우 25%의 학습 정확도가 출력되었다. 학습 데이터와 비교해 검증 데이터의 학습 정확도가 현저하게 떨어지고 있는 상태가 발생하였으며 그 원인을 과적합(overfitting)으로 추론할 수 있다.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 64)	1792
conv2d_1 (Conv2D)	(None, 128, 128, 64)	36928
max_pooling2d (MaxPooling2D)	(None, 64, 64, 64)	0
conv2d_2 (Conv2D)	(None, 64, 64, 128)	73856
conv2d_3 (Conv2D)	(None, 64, 64, 128)	147584
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 128)	0
conv2d_4 (Conv2D)	(None, 32, 32, 256)	295168
conv2d_5 (Conv2D)	(None, 32, 32, 256)	590080
conv2d_6 (Conv2D)	(None, 32, 32, 256)	590080
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 256)	0
conv2d_7 (Conv2D)	(None, 16, 16, 512)	1180160
conv2d_8 (Conv2D)	(None, 16, 16, 512)	2359808
conv2d_9 (Conv2D)	(None, 16, 16, 512)	2359808
max_pooling2d_3 (MaxPooling2D)	(None, 8, 8, 512)	0
conv2d_10 (Conv2D)	(None, 8, 8, 512)	2359808
conv2d_11 (Conv2D)	(None, 8, 8, 512)	2359808
conv2d_12 (Conv2D)	(None, 8, 8, 512)	2359808
max_pooling2d_4 (MaxPooling2D)	(None, 4, 4, 512)	0
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 4096)	33558528
dense_1 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 2)	8194

Total params: 65,062,722  
Trainable params: 65,062,722  
Non-trainable params: 0

그림 5. VGGNET을 이용한 딥러닝 모델  
Fig. 5. Deep learning model using VGGNET

표 1. VGGNET 모델을 이용한 손실 및 정확도 평가  
Table 1. Evaluation of loss and accuracy using VGGNET model

epoch	loss	accuracy	val_loss	val_accuracy
0	1.65970	0.4878	0.7090	0.2500
1	1.6570	0.4878	0.7090	0.2500
2	0.6893	0.6098	0.7525	0.2500
3	0.6985	0.6154	0.8074	0.2500
4	0.6623	0.6341	0.7968	0.2500
5	0.6696	0.6154	0.8280	0.2500
6	0.6580	0.6341	0.8597	0.2500
7	0.6823	0.5854	0.8178	0.2500
8	0.6699	0.6098	0.8324	0.2500
9	0.6798	0.5854	0.8066	0.2500
(omit)				
90	0.6460	0.6667	0.8344	0.2500
91	0.6708	0.6098	0.8415	0.2500
92	0.6447	0.6585	0.8722	0.2500
93	0.7092	0.5366	0.8498	0.2500
94	0.6663	0.6154	0.8345	0.2500
95	0.6598	0.6341	0.8190	0.2500
96	0.6703	0.6098	0.7981	0.2500
97	0.6563	0.6410	0.8251	0.2500
98	0.6708	0.6047	0.8541	0.2500
99	0.6815	0.5897	0.8657	0.2500

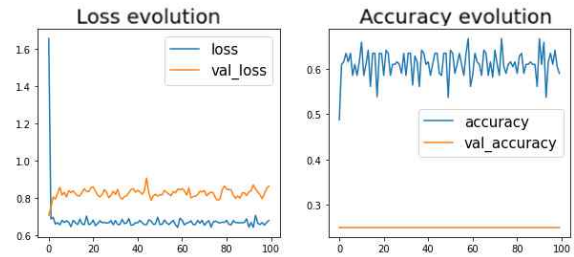


그림 6. VGGNET의 정확도와 손실도 그래프  
Fig. 6. Evaluation of loss and accuracy after VGGNET

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 127, 127, 32)	416
batch_normalization_2 (Batch Normalization)	(None, 127, 127, 32)	128
max_pooling2d_1 (MaxPooling2D)	(None, 63, 63, 32)	0
dropout_2 (Dropout)	(None, 63, 63, 32)	0
flatten_1 (Flatten)	(None, 127008)	0
dense_2 (Dense)	(None, 16)	2032144
batch_normalization_3 (Batch Normalization)	(None, 16)	64
dropout_3 (Dropout)	(None, 16)	0
dense_3 (Dense)	(None, 2)	34

Total params: 2,032,786  
Trainable params: 2,032,690  
Non-trainable params: 96

그림 7. 과적합 해결을 위해 제안된 CNN 딥러닝 모델  
Fig. 7. CNN deep learning model proposed for overfitting resolution

이에 본 연구에서는 그림 7과 같이 학습의 층을 축소하여 과적합을 해결할 수 있는 딥러닝 모델을 설계한다.

### 3-3 과적합 해결을 위해 제안된 CNN 모델을 이용한 방식

본 연구에서 그림 7과 같이 설계된 딥러닝 모델로 Binary Crossentropy를 적용하여 샤인머스켓과 청포도의 자동 분류를 위한 딥러닝을 수행하였다. 학습 데이터는 43 번째 에폭에서 83%의 정확도가 산출되었고 검증용 데이터는 40번째 학습부터 84%의 정확도가 산출되었다. 그림 7과 같은 CNN 모델로 기존 방식[13]의 과적합 문제를 해결할 수 있었으며 수행된 딥러닝 소스 코드는 아래와 같다.

```

model = Sequential()
model.add(Dense(6, input_dim = 5,
                activation = 'relu'))
model.add(Dense(5, activation = 'relu'))
model.add(Dense(2, activation = 'sigmoid'))
model.compile(loss = 'binary_crossentropy',
              optimizer = 'adam',
              metrics=['accuracy'])
    
```

```

hist = model.fit(x_train, y_train,
                validation_data=(x_test, y_test),
                epochs=50,
                batch_size=5,
                callbacks=[tensorboard_callback])
    
```

표 2. CNN 딥러닝 후 손실 및 정확도 평가

Table 2. Loss and accuracy evaluation after CNN deep learning

epoch	loss	accuracy	val_loss	val_accuracy
0	0.8630	0.4012	0.7270	0.4047
1	0.8630	0.4013	0.7270	0.4047
2	0.7014	0.4897	0.6587	0.6190
3	0.6734	0.5918	0.6448	0.6428
4	0.6594	0.6224	0.6232	0.6746
5	0.6408	0.6156	0.6032	0.6746
6	0.6253	0.6088	0.5854	0.6746
7	0.6102	0.7108	0.5700	0.7539
8	0.5969	0.7380	0.5569	0.7539
9	0.5838	0.7414	0.5476	0.7698
(omit)				
40	0.4355	0.7891	0.3977	0.8412
41	0.4319	0.8163	0.3955	0.8412
42	0.4302	0.7755	0.3944	0.8412
43	0.4167	0.8299	0.3846	0.8412
44	0.4278	0.8299	0.3803	0.8412
45	0.4252	0.8027	0.3831	0.8412
46	0.4204	0.8163	0.3752	0.8412
47	0.4190	0.8197	0.4741	0.7460
48	0.4192	0.8027	0.3787	0.8412
49	0.4151	0.8061	0.3700	0.8412

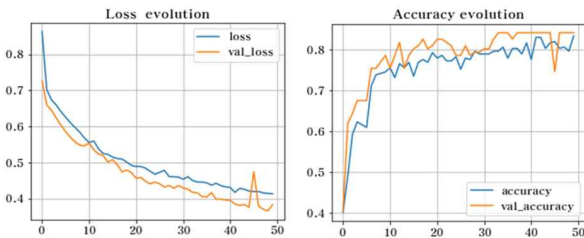


그림 8. 제안된 CNN 딥러닝 후 손실 및 정확도 시각화

Fig. 8. Visualization of loss and accuracy after CNN deep learning

표 2와 그림 8은 그림 7에서 제시한 과적합 해결을 위한 CNN 모델을 기반으로 딥러닝을 수행한 결과를 학습용 데이터셋과 검증용 데이터셋을 기반으로 손실과 정확도를 표현하고 있다.

그림 9는 과적합을 최소화하기 위해서 제안된 CNN 딥러닝 모델을 기반으로 딥러닝이 수행된 결과를 바탕으로 검증용 데이터셋을 대상으로 샤인머스캣과 청포도 이미지를 자동 분류한 결과이다. 자동 분류 결과는 원-핫 인코딩(One-Hot Encoding)[14-17]으로 표기되었고, [1,0]은 청포도 그리고 [0,1]은 샤인머스캣으로 분류되었음을 의미한다.

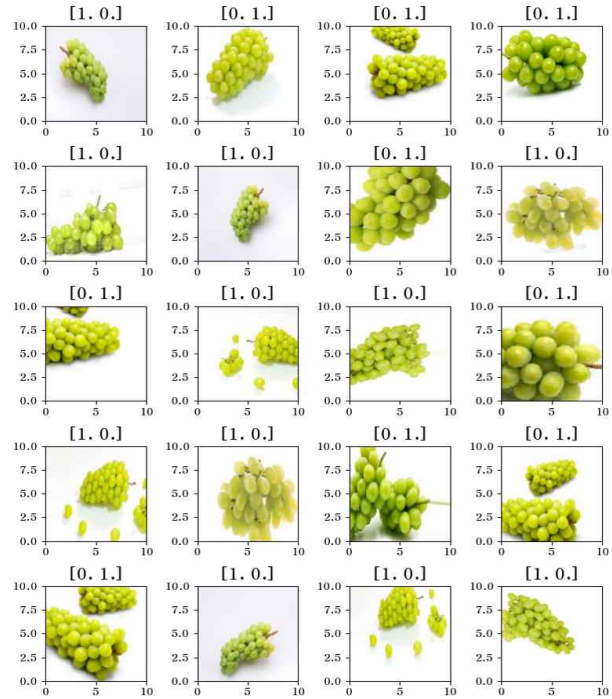


그림 9. 샤인머스캣과 청포도의 자동 분류

Fig. 9. Automatic classification of Shine-muscat and Green-grape

#### IV. 수치 데이터셋 기반의 자동 분류 딥러닝

사람의 육안으로도 청포도와 샤인머스캣을 구분하는 데에는 한계가 있기 때문에 과적합을 해결하였음에도 불구하고 이미지 기반의 CNN 딥러닝 실행 결과에서는 학습 정확도를 높이는 데 한계가 드러났다. 이에 본 연구에서는 더 높은 학습 정확도를 산출하기 위해서 준비된 이미지 데이터셋을 바탕으로 샤인머스캣과 청포도 1알의 특징을 수치 데이터셋으로 구축한 후 딥러닝을 수행하는 방식을 제안한다.

##### 4-1 특징 기반의 수치 데이터셋 구축

본 연구에서는 준비된 샤인머스캣과 청포도 이미지 데이터셋에서 각각 1알씩 100개를 확보하여 수평 및 수직 길이와 둘레, 무게를 수치 데이터로 산출하였다. 산출된 수치 데이터를 바탕으로 분석한 결과, 샤인머스캣과 청포도의 평균 수평 길이는 각각 4.66cm와 2.03cm로 청포도 1알보다 샤인머스캣 1알이 2.63cm 길게 나타났고, 샤인머스캣 1알의 평균 수직 길이는 2.87cm, 청포도 1알의 평균 수직 길이는 2.18cm로 샤인머스캣의 수직 길이가 평균적으로 0.69cm 길게 나타났다. 그리고 샤인머스캣과 청포도 1알의 평균 무게는 각각 9.50g과 6.63g으로 무게 또한 샤인머스캣의 1알이 2.87g 더 무겁게 측정되었다.

표 3. 샤인머스캣과 청포도의 특징 데이터셋

Table 3. Shine-muscat and Green-grape Characteristic Dataset

count	length		circumference		weight
	horizontal	vertical	horizontal	vertical	
0	1.8	2.5	6.6	8	7.41
1	2.1	2.4	7.3	8.2	8.71
2	2	2.3	7.3	8.1	8.8
3	1.4	1.7	5.6	6.1	3.57
4	1.9	2.4	6.9	8.15	7.61
5	2	2.3	7.1	7.6	7.49
6	2.1	2.4	7.5	8.3	8.81
7	1.9	2.3	6.9	7.4	7
8	2.1	2.6	7.4	8.3	9.05
9	1.8	2.3	6.7	7.6	6.91
(omit)					
90	2.5	3.01	8.1	9.05	11.52
91	2.39	3.16	7.5	8.9	9.05
92	2.4	3.5	7.9	9	10.32
93	2.4	3.1	7.7	9	9.49
94	2.35	2.98	7.6	8.62	8.68
95	2.3	2.9	7.15	8.3	7.71
96	2.39	2.82	7.71	8.58	9.19
97	2.4	2.7	7.6	8.2	8.68
98	2.29	2.65	7.4	8.1	7.61
99	2.3	3	7.5	8.7	8.99

길이와 둘레 그리고 무게는 청포도에 비하여 샤인머스캣이 훨씬 길고 무겁지만 각각 수직 및 수평 길이의 비율을 보면 청포도 1알의 수직 길이가 샤인머스캣에 비하여 더 길게 측정되고 있는 특징을 데이터셋 분석을 통해 확인할 수 있었다.

위와 같이 측정된 샤인머스캣과 청포도의 특징에 관한 수치 데이터셋을 표 3에 제시한다.

4-2 수치 데이터셋 기반의 딥러닝

샤인머스캣과 청포도 이미지의 특징이 수치로 표현된 데이터셋을 기반으로 데이터의 특징을 관찰하기 위해서 산점도 그래프로 시각화한 결과를 그림 10에서 보여준다.

산점도 그래프에서 보이는 것처럼 샤인머스캣과 청포도 이미지의 특징은 구분되고 있으며 이는 딥러닝으로 자동 분류가 가능함을 보여준다. 이에 본 연구에서는 샤인머스캣과 청포도의 특징을 수치로 표현한 데이터셋을 기반으로 선형회귀 딥러닝을 수행하고 그 결과를 표 4와 그림 11에 제시한다.

본 논문에서 구축한 특징 기반의 수치 데이터셋을 이용한 딥러닝 수행 결과, 검증용 데이터셋의 경우 손실은 낮추면서 94%까지 샤인머스캣과 청포도의 자동 분류 정확도를 높일 수 있었다.

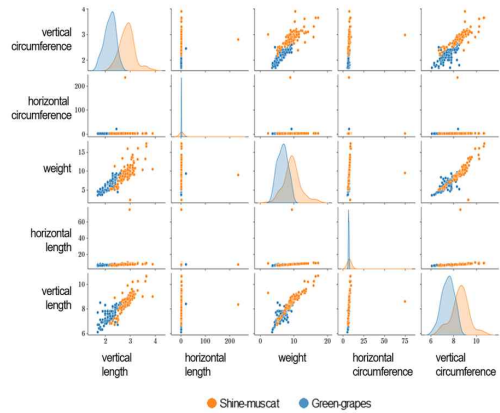


그림 10. 샤인머스캣과 청포도의 특징에 관한 산점도  
Fig. 10. A scatter plot of the characteristics of Shine-muscat and Green-grape

표 4. 수치 데이터 기반의 딥러닝 후 손실 및 정확도 평가

Table 4. Loss and accuracy evaluation after deep learning based on numerical data

epoch	loss	accuracy	val_loss	val_accuracy
0	0.9244	0.3010	0.7741	0.2770
1	0.9243	0.3010	0.7642	0.2778
2	0.7410	0.3010	0.6928	0.4222
3	0.6976	0.4272	0.6793	0.4778
4	0.6824	0.5728	0.6668	0.5889
5	0.6708	0.6165	0.6531	0.6444
6	0.6530	0.6748	0.6274	0.7333
7	0.6105	0.7767	0.5556	0.9333
8	0.5417	0.8495	0.4953	0.9222
9	0.4925	0.8301	0.4722	0.9111
(omit)				
40	0.2686	0.9029	0.3420	0.9333
41	0.2759	0.8883	0.3333	0.9333
42	0.2658	0.8835	0.3046	0.9222
43	0.2732	0.8786	0.3115	0.9222
44	0.2706	0.8932	0.2922	0.9222
45	0.2636	0.8932	0.2981	0.9222
46	0.2670	0.8883	0.2811	0.9222
47	0.2502	0.8981	0.3093	0.9333
48	0.2520	0.9029	0.2731	0.9444
49	0.2606	0.8932	0.2693	0.9444

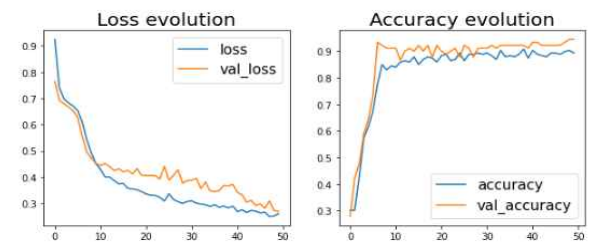


그림 11. 수치 데이터셋 기반의 딥러닝 후 손실 및 정확도 시각화  
Fig. 11. Visualization of loss and accuracy after deep learning based on numerical data

### V. 자동 분류 딥러닝 방식의 비교 분석

본 연구에서는 샤인머스켓과 청포도 자동 분류를 위한 딥러닝 시스템을 개발하기 위해서 구축된 이미지 데이터셋을 기반으로 오이 등급 분류를 위해 사용된 VGGNET 딥러닝 방식[13], 과적합 해결을 위한 CNN 방식 그리고 수치 데이터 기반의 딥러닝 방식으로 모델을 설계한 후 딥러닝을 수행하였다. 이렇게 수행된 딥러닝 결과를 그림 12와 13에 제시한다. 그림 12와 13에서 x축에 해당하는 n은 각 딥러닝 방식에 최적화된 에폭 단계를 의미한다.

이미지 데이터셋 기반의 기존 연구[13]의 딥러닝 방식은 샤인머스켓과 청포도의 자동 분류에서 가장 낮은 성능을 보였고, 과적합 해결을 위해 본 연구에서 제안된 CNN 방식은 검증용 데이터의 경우 자동 분류의 정확도가 84%까지 산출되었다. 반면 샤인머스켓과 청포도의 특징을 수치 데이터셋으로 구축한 후 이를 기반으로 딥러닝을 수행한 경우는 손실은 낮추면서 94%까지 자동 분류의 정확도를 높일 수 있었다.

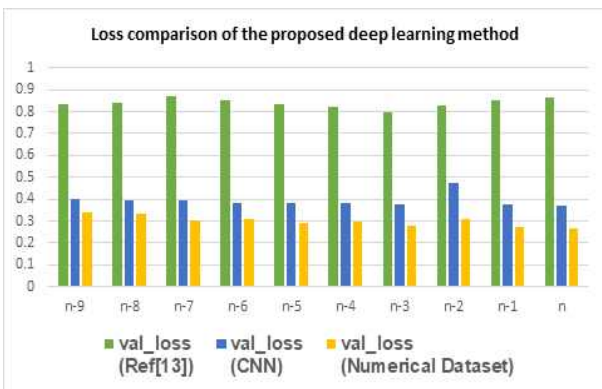


그림 12. 제안된 딥러닝 방식들의 손실 결과 비교  
 Fig. 12. Comparison of loss results of the proposed deep learning methods

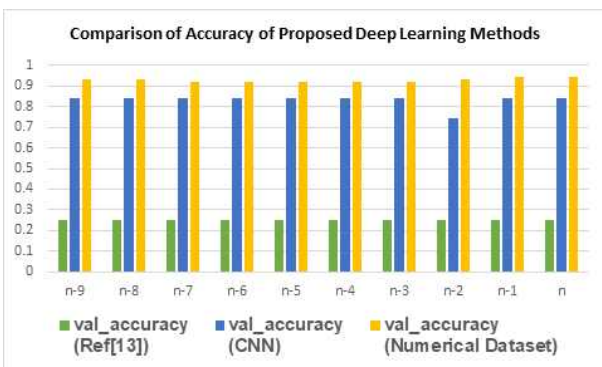


그림 13. 제안된 딥러닝 방식들의 정확도 결과 비교  
 Fig. 13. Comparison of accuracy results of proposed deep learning methods

수치 데이터셋을 이용한 경우 자동 분류의 정확도가 더 높게 나타난 이유는 사람의 육안으로도 청포도와 샤인머스켓을 구분하는 데에는 어려움이 있기 때문에 각 과일의 특징이 수치화된 데이터셋 기반의 딥러닝이 더 높은 정확도를 나타낸 것으로 분석된다.

### VI. 결론

본 연구에서는 샤인머스켓과 청포도를 자동 분류하기 위해서 이미지 데이터셋과 수치 데이터셋을 구축하였다. 구축된 데이터셋 분석을 통해서 샤인머스켓과 청포도가 같은 품종이지만 샤인머스켓이 청포도 보다 수평 및 수직 길이와 둘레가 더 길고, 무겁다는 것이 분석되었다. 본 논문에서 제안한 이미지 데이터셋 구축 방식 및 수치 데이터 방식은 비슷한 특징을 가진 과일 또는 농수산물의 자동 분류를 위한 데이터셋을 구축하기 위한 방식으로 응용이 가능할 것으로 예상된다.

구축된 이미지 데이터셋을 대상으로 본 연구에서는 기존 방식[13]을 기반으로 과적합을 해결하기 위한 CNN 딥러닝 방식과 수치 데이터셋을 대상으로 선형회귀 기반의 딥러닝 방식을 적용하여 비교 실험을 진행하였다. 그 결과, 이미지 데이터셋 보다는 두 과일의 특징을 수치로 표현한 데이터셋을 대상으로 딥러닝 수행한 결과가 자동 분류에서 정확도가 더 높게 나타났다. 사람의 육안으로도 청포도와 샤인머스켓을 구분하는 데에는 어려움이 있기 때문에 이미지 인식 기반의 인공지능 알고리즘을 이용한 경우에도 한계가 있음이 드러났다.

의료영상 인공지능 분야에서 CNN의 한계로 인하여 의료영상으로부터 의학적 정보를 추출하여 환자의 예후예측을 진단하는 학문으로 라디옴릭스(radiomics)가 대두되었듯이, 본 연구는 스마트팜 분야의 이미지 인식에서도 CNN 딥러닝만으로는 해결할 수 없는 부분은 데이터의 특징을 수치화 시켜서 딥러닝을 수행하여 성능 향상을 극대화할 수 있음을 확인하였다.

향후 본 논문에서 제안한 데이터셋 구축 방식과 딥러닝 모델을 기반으로 다양한 과일 및 농수산물 이미지 분류 시스템으로 응용한다면 국산품과 수입품의 구분뿐만 아니라 국내의 지역별 농수산물에 대한 분류 체계를 인공지능 기반으로 자동화할 수 있을 것이라 기대한다.

### 참고문헌

[1] S. G. Han, S. B. Hong, Y. J. kim, B. C. kim, K. M. Jung, S. M. Ahn, *ArtificialIntelligence Lab for Ai Thinking*, Sung Andang, 2020.  
 [2] J. H. Park, G. B. Hwang, H. M. Park, Y. K. Choi, "Application of CNN for Fish Species Classification," *Journal of the Korea Institute of Information and*

*Communicatio Engineering*, Vol. 23, No. 1, pp. 39-46, Jan. 2019.

[3] Y. Le Cun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based learning applied to document recognition," *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278-2324, Nov. 1998.

[4] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, Vol. 2, No. 1, pp. 1-127, Feb. 2009.

[5] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, Vol. 18, No. 7, pp. 1527-1554, Jul. 2006.

[6] S. I. Hassan, L. Dang, S. H. Im, K. B. Min, J. Y. Nam, and H. J. Moon, "Damage Detection and Classification System for Sewer Inspection using Convolutional Neural Networks based on Deep Learning," *Journal of the Korea Institute of Information and Communication Engineering*, Vol. 22, No. 3, pp. 451-457, Mar. 2018.

[7] J. H. Kim, D. S. Choi, H. S. Lee, and J. W. Lee, "Target Classification of Active Sonar Returns based on Convolutional Neural Network," *Journal of the Korea Institute of Information and Communication Engineering*, Vol. 21, No. 10, pp. 1909-1916, Oct. 2017.

[8] Penerapan Edge Detection [Internet]. Available: [http://jagocoding.com/tutorial/500/Penerapan\\_Edge\\_Detection\\_Sobel\\_Prewitt\\_Laplace\\_Robert\\_dan\\_Canny\\_Pada\\_Matlab/](http://jagocoding.com/tutorial/500/Penerapan_Edge_Detection_Sobel_Prewitt_Laplace_Robert_dan_Canny_Pada_Matlab/).

[9] Convolutional Neural Networks [Internet]. Available: <https://excelsior-cjh.tistory.com/180>.

[10] J. J. Kim, 3-Minute Deep Learning Tensorflow Flavor, *Hanbit Media*, 2017.

[11] I. K. Choi, H. E. Ahn, and J. S. Yoo, "Facial Expression Classification Using Deep Convolutional Neural Network," *Journal of Electrical Engineering & Technology*, Vol. 13, No. 1, pp. 485-492, Jan. 2018.

[12] H. S. Kim, "Classification and Combination of Fashion Items Using CNN-Based Deep Learning," *Journal of Digital Contents Society*, Vol. 22, No. 3, pp. 475-482, Mar. 2021.

[13] J. G. Kim, W. H. Jo, M. H. Na, M. H. Jeon, "Development of Automatic sorting system of Vegetables(Cucumbers) by using Image processing technique and Deep learning," *Journal of the Korea Data Analysis Society*, Vol. 21, No. 1, pp. 63-73, Feb. 2019.

[14] T. H. Cho, *Deep learning for everyone*, Gilbut, 2020.

[15] S. C. Lim, S. H. Kim, Y. H. Kim, D. Y. Kim, "Training Network Design Based on Convolution Neural Network for Object Classification in few class problem," *Journal of*

*the Korea Institute of Information and Communication Engineering*, Vol. 21, No. 1, pp. 144-150 Jan. 2017.

[16] N. Y. Lee, J. Y. Kim, J. H. Shim, "Empirical Study on Analyzing Training Data for CNN-based Product Classification Deep Learning Mode," *The Journal of Society for e-Business Studie*, Vol. 26, No. 1, pp.107-126, February. 2021.

[17] H. S. Kim, "Mask R-CNN Deep Learning for fashion element detection," *Journal of Digital Contents Society*, Vol. 22, No. 4, pp. 689-696, Apr. 2021.



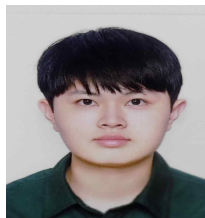
**김명진(Myung-Jin Kim)**

현 재: 한국폴리텍V대학 AI융합과  
(하이테크과정)  
※관심분야 : 빅데이터, 딥러닝 등



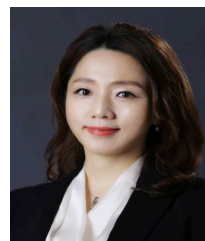
**한영석(Young-Seok Han)**

현 재: 한국폴리텍V대학 AI융합과  
(하이테크과정)  
※관심분야 : 빅데이터, CNN, RNN 등



**김승우(Seung-Woo Kim)**

현 재: 한국폴리텍V대학 AI융합과  
(하이테크과정)  
※관심분야 : 인공지능, 블록체인 등



**김희숙(Hye-Suk Kim)**

1999년 : 전남대학교 대학원 전산통계  
학과 (이학석사)  
2009년 : 전남대학교 대학원 전산학과  
(이학박사)

2018년~현 재: GIST(Gwangju Institute of Science and Technology) Lecture Professor  
※관심분야 : 인공지능(AI), 딥러닝(Deep Learning), 머신러닝(Machine Learning), 데이터과학(Data Science), 컴퓨터 비전(Computer Vision), UX·UI 등