

## 컴퓨터 비전과 딥러닝을 통한 견종식별 연구

정택현\*, 김기천<sup>o</sup>A Study on Dog Breed Identification through  
Computer Vision and Deep LearningTack-hyun Jung\*, Keecheon Kim<sup>o</sup>

요약

본 논문은 컴퓨터 비전(computer vision)과 딥러닝(deep-learning)을 접목한 지능형 ‘반려동물 돌봄’ 서비스를 구현하기 위한 연구의 초석으로 이미지 내 반려견을 식별하여 품종(breed)을 식별하는 모델의 제안과 관련된 실험을 수행한 결과를 기술한다. 견종의 식별에는 Feature representation learning에서 뛰어난 성능을 보이는 CNN(Convolution Neural Network) 모델을 활용하는데, CNN은 이미지의 불필요한 영역과 Noise를 제거하는 것이 성능에 큰 영향을 주는 것으로 알려져 있다. 따라서 제시하는 모델은 U-Net을 접목하여 Instance Segmentation을 통해 개체 외에 불필요한 영역을 모두 제거하고 CNN을 통해 Identification을 수행한다. 제안하는 모델은 CNN을 단독으로 사용하는 모델과 대비하여 17.86% 높은 성능을 보였으며, 여덟 마리의 견종을 식별함에서 91.62%라는 결과를 보였다. 하지만, 실험을 통해 얼굴 혹은 신체 일부만 나타난 견종의 이미지에서는 두 견종 간 색상의 배열이나 형태가 유사한 경우, CNN 모델이 이를 명확하게 구별하지 못한다는 한계점이 식별된 것도 또 다른 의의가 있으며, 이를 개선하기 위한 후속 연구의 방향성을 제시한다고 생각된다.

**Key Words** : Computer Vision, Deep Learning, Companion Animal

## ABSTRACT

In this paper, we propose a new approach to dog breed identification model through the computer vision and deep-learning and perform the relevant experiments and describe the results. These experiments will be a cornerstone for research to implement ‘smart pet care services’. The identification of dog breed is based on CNN(Convolution Natural Network) model, which shows excellent performance in Feature representation learning. CNN is known to improve performance if unnecessary areas and noise are removed. Therefore, the proposed model combines U-Net to remove all unnecessary areas through Instance Segmentation and performs identification through CNN. The proposed model showed 17.86% higher performance than the CNN alone model, and 91.62% accuracy in identifying eight dog breeds. However, experiments have shown that the CNN model does not clearly identification between two breeds of similar color arrangements or shapes in the image of a dog that shows only a face or body part. These results has another different significance to the Identification accuracy and the direction of our future research to improve it.

\* 본 연구는 2019년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보-컴퓨팅기술개발사업의 지원(No. NRF-2017M3C4A7083678)과 한국연구재단의 지원(No. 2017R1A2B4008860)을 받아 수행되었습니다.

• First Author : Konkuk University Department of IT Convergence Information Security, tackhyun12@konkuk.ac.kr, 학생회원

<sup>o</sup> Corresponding Author : Konkuk University Department of Computer Engineering, kckim@konkuk.ac.kr, 종신회원

논문번호 : 201909-194-D-RN, Received September 10, 2019; Revised October 24, 2019; Accepted October 24, 2019

## I. Introduction

최근 1인 가구의 증가와 저출산 고령화 등에 따라 반려동물을 기르는 사람들과 그 시장규모도 함께 커져가고 있다. 이처럼 ‘반려동물’ 문화가 우리 생활에 밀접한 관계를 띄기 시작함에 따라, 반려동물을 더욱 잘 돌보기 위한 다양한 네트워크 서비스와 앱(application) 등이 상용화되고 있다. 이러한 기술 중 대표적인 것으로는 ‘반려동물 관찰 카메라’가 있는데, 이는 가정 내부의 네트워크 장비(공유기 등)를 통해 IP 카메라의 정보를 실시간으로 앱에서 수신하여, 반려동물의 행동을 관찰하는 시스템이다.

하지만, 시중에서 판매하는 ‘반려동물 관찰 카메라’의 대다수 제품은 단순히 영상을 스마트폰으로 전송하거나 낮은 수준의 움직임 감지만이 가능하다. 실제로 본 연구를 진행하기에 앞서, 이러한 시스템의 표본을 일부 분석한 결과, 대부분이 영상의 픽셀 혹은 픽셀 그룹을 추적하여 단기간 내 변화가 있는 곳을 식별하는 정도의 기술만이 적용되어 있었으며, 탐지 성능도 좋지 않았다.

실제로 다른 가족 형태보다 30대 1인 가구가 반려동물을 키우는 비율이 가장 높은 것으로 알려져 있는데<sup>[1]</sup>, 이들은 대부분 직장을 다니기 때문에, 홀로 반려동물을 돌봐야 하는 상황에서 이러한 시스템과 서비스는 현실적으로 충족치 못한 실정이다.

따라서 본 논문은 컴퓨터 비전(computer vision)과 딥러닝(deep-learning)을 접목하여 지능적이고(smart), 안정적인 ‘스마트 반려동물 돌봄’ 서비스를 구현하기 위한 연구의 초석으로 영상 내 반려동물을 식별하여 품종(breed)을 식별하는 연구를 수행하고자 한다. 아직 초기 단계의 연구인만큼, 이미지 데이터의 확보가 수월한 개(dog)를 대상으로 연구 범위를 한정하였고, 이를 통해 컴퓨터 비전과 딥러닝 기술이 영상 정보에서 반려동물의 품종을 식별하는데 어떠한 가능성을 가지는지와 연구결과에 대한 의미를 살피고자 한다.

## II. Related Works

컴퓨터 비전과 딥러닝 기반의 견종식별 연구는 개체를 검출(detection)하고 품종을 인식(recognition)하는 일련의 절차로 구현된다. 검출은 픽셀로 이루어진 이미지에서 대상 object를 수학적으로 혹은 실용적으로 검출된 특징을 추출(Feature Extraction)하여 특징들의 분포로부터 Boundary Decision을 찾는 개념이다.



그림 1. Landmark 및 DAR 적용결과  
Fig. 1. Result of Landmark and DAR apply

[Fig 1]은 이러한 검출 알고리즘 중 가장 대표적인 Landmark Estimation<sup>[2]</sup>와 (1)을 통해 개의 얼굴 영역을 검출한 그림이다. Face Landmark Estimation은 2014년 Vahid Kazemi와 Josephine Sullivan이 발명한 얼굴 검출에 대한 접근방법으로, Landmarks로 정의되는 여러 개의 포인트를 이마 상단, 눈의 가장자리, 코 등 얼굴의 특정 부위에 표시하는 개념이다.<sup>[2]</sup> 본 연구는 이러한 알고리즘을 활용하여 개의 얼굴에 적용하였고, (1)과 같은 수식을 적용하였다.

$$DAR = \frac{\|p_1 - p_4\| + \|p_2 - p_5\|}{2\|p_0 - p_3\|} \quad (1)$$

수식 (1)은 적용된 Landmark의 점( $p_i$ )들을 활용하여, 얼굴의 면적을 계산하는 식을 정의한다. [Fig 1]에서 점  $p_0$ 와  $p_3$ 는 얼굴의 면적 중 세로축을 의미하고, 나머지 점은 가로축을 의미한다. 따라서, 가로축과 세로축의 면적을 통해 계산된 크기의 절대 치를 구하는 것이 주된 개념이며, EAR(Eye Aspect Ratio)<sup>[3]</sup>의 개념을 참조한다. 이는 본 논문에서는 DAR(Dog Aspect Ratio)로 정의하며, 이처럼 계산된 값은 각각의 견종의 생김새에 따라 값이 유의하게 변화하므로, 이를 검출하고 성능 고찰에 관한 실험을 수행하였고 그 결과는 본 논문의 4장에 기술한다. 개체를 식별(identification)하는 것은 검출(detection)보다 더욱 어려운 개념이다. 본 연구는 이미지에서 식별 중 Feature representation learning에서 뛰어난 성능을 보이는 CNN(Convolution Neural Network)<sup>[4]</sup>을 통해 식별 모델을 구현하였다. CNN은 학습시키는 과정에서 이미지의 불필요한 영역과 Noise를 제거하는 것이 성능에 큰 영향을 준다는 사실이 알려져 있다. 따라서, U-Net<sup>[5]</sup> 모델을 CNN과 모델과 결합하였다. 이는, U-Net과 CNN을 결합하는 개념으로 U-Net을 통해 Instance Segmentation을 수행하여 개체 외에 불필요한 영역을 모두 제거하고 CNN을 통해 Identification을 수행하는 개념이다. 해당 내용은 논 논문의 3장에

자세히 기술하였으며, 이를 통해 반려동물 품종 식별의 가능성과 더불어 제시된 가설에 대한 연구결과를 고찰한다.

### III. Proposed Identification Model

#### 3.1 Basic concept of the proposed model

본 연구에서 제안하는 견종식별 모델은 [Fig 2]와 같은 일련의 절차를 갖는다. 먼저, 웹 크롤러(web crawler)를 통해 대량의 개(dog) 이미지를 수집하며, 이를 대상으로 전처리(Pre-Process)를 수행하여 U-Net의 Train Dataset을 생성한다. 이러한 데이터를 기반으로 U-Net 모델의 훈련이 완료되면, 이후부터는 수집된 이미지를 U-Net을 통해 Segmentation을 수행하는 것으로 CNN의 Train Dataset을 추가 생성한다. 이러한 과정을 통해 두 신경망 모델에 학습이 완료된 경우 견종식별을 수행할 수 있다.

견종식별은 Input Image를 대상으로 U-Net을 통해 Instance Segmentation을 수행하여 개체(dog) 외에 불필요한 영역을 모두 제거하고, CNN을 통해 Identification을 수행하는 것으로 정의한다.

#### 3.2 Pre-process and Dataset

컴퓨터 비전(computer vision)의 분야에서는 개체(object)에 대한 인식이나 처리를 수행하기 위해 선행적으로 물체를 배경으로부터 분리하거나 추출한다. 이러한 과정이 수행되는 원인은 일차적으로 이미지에서 불필요한 영역과 Noise를 제거하기 위함이다.<sup>[6]</sup>

따라서, 본 연구는 웹 크롤러(web crawler)를 통해 수집한 이미지를 대상으로 Otsu-Thresholding<sup>[7]</sup>을 적용하는 것으로 이미지가 담고 있는 물체인 개(dog)를 분리하는 실험을 수행하였다.

[Fig 3]은 수집된 이미지를 대상으로 Thresholding을 적용하는 실험의 결과이다. 그림을 살펴보면, 좌측 원본 이미지(Source)를 기준으로 Thresholding을 적용하였을 때 우측 이미지가 나타나며 중간에는 Histogram이 표현된다. 이러한 결과에서는 이미지의 광원과 색상배열뿐 아니라 훨씬 다양한 변수가 작용한다.

따라서, Thresholding의 Best-condition을 고찰하기 위해 50장의 표본 이미지를 대상으로 [Fig 3]의 방식으로 실험을 수행하였다. 그 결과, Gaussian-filter<sup>[8]</sup>를 적용하는 것이, 그렇지 않은 것보다 Noise 제거에서 평균적으로 더욱 효과적인 결과를 보였으며, Open-CV에서 제공하는 Binary-Thresholding<sup>[9]</sup> 알고리즘 보다, Otsu-Thresholding<sup>[7]</sup>이 경계선 검출과 비어있는 구멍(hole)의 생성에 관한 측면에서는 더 좋은 성능을 보였다.

하지만, Thresholding의 과정에서 나타나는 개체 내 비어있는 구멍(hole)은 학습 데이터로 사용하게 될 때 문제가 될 수 있다. 따라서, [Algorithm 1]을 통해 비어있는 구멍들을 채우는 과정을 추가로 구현한다. 이처럼 Open-CV에서 제공하는 floodFill 함수를 활용하면, [Fig 4]과 같이 높은 정확도로 비어있는 구멍들을 찾아 채울 수 있다.

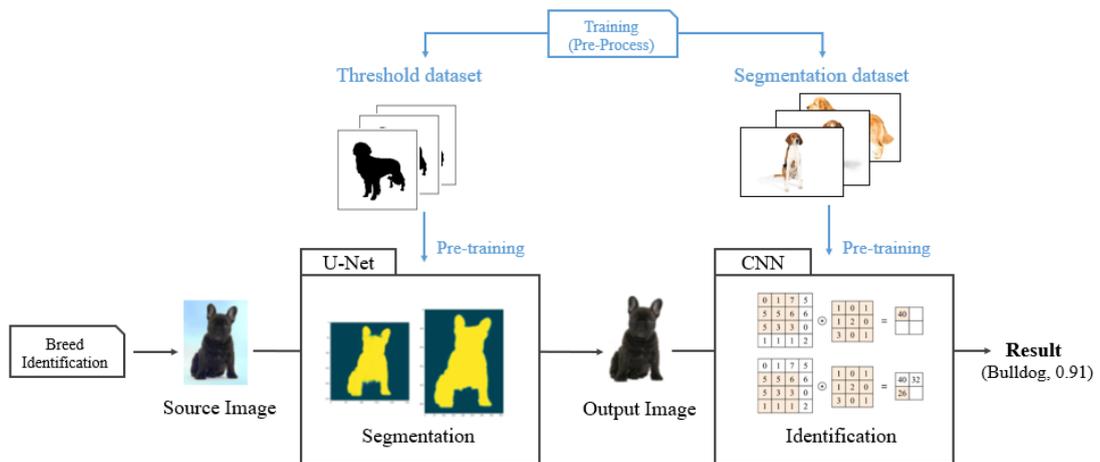


그림 2. 제안하는 견종식별 모델의 기본 개념  
Fig. 2. Basic concept of proposed identification model

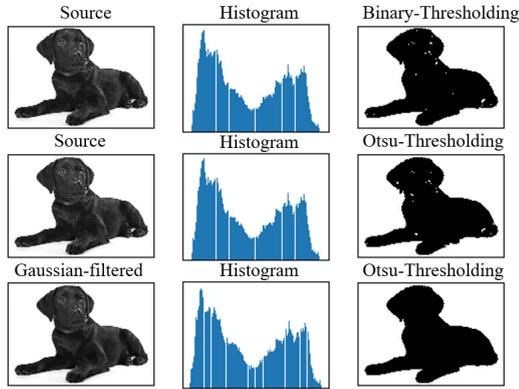


그림 3. 다양한 Thresholding 실험의 결과  
Fig. 3. The results of various Thresholding experimental

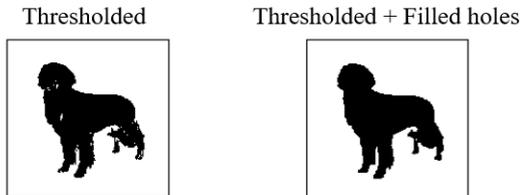


그림 4. Thresholding 이후 비어있는 구멍을 채운 결과  
Fig. 4. Result of filled holes after applied thresholding

**Algorithm 1 : A Method for Filling Holes**

1. **Input** : img(thresholded)  
cv2.floodFill(img, mask=np.zeros((h+2, w+2), np.uint8),  
seedPoint=(0,0), newVal=255)
2. filled\_img = cv2.bitwise\_not(img)
4. **Output** : filled\_img

**3.3 Implement Identification Model**

Pre-processing의 과정을 거친 표본 이미지 데이터는 U-Net<sup>5)</sup>의 학습 표본으로 활용한다. 이처럼, U-Net을 사용하는 이유는 이미지로부터 개(dog)를 추출하는 과정을 Thresholding보다 더욱 정밀하고 좋은 결과를 얻기 위함이다. 또한, 기존의 U-Net이 이미 다양한 연구를 통해 Segmentation<sup>10)</sup>의 분야에서 높은 성능을 완성하였음을 근거로 한다.

실제로 단순히 Thresholding만을 적용하는 것은 여러 실험을 통해 조건들을 정밀하게 설정하더라도, 유사한 색상배열(열은 회색-진한 회색 등)에서 큰 한계점을 보인다. 따라서, 본 연구는 Pre-processing의 과정을 통해 얻은 Masking 데이터를 가공하여 U-Net에 학습시키는 것으로 Segmentation model을 구축하였다.

Table 1. Used convolutional architecture of U-Net

Layer	Output Shape	Connected to
input-layer	(None, 256, 256, 3)	
conv2d-01 (Conv2D)	(None, 256, 256, 32)	input
conv2d-02 (Conv2D)	(None, 256, 256, 64)	conv2d-01
max_pooling2d-01 (MaxPooling2D)	(None, 128, 128, 64)	conv2d-02
conv2d-03 (Conv2D)	(None, 128, 128, 64)	max_pooling2d-01
conv2d-04 (Conv2D)	(None, 128, 128, 128)	conv2d-03
max_pooling2d-02 (MaxPooling2D)	(None, 64, 64, 128)	conv2d-04
conv2d-05 (Conv2D)	(None, 64, 64, 128)	max_pooling2d-02
conv2d-06 (Conv2D)	(None, 64, 64, 256)	conv2d-05
max_pooling2d-03 (MaxPooling2D)	(None, 32, 32, 256)	conv2d-06
depthwise_conv2d-01 (DepthwiseConv2D)	(None, 32, 32, 256)	max_pooling2d-03
conv2d-07 (Conv2D)	(None, 32, 32, 256)	depthwise_conv2d-01
depthwise_conv2d-02 (DepthwiseConv2D)	(None, 32, 32, 256)	conv2d-07
conv2d-08 (Conv2D)	(None, 32, 32, 256)	depthwise_conv2d-02
add_layer-01 (Add)	(None, 32, 32, 256)	max_pooling2d-03 conv2d-08
up_sampling2d-01 (UpSampling2D)	(None, 64, 64, 256)	add_layer-01
concatenate-01 (Concatenate)	(None, 64, 64, 512)	up_sampling2d-01 conv2d-06
conv2d-09 (Conv2D)	(None, 64, 64, 128)	concatenate-01
conv2d-10 (Conv2D)	(None, 64, 64, 128)	conv2d-09
add_layer-02 (Add)	(None, 64, 64, 128)	max_pooling2d-02 conv2d-10
up_sampling2d-02 (UpSampling2D)	(None, 128, 128, 128)	add_layer-02
concatenate-02 (Concatenate)	(None, 128, 128, 256)	up_sampling2d-02 conv2d-04
conv2d-11 (Conv2D)	(None, 128, 128, 64)	concatenate-02
conv2d-12 (Conv2D)	(None, 128, 128, 64)	conv2d-11
add_layer_03 (Add)	(None, 128, 128, 64)	max_pooling2d-01 conv2d-12
up_sampling2d-03 (UpSampling2D)	(None, 256, 256, 64)	add_layer-03
concatenate-03 (Concatenate)	(None, 256, 256, 128)	up_sampling2d-03 conv2d-02
conv2d-13 (Conv2D)	(None, 256, 256, 32)	concatenate-03
conv2d-14 (Conv2D)	(None, 256, 256, 32)	conv2d-13
conv2d-15 (Conv2D)	(None, 256, 256, 2)	conv2d-14
reshape-01 (Reshape)	(None, 65536, 2)	conv2d-15
activation-01 (Activation)	(None, 65536, 2)	reshape-01
reshape-02 (Reshape)	(None, 256, 256, 2)	activation-01

[Table 1]은 구현된 U-Net 신경망 구조를 의미한다. Up-sampling의 과정에서 output을 이전 convolution feature map과 channel의 방향으로 resizing을 수행한 뒤 concatenation 하는 일반적인 형태로 구현하였다.<sup>15)</sup>

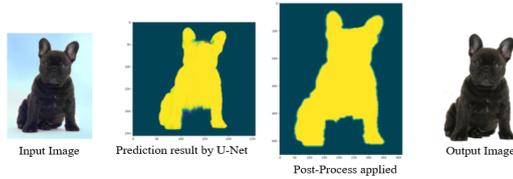


그림 5. U-Net을 통한 Segmentation 처리 과정 및 결과  
Fig. 5. Process and result of segmentation through U-Net

Optimizer는 gradient와 learning rate를 모두 고려하는 Adam optimizer<sup>[11]</sup>를 활용하였고, 손실함수(Loss)는 categorical cross entropy<sup>[12]</sup>를 활용하였다.

[Fig 5]는 이미지를 대상으로 U-Net을 통한 Segmentation의 처리 과정과 그 결과를 의미한다. 그림을 살펴보면, 먼저 학습된(Trained) U-Net을 통해 Prediction을 수행하여 Segmentation을 수행한다. 이러한 결과인 output은 Segmentation map으로 나타나며, 대상(object)은 1로, 나머지는 0으로 처리한다.

이후, Post-Process(후처리) 과정을 통해 결과물을 보정하는 과정을 거치는데, 이는 Open-CV<sup>5)</sup>에서 제공하는 Morphology<sup>13)</sup> 연산 중 침식(Erode)을 활용하여 작은 Noise의 제거와 경계에 대한 명확성을 확보하고, Gaussian\_Blur<sup>8)</sup>를 통해 외곽선(가장자리)을 부드럽게 하여 개체 분리의 성능을 개선하는 원리를 갖는다. 이러한 Post-Process에 대한 구현명세는 [Algorithm 2]로 정의한다.

Segmentation을 통해 개(dog)의 객체를 따로 분리한 이후에는, CNN(Convolution Neural Network)<sup>4)</sup>을 통해 Feature representation learning을 수행한다. 이미지에서 특징을 찾는 Feature Engineering을 스스로 수행하는 CNN은 이미지에서 불필요한 영역과 Noise를 제거하는 것이 학습 성능에 큰 영향을 준다고 알려져 있다. 이러한 사실에 따라, 앞서 적용한 Segmentation은 학습 대상인 개(dog)를 제외하고 모

Table 2. Used convolutional architecture of CNN

Layer	Output Shape	Connected to
conv2d-01 (Conv2D)	(None, 64, 64, 32)	input_layer
max_pooling2d-01 (MaxPooling2D)	(None, 32, 32, 32)	conv2d-01
dropout_01 (Dropout)	(None, 32, 32, 32)	max_pooling2d-01
conv2d-02 (Conv2D)	(None, 32, 32, 64)	dropout_01
max_pooling2d-02 (MaxPooling2D)	(None, 16, 16, 64)	conv2d-02
dropout_02 (Dropout)	(None, 16, 16, 64)	max_pooling2d-02
conv2d-03 (Conv2D)	(None, 16, 16, 128)	dropout_02
conv2d-04 (Conv2D)	(None, 16, 16, 128)	conv2d-03
max_pooling2d-03 (MaxPooling2D)	(None, 8, 8, 128)	conv2d-04
dropout_03 (Dropout)	(None, 8, 8, 128)	max_pooling2d-03
flatten_01 (Flatten)	(None, 8192)	dropout_03
dense_01 (Dense)	(None, 256)	flatten_01
dropout_04 (Dropout)	(None, 256)	dense_01
dense_02	(None, 13)	dropout_04

두 제거하는 원리이기 때문에 CNN의 성능 개선점과 큰 상관관계를 갖는다.

[Table 2]는 구현된 CNN 모델의 구조를 의미한다. 해당 모델의 학습을 위한 Pre-Process로는 데이터를 float 자료형으로 변환하는 가장 보편적인 방법의 정규화(Normalization)<sup>14)</sup>를 통해 값의 Scale을 조정하였다.

[Table 3]은 견종별 학습 표본을 의미한다. 표본은 웹 크롤러(web crawler)를 통해 대량의 개(dog) 이미지를 수집하였으며, 그 과정에서 해시 기반으로 중복 검사를 수행하여 중복되는 이미지를 최대한 배제하였다. 표본은 앞서 정의한 U-Net 모델을 통해 개(dog)의 영역 외의 모든 영역과 noise가 완벽히 제거하였으며, 총 8개의 견종을 카테고리화 선정하였다. 이는, 난이도를 유의하게 조절하여 성능을 고찰하기 위한 목적으로, American Bulldog 그리고 French Bulldog과 같은 유사한 여덟 마리의 견종을 표본으로 선정하였다.

**Algorithm 2 : A Method for Segmentation Performance Improvement**

1. **Input** : img(applied Segmentation) element = cv2.getStructuringElement(cv2.MORPH\_RECT, (2\*EROSION+1, 2\*EROSION+1), (EROSION, EROSION))
2. img = cv2.erode(img, element)  
improved\_img = cv2.GaussianBlur(img, ksize=(10, 10), sigmaX=6, sigmaY=6)
4. **Output** : improved\_img

Table 3. The training samples by dog breed

Dog breed	Count	Percentage(%)
 American Foxhound	4,852	14.30
 American Bulldog	3,917	11.54
 Beagle	4,792	14.12
 Basset Hound	3,804	11.21
 Dalmatian	4,798	14.14
 French Bulldog	3,872	11.41
 Golden Retriever	3,109	9.16
 Korean Jindo Dog	4,782	14.09
<b>Total</b>	<b>33,926</b>	<b>100</b>

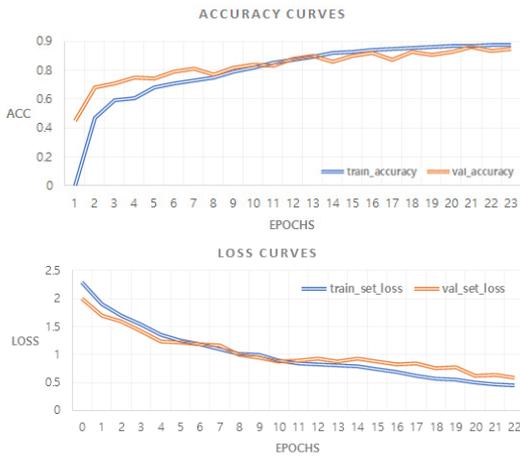


그림 6. CNN 모델의 training 간의 정확도와 손실 곡선  
Fig. 6. The Accuracy and Loss curves to training of CNN model

#### IV. Experiments

##### 4.1 Accuracy of Proposed Model

본 연구는 제안하는 모델의 성능에 관한 측정과 관련된 연구들과의 성능을 비교/분석하는 두 가지 실험을 수행하였다. 먼저, 제안하는 모델의 [Fig 2]의 구조를 가지며, 성능 측정은 [Table 3]의 학습 표본을 바탕으로 모델을 Train 하여 총 474장의 개(dog) 이미지에

Table 4. The experimental samples and results by proposed model

Dog breed	Experimental Samples and Result		
	Count	Percentage(%)	Accuracy(%)
American Foxhound	58	12.23	89.65
American Bulldog	72	15.18	93.05
Beagle	62	13.08	85.48
Basset Hound	58	12.23	87.93
Dalmatian	60	12.65	93.33
French Bulldog	51	10.75	92.15
Golden Retriever	55	11.60	100
Korean Jindo Dog	58	12.23	91.37
<b>Total</b>	<b>474</b>	<b>100</b>	<b>91.62</b> (Average)

대한 Identification Accuracy를 측정하였다. 이때, 측정에 활용되는 이미지는 학습에 사용하지 않은 표본을 사용하였다.

[Table 4]는 실험에 사용된 표본에 관한 정보와 제안하는 모델에 의한 식별실험의 결과이다. 여덟 마리 견종의 Accuracy 평균은 91.62%이며, Golden Retriever(🐶)의 경우 55장의 이미지 모두를 정확히 추론하였다. 하지만, Beagle(🐶)과 Basset Hound(🐶)의 경우, 각각 85.48%과 87.93%로 평균보다 크게 낮은 성능을 보였다. 이는, 실험에 사용된 표본에서 개(dog)의 전신이 모두 나타난 이미지에서는 높은 정확도를 보였지만, 얼굴 혹은 신체 일부만 나타난 이미지에서는 두 견종 간 색상의 배열이나 형태가 유사하여, CNN 모델이 이를 명확하게 구별하지 못하였다.

##### 4.2 Performance Comparison Analysis

[Table 5]는 견종을 식별하는 관련 연구들과의 성능을 비교/분석하는 실험의 정보와 그 결과를 의미한다. 실험은 2장에 기술한 Dog Landmark<sup>[2]</sup>의 성능과 CNN을 단독으로 사용 시에 성능, 그리고 본 논문에서 제안하는 모델의 성능을 비교하였다.

Dog Landmark 실험의 경우 피사체(dog)가 정면을

Table 5. The breed identify performance of various algorithms

Category	Algorithm		
	Land Mark <sup>[2]</sup>	CNN	Proposed Model
Dataset	5,892	33,926	33,926
Accuracy (50% Trained)	33.6591%	68.1298%	87.8592%
Accuracy (100% Trained)	37.0837%	73.7525%	91.6237%

바라보고 있지 않으면 인식률이 매우 떨어지므로, 정면을 바라보고 있는 이미지만 표본화하여 학습을 수행하였다. CNN 단독 사용의 실험은 U-Net을 사용하는 본 모델과의 성능 비교를 위한 목적으로, Segmentation의 과정 없이 CNN의 독자적인 성능을 측정하는 개념이며, Convolutional Architecture는 제안하는 모델과 동일한 구조이다.

[Table 6]은 Dog Landmark를 통한 견종식별 실험에 사용된 표본에 관한 정보와 실험의 결과이다. 여덟 마리 견종의 Accuracy 평균은 37.08%이었지만, 실제로 여러 번의 실험에서 일관성 있는 결과가 나타나지 않았다. 이는, Landmark를 통해 나타난 결과에서 mark의 위치 정보(x, y)가 항상 일정하지 않았고, 특정 이미지에서는 여러 번의 인식에서 조금씩 유의한 변화가 있었기 때문이다. 이러한 실험의 결과는 해당 방법론이 개체검출(detection)에서는 의미가 있지만, 구분(classification)의 개념에서는 학습 데이터의 양과 관계없이 부적합함을 시사한다.

[Table 7]은 CNN을 통한 견종식별 실험에 사용된 표본에 관한 정보와 실험의 결과이다. CNN을 통한 여덟 마리 견종의 Accuracy 평균은 73.75%로 본 연구에서 제안하는 모델과는 17.86%의 차이가 있었다. 실험에서 American Bulldog(🐾), French Bulldog(🐾), Korean Jindo dog(🐾)의 경우, 각각 62.50%, 60.78% 그리고 70.68%로 평균보다 크게 낮은 성능을 보였는데, 특히 흰색 품종의 개들끼리 서로 오탐을 하는 결과를 보였다. 이는, 실험에 사용된 표본의 배경(background)의 색상이 흰색인 표본에서 유독 오탐률이 높다는 점으로 미루어 볼 때, 이미지에서 불필요한 영역과 Noise를 제거하는 것이 CNN의 성능과 큰 상관관계가 있다는 결론이 시사된다.

Table 6. The experimental samples and results by Landmark[2]

Dog breed	Experimental Samples and Result		
	Count	Percentage(%)	Accuracy(%)
American Foxhound	58	12.23	39.65
American Bulldog	72	15.18	37.5
Beagle	62	13.08	41.93
Basset Hound	58	12.23	32.75
Dalmatian	60	12.65	36.66
French Bulldog	51	10.75	37.25
Golden Retriever	55	11.60	38.18
Korean Jindo Dog	58	12.23	32.75
<b>Total</b>	<b>474</b>	<b>100</b>	<b>37.08375</b> (Average)

Table 7. The experimental samples and results by CNN

Dog breed	Experimental Samples and Result		
	Count	Percentage(%)	Accuracy(%)
American Foxhound	58	12.23	68.96
American Bulldog	72	15.18	62.50
Beagle	62	13.08	77.41
Basset Hound	58	12.23	77.58
Dalmatian	60	12.65	86.66
French Bulldog	51	10.75	60.78
Golden Retriever	55	11.60	85.45
Korean Jindo dog	58	12.23	70.68
<b>Total</b>	<b>474</b>	<b>100</b>	<b>73.7525</b> (Average)

### 4.3 Future works and contribution

본 실험의 결과는 미래의 스마트 반려동물 돌봄 서비스를 구현하기 위한 연구의 초석이 된다. 따라서, [Fig 7]은 이러한 서비스의 구조를 정의한다.

본 연구는 [Fig 7]의 전체 스마트 반려동물 돌봄 서비스의 구조 중 “Proposed Architecture”의 영역을 가진다. 이는, IP 카메라의 영상 정보를 기반으로 제안하는 아키텍처를 통해 견종식별을 수행하고 그 결과를 기반으로 맞춤형 반려동물 돌봄 서비스로 연계시키는 역할이 된다.

맞춤형 반려동물 돌봄 서비스는 견종별 맞춤 사료(Breed Health Nutrition)를 자동화하여 주문하거나, 견종 별 유의한 예방접종 주기의 분석 그리고 반려동물의 움직임이나 패턴을 파악하여 동일 견종과의 대조로 행동 및 정서를 분석하는 등의 다양한 서비스가 가능하다. 이러한 서비스는 다양한 수의학적 정보를 DB화하여 구축하고, 제안하는 아키텍처를 통해 식별된 견종을 통해 검색되는 구조로 연계될 수 있다.

이러한 서비스를 구현하기 위해서는 실험을 통해 식별된 CNN 모델의 한계점을 개선하기 위한 연구를

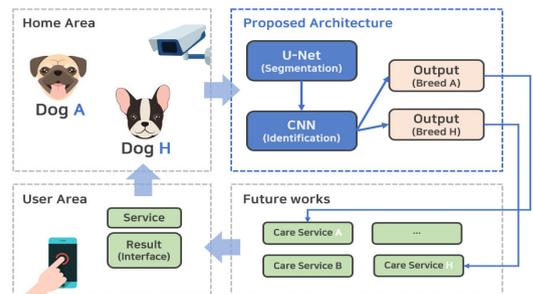


그림 7. 미래의 스마트 반려동물 돌봄 서비스  
Fig. 7. The Smart pet care service in future

포함하여 다양한 후속 연구가 필요하다. 본 논문은 이러한 개선의 방향성을 제시한다고 생각된다.

## V. Conclusion

본 연구는 컴퓨터 비전(computer vision)과 딥러닝(deep-learning)을 접목하는 지능형 ‘반려동물 돌봄’ 서비스를 구현하기 위한 연구의 초석으로 이미지 내 반려견을 식별하여 품종(breed)을 식별하는 모델의 제안과 관련된 실험을 수행하였다.

제안하는 모델은 CNN을 단독으로 사용하는 모델과 대비하여 17.86% 높은 성능을 보였으며, 여덟 마리의 견종을 식별함에서 91.62%라는 결과를 보였다. 하지만 개(dog)의 전신이 모두 나타난 이미지에서는 높은 정확도가 보장되지만, 얼굴 혹은 신체 일부만 나타난 이미지에서는 두 견종 간 색상의 배열이나 형태가 유사한 경우, CNN 모델이 이를 명확하게 구별하지 못한다는 한계점이 식별되었다. 이러한 결과는 GANs<sup>16)</sup>를 추가 접목하는 등의 개선을 위한 후속 연구의 필요성을 제기하며, 본 연구는 기존 CNN에 U-Net을 접목하는 새로운 접근방법으로 이미지에서 불필요한 영역과 Noise를 제거하는 것이 CNN의 성능과 큰 상관관계가 있다는 결론을 실험을 통해 도출한 것에 의의가 있다.

## References

[1] B. Y. Oh, "A study on service design for collaborative caring for pets owned by single working households," *KSDC*, vol. 24, no. 4, 2018.

[2] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proc. 2014 IEEE Conf. CVPR*, pp. 1867-1874, Jun. 2014.

[3] T. Soukupova and J. Cech, "Real-time eye blink detection using facial landmarks," *21st Computer Vision Winter Workshop*, Feb. 2016.

[4] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98-113, Jan. 1997.

[5] O. Ronneberger, P. Fischer, and T. Brox,

"U-Net: Convolutional networks for biomedical image segmentation," *MICCAI 2015*, pp. 234-241, 2015.

[6] A. K. Boyat and B. K. Joshi, "A review paper: Noise models in digital image processing," *SIPIJ*, vol. 6, no. 2, Apr. 2015.

[7] S. Bangare, A. Dubal, P. S. Bangare, and S. Patil, "Reviewing otsu's method for image thresholding," *Int. J. Appl. Eng. Res.*, vol. 10, no. 9, pp. 21777-21783, 2015.

[8] E. Elboher and M. Werman, "Efficient and accurate gaussian image filtering using running sums," *12th Int. Conf. Intell. Syst. Design and Appl. (ISDA)*, 2012.

[9] I. Culjak, et al., "A brief introduction to OpenCV," in *Proc. 35th Int. Convention MIPRO*, Opatija, Croatia, May 2012.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Conf. Computer Vision and Pattern Recognition(CVPR)*, pp. 3431-3440, Boston, MA, USA, Oct. 2015.

[11] D. P. Kingma and J. Lei Ba, "ADAM: A method for stochastic optimization," *Published as a Conf. Paper at the 3rd Int. Conf. Learning Representations*, San Diego, 2015.

[12] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in Neural Inf. Process. Syst. 31 (NIPS)*, 2018.

[13] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image analysis using mathematical morphology," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 9, no. 4, Jul. 1987.

[14] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," in *Proc. Comput. Sci.*, Mar. 2015.

[15] N. Srivastava, et al., "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learning Res.*, 15, 2014.

[16] I. J. Goodfellow, et al., "Generative adversarial networks," *Advances in Neural Inf. Process. Syst.*, vol. 27, 2014.

정택현 (Tack-hyun Jung)



2018년~현재 : 건국대학교 IT융  
합정보보호학과 석사과정  
<관심분야> 사이버 보안, 인공  
지능, 컴퓨터 비전  
[ORCID:0000-0002-9172-0817]

김기천 (Keecheon Kim)



1992년 : Northwestern Univ.  
공학박사  
1998년~현재 : 건국대학교 컴퓨  
터공학과 교수  
<관심분야> 통신공학, 사이버  
보안, 미래인터넷, IoT  
[ORCID:0000-0003-3445-3334]