

이미지 학습을 통한 유기견 탐색 모델 구현

김유진(전전컴), 김성훈(통계), 임찬영(통계)

E-mail address: jinyu54@naver.com , sainthun11@naver.com , cksdud585@naver.com

Abstract. 유기견의 발생 횟수는 2016년에는 약 6만 건이었던 반면, 2020년에는 약 9만 건을 기록할 정도로 매년 증가하는 추세이다.¹⁾ 이런 유기견들은 가축이나 사람을 공격하거나 교통사고를 유발하는 등 사회 전반적인 문제를 야기할 수 있다. 따라서 유기견을 효과적으로 구조할 수 있는 시스템의 필요성을 느낀 본 연구팀은 AI를 활용한 유기견 탐색 모델을 제안하는 바이다.

Densenet은 특성상 은닉층이 깊지만, 모든 층에서의 값을 누적 및 전달하기 때문에 보다 정확한 분류를 가능하게 한다. 따라서, 해당 모델을 구현하기에 적합하다고 판단하였다.

1. Introduction

• A. 연구 배경 및 연구 목적

최근 급증한 유기견 문제를 해결하기 위해 지역사회는 유기견 입양에 관해 적극적인 지원을 하는 중이다. 특히 서울시에서는 지난 4월 DB 손해보험과 협업하여, 서울시에서 발생한 유기견을 입양하는 보호자에게 ‘프로미 반려동물보험’의 1년치 보험료를 지원하겠다고 밝혔다(고은하, 유기견 입양시 서울시에서 ‘DB 손해보험의 펫보험’ 선물, 뉴스투데이, 2021). DB 손해보험에서 제공하는 이 보험은 반려동물의 질병이나 상해로 인한 수술비와 치료비를 보장하고, 타인의 반려동물에 손해를 입힐 경우 부담해야 하는 배상책임 손해도 보장해준다. 유기견을 입양하는 보호자에게 실질적인 도움을 줄 수 있는 것이다.

연예인들도 유기견에 관해 많은 관심을 보였는데, 특히 배우 조승우 씨와 유기견 ‘곰자’의 일화가 많은 화제가 되었다. 평소 곰자에 관심이 있었던 조승우 씨는 곰자가 안락사 명단에 뜨고 이를 두고 볼 수 없어 결국 입양을 결정했다. 곰자와 끝까지 하겠다고 밝힌 일화는 많은 사람들에게 잔잔한 감동을 주었다. 또 가수 이효리 양도 직접 유기견 봉사활동을 하며 사회적 동참을 요구해 많은 호응을 이끌어 내기도 했다.

이렇듯 유기견 입양에 관해선 사회적 관심도도 올라가고 지원도 늘어났지만, 정작 유기견을 구조하는 데는 많은 문제가 남아있다. 유기견을 입양하려면 먼저 유기견의 구조가 이루어져야 하는데, 현재 시스템으로는 원활한 구조에 어려움이 있는 것이다.

첫 번째로, 유기견이 발견되기 어려운 곳에서 활동한다면, 혹은 발견되지 않는다면 구조되기 어렵다. 두 번째로, 발견자가 유기견을 발견하더라도 신고를 하지 않을 경우 마찬가지로 구조되기 어렵다. 두 가지 이유 모두 현재 구조시스템이 전적으로 발견자의 신고에 의존하기 때문인데, 결국 신고가 이루어지지 않으면 구조할 시도조차 할 수 없다.

세 번째로, 신고에서 구조까지의 시간이 오래 걸릴 수 있다. 만약 신고자가 유기견을 발견한 곳이 익숙치 않은 장소라면 위치를 설명하는데 어려움을 겪을 수 있고, 유기견의 상태를 제대로 설명하지 못한다면 구조에 필요한 장비를 제대로 준비하지 못해 효율적인 구조가 어려울 것이다.

만약 본 연구팀이 개발한 유기견 탐색 모델을 구조에 활용한다면, 위와 같은 문제들을 해결할 수 있다. 먼저 카메라가 자동으로 유기견들을 탐색해 알려줌으로써 더 이상 발견자의 신고에 의존하지 않아도 된다. 즉 카메라가 있는 곳을 지나가기만 한다면 구조될 확률이 매우 높아지는 것이다. 또한 구조기관에 유기견을 탐색할 때 쓰였던 사진이 전송된다면 유기견의 상태를 명확하고 객관적으로 파악할 수 있어 효율적인 구조가 가능하다. 마지막으로 카메라의 위치 정보도 함께 전송된다면 유기견의 위치를 바로 파악할 수 있어 신속한 구조가 가능하다.

종합하자면, 이번에 구현한 탐색 모델은 기존의 수동적인 탐색에서 벗어나 인공지능을 활용해 최대한 많은 유기견을 신속하고 정확하게 구조하는 것을 목표로 한다. 이렇게 유기견들이 많이 구조된다면 유기견들에 의해 발생하는 여러 가지 사회문제들, 예를 들어 앞서 언급했던 교통사고 유발이나 사람, 가축을 공격하는

1) 유실·유기동물 분석 보고서, 동물자유연대, 2021

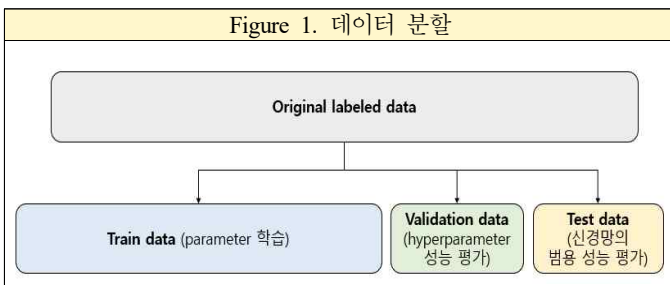
문제들을 줄일 수 있고, 야생에서의 교배가 줄어들어 전체적인 유기견의 수 또한 감소할 것이다.

B. 연구 이론

1. Dataset & ImageDataGenerator

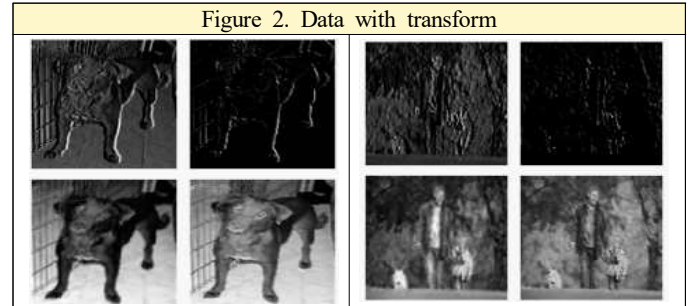
머신러닝 및 딥러닝 모델 구현에 앞서, 데이터를 모으고 처리하는 방법을 정해야 한다. 우선 중복되지 않는 데이터 1200개(유기견 600개, 일반 강아지 600개)를 이용하고자 한다. 단, 유기견으로 분류하는 기준은 ‘사람이 안고 있거나’, ‘목줄만 착용하고 있거나’, ‘목줄과 주인이 함께 있는’ 상황, 총 세 가지로 정하고 각각 200개씩 ‘한 가지 상황에 치우치지 않는 데이터’를 수집한다.

이때, 수집된 데이터를 training set, validation set, test set 총 세 가지로 분류할 수 있다. [Figure 1] 일반적으로는 6:2:2 비율을 사용한다.

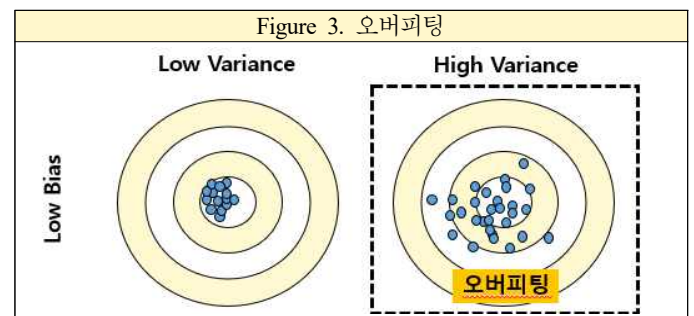


Training set은 말 그대로 훈련용 데이터를 의미하는 것으로, 주로 모델을 학습시킬 때 이용한다. 이렇게 학습된 모델의 성능을 높여가기 위해선, 학습된 데이터가 아닌 ‘새로운’ 데이터가 필요하다. 이를 validation set이라 한다. 마지막으로 test set을 이용해 ‘최종 모델 성능’을 평가한다.

위와 같이 수집한 dataset 각각을 처리해야 한다. 이를 데이터 증강(augmentation)이라 하며, 파이 토치에서는 torchvision.transform을 이용해 구현하는 반면, 텐서플로는 ImageDataGenerator를 선언해야 한다. 그 종류는 다음과 같다. 이미지 크기를 조정하는 rescale, 좌우로 움직이는 범위를 조정하는 width_shift_range, 사진을 확대 혹은 축소하는 zoom_range, 사진을 좌우 반전시키는 horizontal_flip이 있다. 물론 이 외에도, 밝기를 조절하거나, 위/아래로 움직이는 방법도 있다. 다만, 우리는 위에서 정리한 네 가지 방식만을 적용한다. [Figure 2] 또한, train data를 제외한 나머지 data에는 rescale만 적용한다.

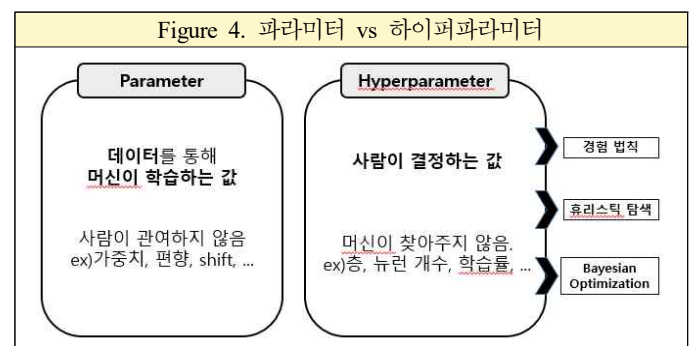


*이것은 오버피팅을 방지하기 위해서 사용한다. 이때, 오버피팅 [Figure 3]이란 학습 데이터에 과도하게 최적화되어, 새로운 데이터에는 올바른 값을 내보내지 못하는 것을 말한다.



2. 하이퍼파라미터 설정

하이퍼파라미터란 모델링할 때, 사용자가 직접 세팅해주는 값을 말한다. 사람이 관여하지 않고, 머신이 데이터를 통해 학습하는 파라미터와는 의미적인 차이가 있다. [Figure 4] 이것은 정해진 최적값이 없어 일반적으로 경험법칙, 휴리스틱한 방법을 사용한다. 다만, Bayesian과 같이 자동으로 하이퍼파라미터를 선택하는 라이브러리를 이용하기도 한다. 이때, 경험법칙은 말 그대로 경험적인 방법에서 도출된 일반적인 수단이나 방침을 말하고, 휴리스틱한 방법은 정보가 부족해 합리적 판단을 할 수 없을 때, 최적화가 될 수 없을 만한 값을 지우는 것을 말한다. 두 방법 모두 과학과는 거리가 멀고, 수행자의 ‘지혜’와 ‘직관’에 의존하는 것을 알 수 있다.



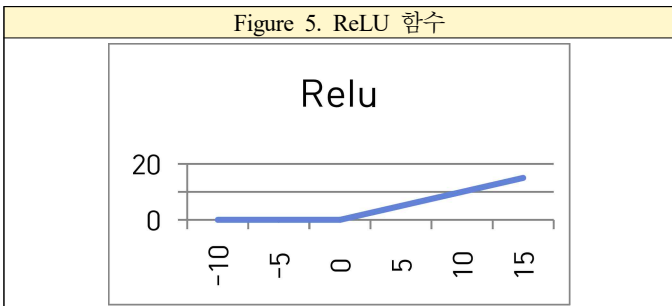
대표적으로는 ‘전체 데이터 셋에 대해 forward pass (or backward pass) 과정을 거친 것’을 의미하는 epoch 이 있다. 즉, 전체 데이터 셋에 대해 한 번 학습을 완료한 상태를 의미한다. 또한, 한 번의 batch마다 주는 데이터 샘플의 size를 batch size라 하고, 데이터를 확인하는 보폭(step size)은 학습률(learning rate)이라 한다. 이는 모두 사람이 설정하는 것으로, 너무 작지도 크지도 않게 설정하는 것이 매우 중요하다. 따라서, 데이터를 개수를 고려해 적절하게 역할을 분배한 후, 최적화 과정을 도입해 나가야 한다.

3. 활성화함수

활성화 함수란 입력 신호의 총합을 출력 신호로 변환해주는 함수이다. 즉 활성화 함수를 통해 변환된 값은 인공신경망에서 한 층의 최종 결과값이며, 이는 다음 층에 입력되는 값이 된다.

$$\text{ReLU} : y_k = \max(0, x_k) \quad (1)$$

[Figure 5] ReLU 활성화함수는 위 (1)식의 형식을 취하는 함수로 입력이 0을 넘으면 그대로 출력하고, 0 이하이면 0을 출력한다. 단순한 알고리즘으로 작동하므로 계산이 매우 빠르고 특히 이미지 분류에서 좋은 성능을 보여준다.



$$\text{Softmax} : y_k = e^{a_k} / \sum_{i=1}^n e^{a_i} \quad (2)$$

Softmax 활성화함수는 (2)식의 형식을 취하는 함수로 출력값끼리의 상대적인 크기 비교가 가능하다. 특히 모든 출력값을 더하면 1이 되므로 확률적 해석이 가능하다.

본 연구팀은 ReLU 활성화함수를 모델에 적용하여 연산량을 줄이고 밝은 픽셀값만을 남겨 학습시킨다. 모델의 마지막 최상위 계층에 Softmax 활성화함수를 적용하여 유기견과 유기견이 아닌 반려견으로 분류한다.

4. 손실함수

손실함수는 예측값과 정답 간의 차이를 수치화해주는 함수이다. 즉, 오차가 클수록 손실함수도 증가하고

반대로 오차가 작을수록 손실함수는 감소한다. 따라서, 우리는 손실함수의 값을 최소화하는 ‘가중치’를 학습해나가는 것이 목표이다.

그 종류로는 크게 평균 제곱 오차와 cross entropy로 구분된다. 평균 제곱 오차는 보통 연속형 변수를 예측할 때 사용된다. 다만 일반적으로는 교차 엔트로피 오차가 사용된다. Log 함수를 이용하기 때문에, 낮은 확률로 예측해서 맞춘 경우나 높은 확률로 예측해서 틀리는 경우 등을 모두 고려할 수 있다. 이진 분류는 binary_crossentropy, 다중 분류는 categorical_crossentropy라 한다.

$$E = -\sum_k t_k \log y_k$$

* 다음으로는, 손실함수를 줄여나가면서 학습할 수 있도록 ‘최적화 방법’을 선택해야 한다.

기본적인 최적화로는 ‘경사하강법(gradient descent)’이 있다. 이 알고리즘을 이용하면 경사를 따라 내려가면서 가중치를 업데이트한다. 이때, 전체 데이터에 대해 계산할 경우 속도, 용량 면에서 성능이 떨어지기 때문에 ‘랜덤으로 선택한 하나의 데이터’에 대해서만 계산하는 방법인 ‘확률적 경사하강법(SGD)’이 제안되었다. 하지만, 이것은 총 세 가지 문제점을 가진다. 첫 번째로는 learning rate가 고정되어 있어 학습속도가 느리다. 두 번째로는 무작위로 sample을 추출하기 때문에 경사 하강에 있어서 불안정하다. 마지막으로 기울기가 0이면 update가 되지 않는다. 따라서, global한 실제 최솟값이 아닌 local한 최솟값인 안장점이나 극소점에 수렴할 수도 있다.

이러한 문제점 때문에 여러 가지 최적화 방법이 제안되었다. 첫 번째로는, 일반적인 경사하강법에 현재의 관성을 고려한 **momentum 기법**이다. 원래 이동해오던 관성의 힘을 이용해 또랑을 벗어날 수 있다. 즉, local한 최소값에 수렴하는 문제를 해결할 수 있다.

다음으로는 ‘변수마다’, ‘스텝마다’ 학습률이 바뀌는 **AdaGrad 기법**이다. 시간이 지날수록 학습률은 줄어듦, 큰 변화를 겪은 변수의 학습률은 대폭 줄이고(이미 최적에 가깝다고 생각), 작은 변화를 겪은 변수는(최적에 아직 멀다고 생각) 소폭 줄인다. 하지만, 이렇게 무한히 학습하면 결국에는 갱신량이 0이 되어 멈춰버린다. 따라서, 과거의 기울기는 잊고, 새로운 기울기 정보를 크게 반영하는 **RMSProp 기법**이 제안되었다.

본 연구팀은 위에서 제시된 RMSProp을 이용해 학습률에 따른 학습속도를 개선하고, Momentum 기법을 이용해 local 최소값을 벗어날 수 있도록 하고자 한다. 따라서, 이 두 가지 기법을 모두 합친 **Adam 최적화 방법**을 사용하고자 한다.

2. 연구 방법

• C. 연구 과정 설명

1. 연구 아이디어

위에서 언급한대로 데이터를 수집 및 분할한다. 단, 데이터가 일반적인 모델들보다 매우 적게 수집되었기 때문에, 6:2:2의 비율이 아닌 8:1:1의 비율로 분할한다.

* 중복되지 않는 데이터 1200개(유기견 600개, 일반 강아지 600개)를 이용하고자 한다. 단, 유기견으로 분류하는 기준은 ‘사람이 안고 있거나’, ‘목줄만 착용하고 있거나’, ‘목줄과 주인이 함께 있는’ 상황, 총 세 가지로 정하고 각각 200개씩 ‘한 가지 상황에 치우치지 않는 데이터’를 수집한다.

수집 데이터 (예시)	
유기견	유기견X (사람)
	
유기견X (목줄)	유기견X (사람+목줄)
	

다음으로는 앞서 설명한 하이퍼파라미터 설정을 진행한다. num_epoch은 60, batch_size는 32로 설정한다. 일반적인 모델보다 적은 데이터로 구현한 만큼 epoch과 배치 크기를 크게 설정할 필요가 없다. 또한, 보폭(학습률)은 0.001로 지정하였다. 다만, Adam 기법 특성상 학습률은 학습을 거듭하며, 학습률은 감소할 것이다.

* 이때, Adam 기법에 주의할 특징이 있다. 이 기법은 학습률을 조정하는 RMSProp 기법의 특징과, 관성을 반영하는 Momentum 기법의 특징을 모두 가지고 있다. 이는 β_1 , β_2 라는 가중치를 이용해 다음과 같이 구현한 것이다. (기본 설정값 : $\beta_1 = 0.9$, $\beta_2 = 0.999$)

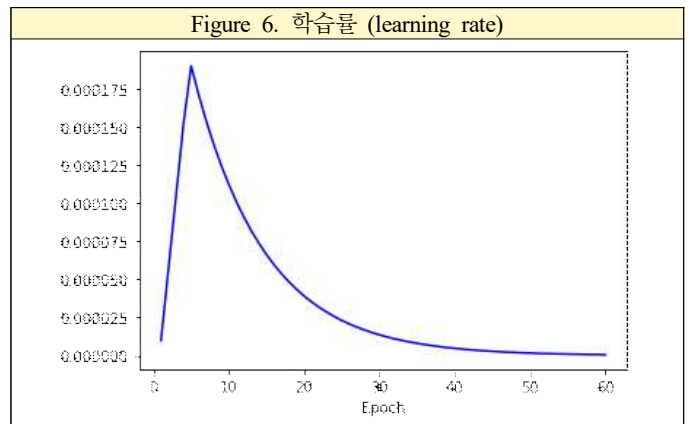
Momentum (가중치 β_1) : $v_n = \alpha v_{n-1} - \eta \nabla f(x_n)$

$$m_n = \beta_1 m_{n-1} + (1 - \beta_1) \nabla f(x_n), \quad m_{-1} = 0$$

RMSprop (가중치 β_2) : $h_n = \gamma h_{n-1} + (1 - \gamma) \nabla f(x_n) \odot \nabla f(x_n)$

$$v_n = \beta_2 v_{n-1} + (1 - \beta_2) \nabla f(x_n) \odot \nabla f(x_n), \quad v_{-1} = 0$$

가중치의 기본 설정값이 1에 근사한다는 것과 초기값이 0이라는 점을 고려하면, 이 기법은 초반에 0에 편향된다. 따라서, 다음과 같은 학습률 보정 작업이 이루어진다.

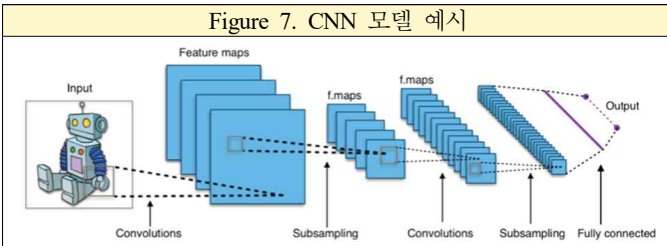


모델이 처음 학습을 진행할 때는 [Figure 6]에서 보는 모습처럼 0.00001의 학습률로 이미지 학습을 시작하지만, 학습하는 총 epoch이 크지 않아 4 epoch을 달성할 때, 0.0002의 최대 학습률을 달성토록 하였다. 즉, 손실 함수를 최저점에서부터 빠른 속도로 정답에 다다르게 하여 정확도를 올리고자 하였다. 그 후 다시 하락해 0.00001의 학습률로 조금씩 학습을 진행토록 해 손실 함수의 최저점을 이탈하지 않도록 보정하였다.

일련의 과정을 거치면, **유기견 탐색 모델 구현**이 가능해진다. 다만, ‘실시간’ 탐색을 위해서는 물체를 추적할 수 있는 코드를 추가해야 한다.

2. 모델 구현

딥러닝은 합성곱 신경망(convolutional neural networks, CNN) [Figure 7]의 등장으로 이미지나 영상에서 각 물체에 대한 특성 맵(feature map)을 추출하는데 큰 발전을 이루었다. Feature map을 추출하기 위해 합성곱 계층(Convolution Layer)과 통합계층(Pooling Layer)을 사용하며, 여기서 추출한 특성 맵을 일반적인 인공 신경망(Fully-Connected Layers)의 입력하여 이미지를 분류할 수 있다.

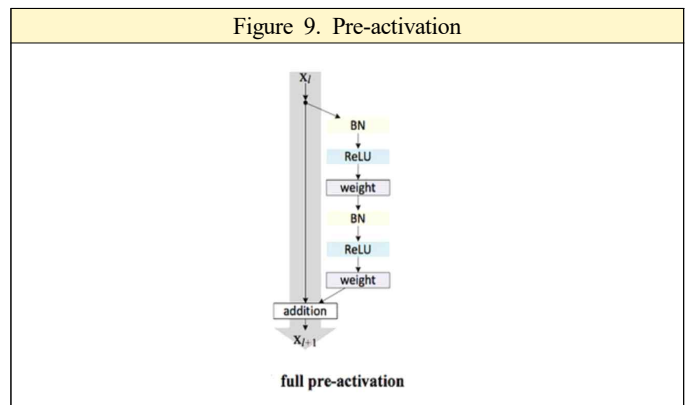
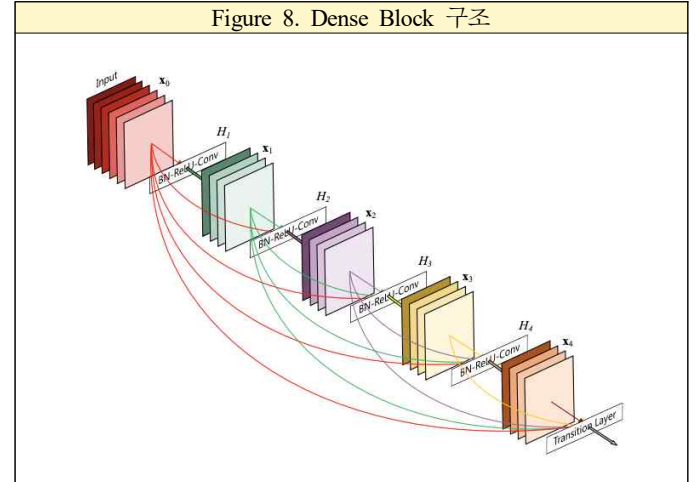


합성곱 계층은 커널을 일정한 간격으로 이동해가며 합성곱 연산을 수행하여 특성 맵을 뽑아내고, 통합 계층은 합성곱 계층에서 출력된 특성 맵의 크기를 줄여 연산량을 줄이는 데 사용된다. 그러나 이 합성곱 계층을 계속 늘린다고 이미지 분류에 대한 정확도가 오르지 않았고 이후 다양한 CNN 모델이 제안되었다.

본 연구에서는 ResNet의 아이디어를 계승하여 최근 객체 구분 영역에서 높은 정확도로 알려진 DenseNet 모델을 사용하였다. Densenet 신경망으로 구성된 모델은 CNN 모델에서 기울기 소실 방지 및 파라미터의 수를 절약하여 연산 속도를 빠르게 하는 데 목적을 둔다. 또한, 모든 층들이 각각의 다른 [Figure 8] Dense Block과 직접적으로 연결되어 있다.

[Figure 9] ResNet 모델과 같이 pre-activation을 적용하여 Dense Block을 배치정규화 - 활성화함수 - 합성곱 연산(BN-ReLU-Conv)의 순서로 구조화하여 최적화가 더 쉬워졌고 오버피팅이 감소하는 결과를 얻었다.

그러나 여기서 DenseNet과 ResNet의 차이점은 특성 맵의 정보를 합칠 때 +연산이 아닌 채널 단위로 concatenate하는 방식을 사용한다는 것이다. concatenate 연산을 통해 이전 정보와 현재 정보가 섞이지 않고 이전 특성 맵을 누적하였다. 이 연산법은 +연산보다 상대적으로 연산량이 적어 모델 적합에 드는 시간을 단축하기도 하였다. 위 구조를 통해 DenseNet은 네트워크 내 계층 간의 정보 흐름을 극대화하고 이전 특성 맵을 재사용할 수 있게 되었으며 기울기 소실문제를 완화시킬 수 있었다.



그러나, 층이 깊어질수록 특성 맵이 누적되어 channel의 개수가 많은 경우 더 급격하게 channel이 늘어나게 되고 연산량이 계속해서 증가하게 된다. 이 문제를 해결하기 위해 **channel의 개수를 작은 값으로 설정하는** 데, 이를 Growth Rate:k이라고 부르며 이는 각 층에서 몇 개의 특성 맵이 생성될지 조절하는 파라미터이다.

[Figure 10] 연산량을 줄이기 위한 방법으로 DenseNet은 특성 맵의 크기를 감소시키는 pooling 연산을 Dense Block 사이에 수행한다. pooling 연산은 BN, ReLU, 1x1Conv, 2x2Average pooling을 수행한다. 이를 Transition Layer라고 부른다. Transition Layer는 0과 1 사이의 값을 가지는 θ 파라미터를 가지고 있어 Transition Layer 입력값의 channel 수가 m이면 $\theta \times m$ 개의 channel 수를 출력하여 channel 수를 줄인다.

또한, Dense Block 계산 시, 층이 깊어지면서 연산량이 급격히 증가하는 것을 막기 위해, 1x1Conv를 이용한 Bottleneck Layer를 사용하였다. 1x1Conv의 filter수만큼 특성 맵의 channel 수가 감소하게 되고, channel 수가 줄어든 특성 맵을 통해 3x3Conv의 연산량을 줄인다.

Figure 10. Densenet 구조 (Dense Block, Transition Layer)

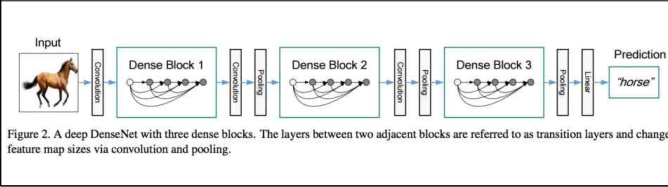


Figure 2. A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature map sizes via convolution and pooling.

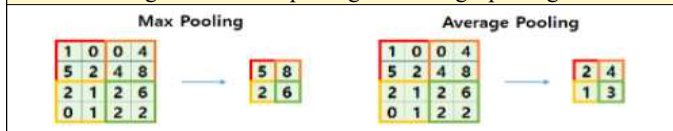
[Figure 11]는 ImageNet dataset에 대한 DenseNet 전체 구조이다. ImageNet dataset은 (224, 224)의 크기를 받고 이 이미지에 대해 DenseNet은 Growth Rate: $k=32$ 를 사용한다. k 가 너무 크기 때문에 연산량을 줄이기 위해 Bottleneck 구조는 필수적으로 사용된다.

* ImageNet dataset는 시각적 객체 인식 소프트웨어 연구에 사용하도록 설계된 대형 시각적 데이터베이스로, 처음 보는 이미지를 넣어도 좋은 capture를 뽑아낼 수 있게 도와준다.

본 연구는 ImageNet dataset에 대한 DenseNet-121을 모델을 기초로 하여 조금 더 정교한 분류를 위해 수정하여 사용하였다. [Figure 11]의 구조에서 DenseNet-121 모델은 이미지를 받아 데이터의 모서리에 0을 둘러주는 zero-padding을 먼저 적용하여 이미지 데이터의 축소를 막고 실제 정보가 있는 데이터를 더 많이 사용하도록 한다.

그 후에 [Figure 11]의 구조처럼 첫 특성 맵을 추출하고 pooling 연산을 수행한다. pooling 연산에는 위에서 설명한 것처럼 특성 맵의 크기를 감소시켜 연산량을 줄이는 기능을 하며 그 중 max-pooling과 average pooling [Figure 13]이 주로 사용된다. max-pooling은 해당 영역에서 최대값을 찾는 연산이고 average pooling은 해당 영역의 평균값을 계산하는 연산이다.

Figure 13. Max pooling & Average pooling



첫 pooling 연산에서는 특성 맵의 해당 영역의 최대값을 찾는 max-pooling 연산을 수행하여 특징을 가장 잘 나타내도록 하고 출력값에 가장 큰 영향을 주도록 하였다. 그 후에는 average pooling을 사용하여 이미지를 부드럽게 해주고 밝은 픽셀, 어두운 픽셀 한쪽으로 출력값이 기울지 않도록 해주었다.

기본적인 구조는 위에서 설명한 DenseNet 구조와 동일하다. Dense Block은 pre-activation을 사용하고 concatenate 연산을 통해 특성 맵을 누적하였고, Dense Block 사이에 Transition Layer 연산량을 감소시키며 DenseNet의 목적에 맞게 적용되도록 구현하였다.

[Figure 11]의 Classification Layer를 수정하여 [Figure 12]과 같은 Classification Layer를 구현하였다. Classification Layer에서 global average pooling을 사용하여 특성을 모두 1차원 벡터로 만들어 연산량을 줄이고 오버피팅을 방지하였다. 이후 바로 Fully Connected Layer에 연결하지 않고 한 번 더 BN, ReLU를 적용하여 과적합을 방지하고 조금 더 정확하게 분류할 수 있도록 하였다. 그 후에 softmax 활성화 함수를 적용한 Fully-Connected Layer를 연결하여 더 쉽게 이미지를 분류할 수 있도록 모델을 구현하였다.

Figure 11. ImageNet에 대한 DenseNet-121 구조

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	1 × 1 conv 3 × 3 conv	× 6	1 × 1 conv 3 × 3 conv	× 6
Transition Layer (1)	56 × 56	1 × 1 conv			
Dense Block (2)	28 × 28	1 × 1 conv 3 × 3 conv	× 12	1 × 1 conv 3 × 3 conv	× 12
Transition Layer (2)	28 × 28	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	1 × 1 conv 3 × 3 conv	× 24	1 × 1 conv 3 × 3 conv	× 48
Transition Layer (3)	14 × 14	1 × 1 conv			
Dense Block (4)	7 × 7	1 × 1 conv 3 × 3 conv	× 16	1 × 1 conv 3 × 3 conv	× 32
Transition Layer (4)	7 × 7	2 × 2 average pool, stride 2			
Dense Block (5)	7 × 7	1 × 1 conv 3 × 3 conv	× 12	1 × 1 conv 3 × 3 conv	× 24
Transition Layer (5)	7 × 7	1 × 1 conv			
Classification Layer	1 × 1	7 × 7 global average pool 1000D fully-connected, softmax			

Table 1: DenseNet architectures for ImageNet. The growth rate for all the networks is $k = 32$. Note that each "conv" layer shown in the table corresponds the sequence BN-ReLU-Conv.

Figure 12. Classification Layer 구조

global_average_pooling2d_2 (Glo	(None, 1024)	0	relu[0] [0]
dense_3 (Dense)	(None, 1024)	1049600	global_average_pooling2d_2[0] [0]
batch_normalization (BatchNorma	(None, 1024)	4096	dense_3[0] [0]
out_relu (ReLU)	(None, 1024)	0	batch_normalization[0] [0]
output_layer (Dense)	(None, 2)	2050	out_relu[0] [0]

3. Feature map

딥러닝은 사람이 특성 맵을 선정하는 것이 아니라 데이터에서 모델을 학습하는 과정에서 목표를 잘 달성할 수 있는 특성 맵(feature map)을 스스로 찾는다.

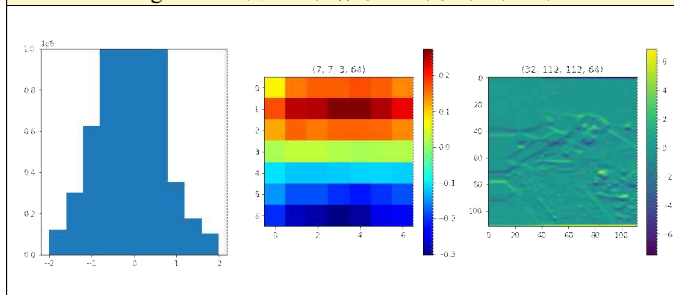
[Figure 14]은 위 사진부터 각각 유기견으로 판단하지 않는 사진과 유기견으로 판단하는 사진을 구현한 모델의 첫 번째 합성곱 계층(Convolution Layer)에서 입력 데이터와 합성곱을 통해 얻은 64개의 첫 번째 특성 맵 중 일부이다.

Figure 14. 첫 번째 특성 맵 (feature map)



주어진 실제 이미지의 정보를 더 잘 사용하기 위해 zero-padding을 해준 후 수행된 첫 번째 합성곱 계층은 64개의 특성 맵을 뽑고 kernel의 크기는 (7, 7), stride는 (2, 2)를 적용하여 [Figure 15]와 같은 7×7 행렬의 가중치를 가짐을 볼 수 있다.

Figure 15. 첫 번째 합성곱 계층의 가중치



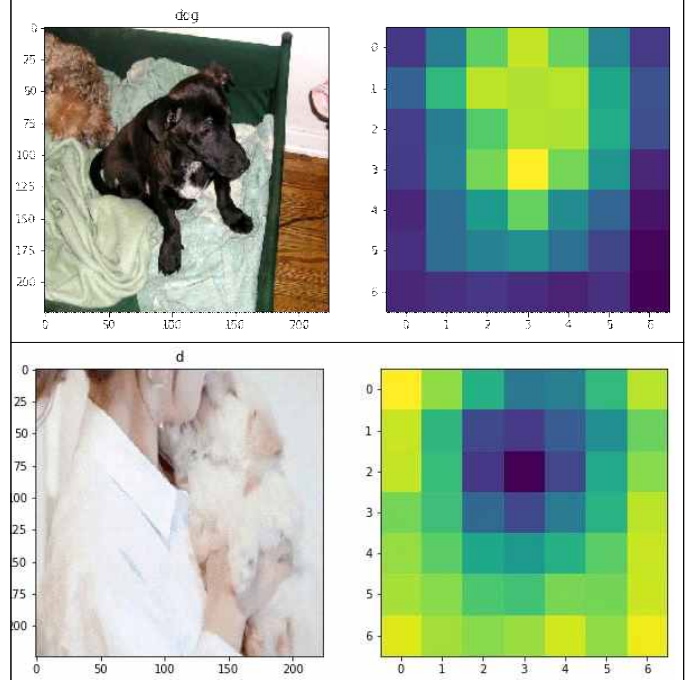
위와 같은 특성 맵을 가지고 최적화를 반복하여 데이터가 모델의 상위계층인 Classification Layer 바로 전 Dense Block에서 ReLU 활성화 함수를 적용한 후 특성 맵에 각각의 가중치를 곱하여 [Figure 16]의 Class Activation Map을 얻을 수 있었다.

[Figure 16]을 살펴보면 유기견으로 판단되는 dog의 이미지는 강아지를 중심으로 큰 가중치를 가지고 있다는 것을 알 수 있고, 유기견이 아닌 반려견으로 판단하는 d의 이미지는 강아지를 제외한 부분을 중심으로

큰 가중치를 가지는 모습을 볼 수 있다.

즉, 유기견과 유기견이 아닌 반려견을 분류하는 데 있어서 강아지 주변에 강아지가 아닌 사람 혹은 목줄 등이 함께 보이는 경우 반려견으로 판단하고 분류한다는 것을 알 수 있다.

Figure 16. Class Activation Map

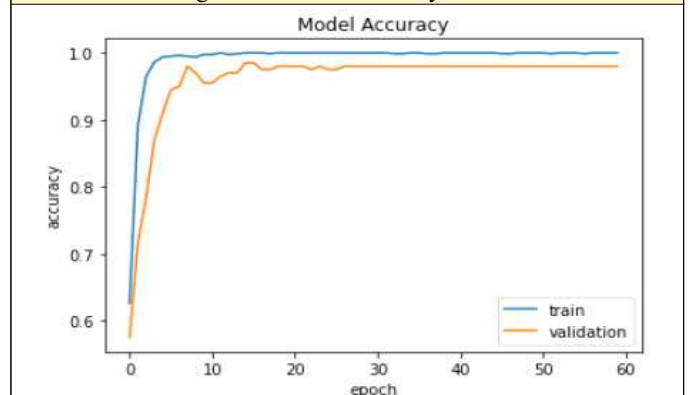


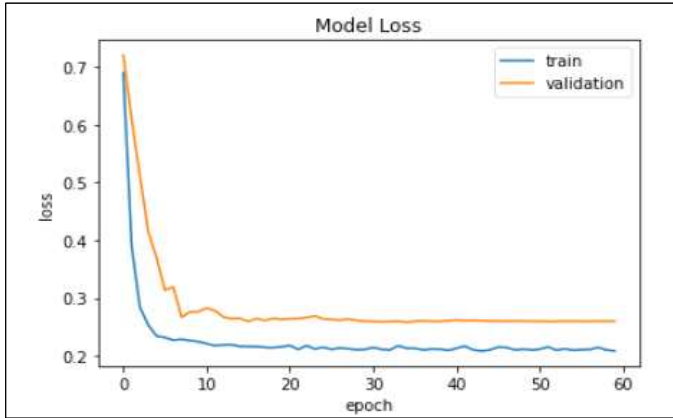
3. 연구 결과

• A. 연구 결과

본 연구팀은 위에서 언급한 바와 같이 ResNet에 기반한 ‘Densenet’ 구조를 이용했다. 이때 논의되는 ‘연산량’, ‘과적합’ 문제를 해결하기 위해 channel 개수에 한계를 두었고, 적절한 연산(pooling) 방법을 선택해 층을 구성하였다.

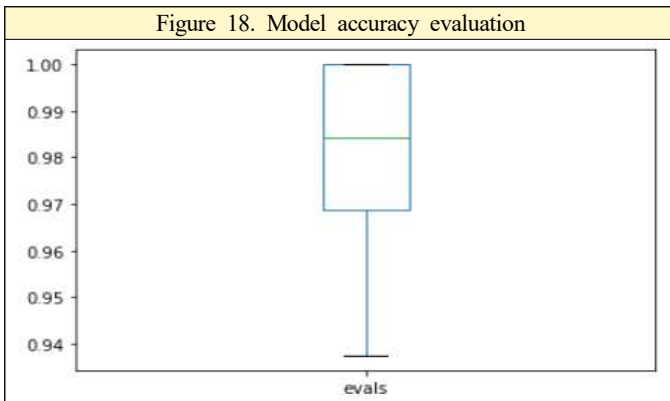
Figure 17. Model accuracy & loss





일련의 과정을 거쳐 구현한 학습 모델의 결과는 다음과 같다. [Figure 17] 모델의 정확도는 97%로 매우 이상적인 결과를 보인다. 또한, 손실함수도 0.25로 예상치와 비슷하며 이는 곧 올바른 최적화 기법을 사용했음을 의미한다.

위 그래프는 Train & Test 작업을 한 번 진행했을 때 결과이다. 우리 연구팀은 Test 작업을 총 20번 진행한 뒤, 정확도를 다음과 같이 [Figure 18] 그래프로 나타냈다. 평균적으로 98% 이상의 정확도를 보이며 연구 결과의 높은 신뢰도를 보장한다.



• B. 연구 한계

우선, 가장 먼저 논의되어야 하는 한계점은 ‘데이터의 부족’이다. Keras에 업로드된 [고양이 vs 강아지 분류]로 등록된 모델은 일반적으로 n 만개 이상의 데이터를 이용했다. 그에 반해, 우리가 구현한 모델은 총 1,200개의 데이터로 그 수가 매우 적다. 따라서, 여러 가지 경우의 수(목줄이 잘 보이지 않는 경우 등)를 고려하지 못했을 가능성이 있다.

다음으로는, ‘실시간’ 구현의 어려움이다. 현재 구현

된 모델은 ‘유기견 탐색 모델’로, 이를 CCTV에 적용하려면 ‘실시간’ 탐색이 동반되어야 한다. 다만, 이를 위해서는 물체를 추적하는 코드가 추가되어야 한다. 또한, 실제로 사용되는 CCTV의 경우 일반 카메라(or 핸드폰)의 화질보다 떨어지는 경우가 많다. 따라서, 화질에 대한 보강 작업이 필요하다.



또 다른 문제점은 유기견과 유기견이 아닌 개를 완벽히 구분해내기 위해선 보다 엄밀한 과정이 필요하다는 것이다. 해당 모델에선 목줄을 찬 경우, 사람과 함께 있는 경우, 목줄·사람과 함께 있는 경우를 유기견이 아닌 경우로 판단하고 나머지 경우를 유기견으로 판단하여 학습을 진행했다. 하지만 이 분류가 유기견이 아니냐를 완벽하게 검증해주지는 못한다. 예를 들어 유기견이 우연히 옆에 지나가는 사람과 같이 사진에 찍혔을 경우, 유기견이 아닌 것으로 인식될 것이다. 또 목줄을 하고 있거나 사람과 함께 있는 개임에도 일부분이 잘려서 찍힌다면 유기견으로 인식될 수도 있다.

마지막으로 똑같은 개체를 식별할 수 있는 기능이 추가되어야 한다. 예를 들어 한 유기견이 같은 장소를 배회할 경우 CCTV에 여러 번 찍힐 수 있다. 이때 같은 개체임을 인식하지 못한다면 구조대에게 여러 번의 알림이 갈 것이고 구조대가 직접 사진을 보고 판단해야 하는 등 구조의 효율성을 떨어뜨릴 수 있다.

4. 결론

• A. 추후 발전 가능성

일반견(주인O or 목줄O)와 유기견(그 외)로 나누어 학습 모델을 구현하였다. 이를 이용했을 때, 추후 발전 가능성에 대해 논의해 볼 수 있다.

첫째로, 탐색 모델을 CCTV에 적용할 수 있다. 프레임에 수집된 오브젝트(유기견)의 유사도를 비교해 종류 및

출현 시간, 이동 경로를 기록할 수 있다. 이는 기존에 제안된 ‘오브젝트 추적’을 이용할 수 있다. 자동차, 새, 나무와 같은 환경과 ‘움직이는 오브젝트’ 즉, ‘유기견’을 구분해 낼 수 있다. 다만, 영상의 노이즈나 비슷한 크기를 가진 동물(ex. 고양이)와의 차별점을 두어야한다. 또한, 주변 밝기 변화와 방향 및 크기에 민감하게 반응하는 SIFT(scale invariant feature transform)을 이용해 구현할 수 있다.

두 번째로, **최적의 유기견 보호소 장소를 발견**할 수 있다. 유기견이 가장 많이 발생하는 지역을 탐색해, 적절한 예산 지원 분배가 가능하다. 실제로, 국회 입법조사처는 2021년 8월, 유기동물 보호시설 확충은 매우 중요한 사안이라고 강조한 바 있다. 유기동물을 관리하기 위한 비용이 급격히 늘어나고 있는 시점에서, 이 모델을 통해 효율적인 장소 선정 및 유기견 확인 및 보호를 이끌 수 있다.



다만, 이를 위해서는 ‘**강아지 품종 분류**’가 추가되어야 한다. 이는 kaggle에서 개최된 prediction competition ‘dog breed identification’을 보면, 총 120종을 90%의 확률로 구분할 수 있음을 확인하는 코드가 제시되어있다. 더 많은 데이터를 수집하고, data augmentation을 높은 수준까지 끌어올린다면 90% 이상의 확률도 기대할 수 있을 것으로 보인다.

5. References

- [1] 남상범, “CCTV 환경에서의 딥러닝 기반 실시간 오브젝트 트래킹 = Deep learning-based real-time object tracking on CCTV”, 광운대학교 대학원, 2018
- [2] 유실·유기동물 분석 보고서, 동물자유연대, 2021
- [3] 이승철, 정해동, 박승태, 김수현, “딥러닝” 소음 진동 27.3, 2017
- [4] 황윤재, 정성원. “CNN 딥러닝을 이용한 범죄 유형별 거리의 물리적 요소 비교” 대한건축학회 학술발표대회 논문집, 2021
- [5] Zhang, Ke, Yurong Guo, Xinsheng Wang, Jinsha Yuan, and Qiaolin Ding. "Multiple Feature Reweight DenseNet for Image Classification." IEEE Access 7, 2019
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. “Identity Mappings in Deep Residual Networks”, Cornell University, 2016
- [7] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. “Densely Connected Convolutional Networks”, Cornell University, 2016



세 번째로는, **유실견(혹은 유기견) 보호소 카페 및 보호소와 협업**할 수 있다. 위와 같이 개발된 ‘유기견 자동 탐색 인공지능’을 바탕으로 이미 등록된 ‘유실견 사진’과 빠르게 비교해볼 수 있다.