Pusan National University

Computer Science and Engineering

Technical Report 2022-09

어류 판별과 무게 예측 시스템 개발

저자 1 : 201624452 김준성

저자 2 : 201814523 유수빈

저자 3 : 201624530 윤도현

지도교수 : 이도훈 교수님

목 차

1. 서론

1.1. 연구 배경

수산시장을 가서 물고기를 사봤던 경험이 있다면 떠올려보자. 물고기의 가격을 정할 때 물고기의 종류와 무게를 토대로 가격을 책정한다. 하지만 무게를 재는 저울이 일정하지 않다면 우리는 원래의 가격에 비해 더 높은 금액을 지불하고 물고기를 구매해야 할지도 모른다. 그리고 더 큰 금액을 지불하고 물고기를 구매했다는 사실조차 깨닫지 못할 수도 있다. 이러한 사기를 치는 행위를 시장에서는 저울치기라고 부른다. 그리고 생선을 샀는데 그 생선이 해당 생선인지 아니면 비슷하게 생긴 다른 생선인지 모른다면 위에 설명한 저울치기 같은 사기 행각을 당한 줄도 모를 것이다.

이러한 사기행각을 구분하여 예방할 수 있는 서비스가 있다면 사기 행각을 일삼는 공급자들을 시장에서 퇴출할 수 있고 소비자들도 안심하고 소비를 할 수 있을 것이다. 그래서 우리는 이러한 서비스의 기초가 될 수 있도록 어류를 판별하고 그 어류에 대한 무게를 추정하는 기능을 구현해보려고 한다. 그리고 구현이 완료되어 서비스화된다면 물고기에 대한 지식을 늘리는 것 또한 효과적일 것이다. 예를 들어, 모르는 물고기가 보인다면 서비스를 이용하여 물고기가 어떤 종류이고 무게는 어떻게 나가는지 알아볼 수 있다.

1.2. 기존 문제점

어류 판별과 길이 측정에 대한 서비스가 독립적으로 존재하였으나 서비스를 합쳐서 운영중인 곳은 없었고 무게 예측까지 해주는 서비스도 없었다.

1.3. 연구 목표

딥러닝을 이용하여 우리나라 근교에서 포획되는 어류 10종을 판별한다. 그리고 컴퓨터 비전을 이용하여 기준 물체를 통해 물고기의 길이를 측정한다. 그중 3종의 어류(도루묵, 참조기, 방어)에 대해서는 무게를 예측하는 시스템 개발을 목표로 한다.

2. 연구 배경

2.1. 데이터 수집

분류 모델 학습을 위한 데이터 수집의 경우 크롤링을 통해 구글, 네이버 등의 웹사이트에서 이미지를 수집하였다. 각 어종당 raw data의 수는 다음과 같다.

표 **1.** 분류 모델 학습을 위한 **raw data** 개수

어종	갈치	고등어	멸치	방어	도루묵	삼치	전갱이	참조기	황아귀	청어
개수	75	125	75	100	150	100	111	159	110	107

각 어종에 대한 raw data 개수는 평균적으로 약 100장 정도이다. 이는 모델 학습을 위해선 부족한 양이라고 판단하여 Data Augmentation을 진행했다. 최종 train data의 수량은 각 어종당 1,000장으로 총 10,000장이다. 다음은 사용한 Augmentation 방법들이다.

a) 이미지 좌우/상하 반전

PIL의 Image를 import 하여 이미지를 좌우/상하 반전시켰다.

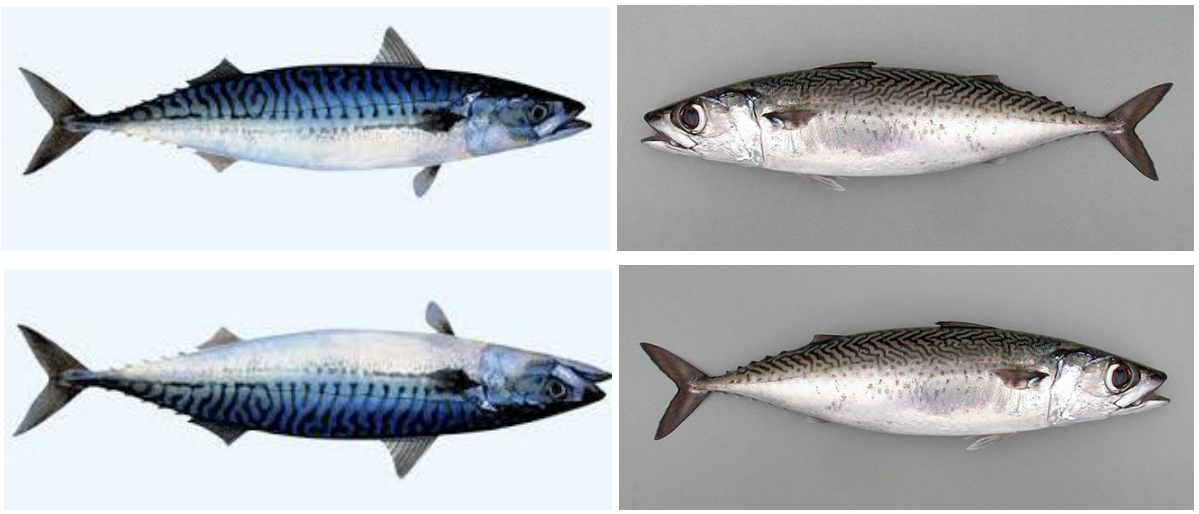


그림 **1.** 이미지를 상하/좌우 반전시킨 예시 사진

b) 이미지 기울이기

imutils를 import 하여 이미지를 기울였다. 이때, 기울인 이미지가 잘리지 않도록 사각 틀에 맞춰서 기울이기 작업을 수행했다.

그림 **2.** 이미지를 기울인 예시 사진

c) noise 추가하기

imgaug.argumenters를 import 했다. 해당 library의 AdditiveGaussianNoise 함수를 통해 이미지에 noise를 추가하였다.

그림 **3. noise**를 추가한 예시 사진

d) brightness/contrast/saturation/hue 변경

torchvision을 import 했다. 해당 library의 ColorJitter 함수를 통해 이미지의 brightness / contrast / saturation / hue를 변경하였다.

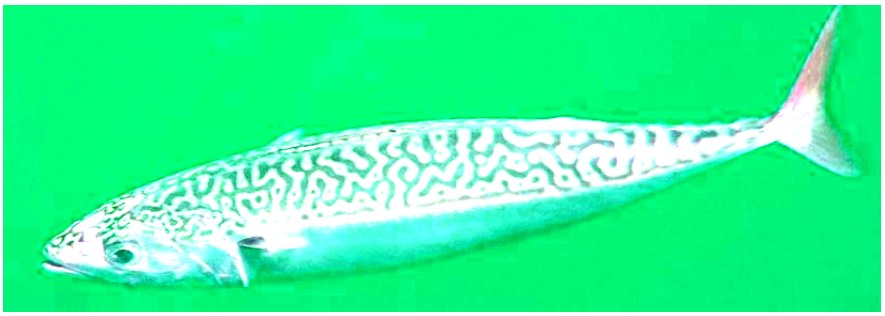


그림 **4. ColorJitter**를 적용한 예시 사진

길이 측정을 위한 데이터의 경우 직접 사진을 찍어 데이터로 이용하였다. 그림 **5**와 같이 길이 측정을 위해선 물고기와 기준 물체인 카드가 함께 있어야 한다.

그림 **5.** 길이 측정을 위한 예시 사진 **-** 기준 물체인 카드와 물고기가 함께 찍혀있다**.**

무게 예측을 위한 데이터의 경우 도루묵, 참조기, 방어에 관한 논문[1], [2], [3] 에서 측정한 무게-길이 데이터를 이용하였다. 어종별 무게-길이 데이터의 개수는 다음과 같다.

표 **2.** 무게 예측을 위한 **raw data** 개수

어종	도루묵	참조기	방어
개수	25	48	9

도루묵과 참조기의 데이터는 선형회귀모델을 훈련시키기에 충분했으나, 방어 같은 경우 데이터의 개수가 상대적으로 적었다. 적은 데이터로 선형회귀모델을 훈련시킬 수 있으나 정확도 향상을 위해 시스템을 사용하며 데이터를 추가하는 방안을 고안해보았다. 이는 향후 개선 사항에서 자세히 설명하겠다.

2.2. 개발 언어 및 도구

i) 개발 언어

Python, JavaScript

ii) 개발 도구

Django, Kaggle, Python Lib : tensorflow, opencv, rembg, gradio

3. 연구 내용

3.1. 데이터 처리

어류 분류 모델을 학습시키기 위해 데이터 전처리 과정이 필요했다. 앞서 언급했듯이 크롤링을 통해서 데이터를 수집하였는데 각각의 이미지 파일들의 크기는 모두 달랐다. 따라서, 학습에 용이하도록 사이즈를 모두 (224, 224)로 재설정하였다. 그리고 크롤링을 통해 수집된 이미지들은 낚시꾼들이 물고기와 함께 찍은 사진이거나, 배경 부분이 깔끔하지 않은 파일들이 대다수였다. 따라서, bounding box 작업을 통해 사진에서 학습에 방해되는 부분을 제거하고 필요한 부분만을 추출하였다.

그림 **6. bounding box** 작업 예시사진 **- bounding box** 작업을 통해 불필요한 부분을 제거하였다**.**

3.2. 모델 설계

i) 분류 모델

물고기 분류 모델의 경우 3개의 분류 모델을 학습시킨 후 앙상블 기법을 이용하였다. 각 모델의 학습을 통해 나온 정확도와 평가 점수를 이용하여 결과를 선택하도록 설계하였다. 3개의 모델은 최근에 나온 [4],[6]EfficientNet과 [5],[7]NFNet을 사용하기로 하였다. 먼저 EfficientNet은 이전의 모델들보다 파라미터가 작은 대도 불구하고 좋은 성능을 발휘한다. EfficientNet은 다양한 scaling 방법들을 조합하여 성능을 높여보자는 취지에서 제안된 모델이다. 기존 cnn 모델들이 Model scaling을 할 때 하나의 종류만을 이용하여 진행하는 반면 EffieientNet의 경우 여러 종류를 조합하여 진행한다. Model scaling 방법으로는 Filter(이미지 분류에 사용되는 Feature를 추출)의 크기를 늘리는 width Scaling, Layer를 늘리는 Depth Scaling, 분류에 사용되는 이미지의 해상도를 크게 하는 Resolution Scaling이 있다.

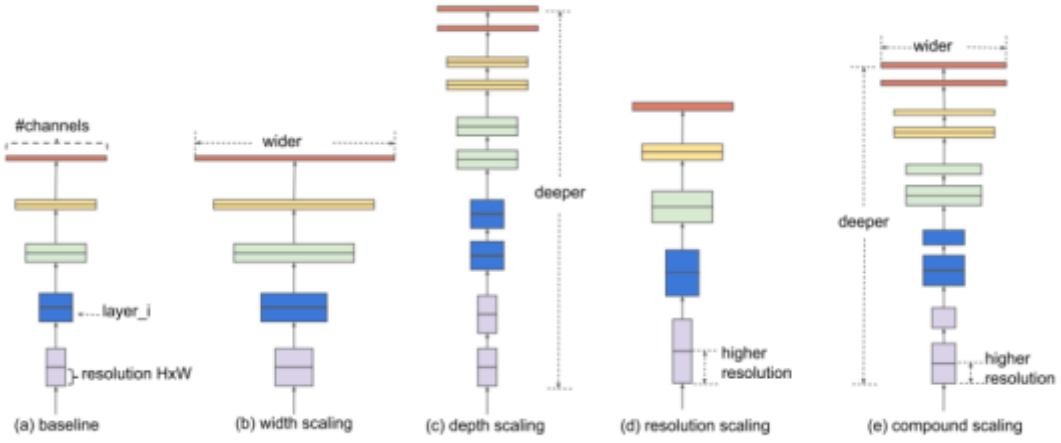
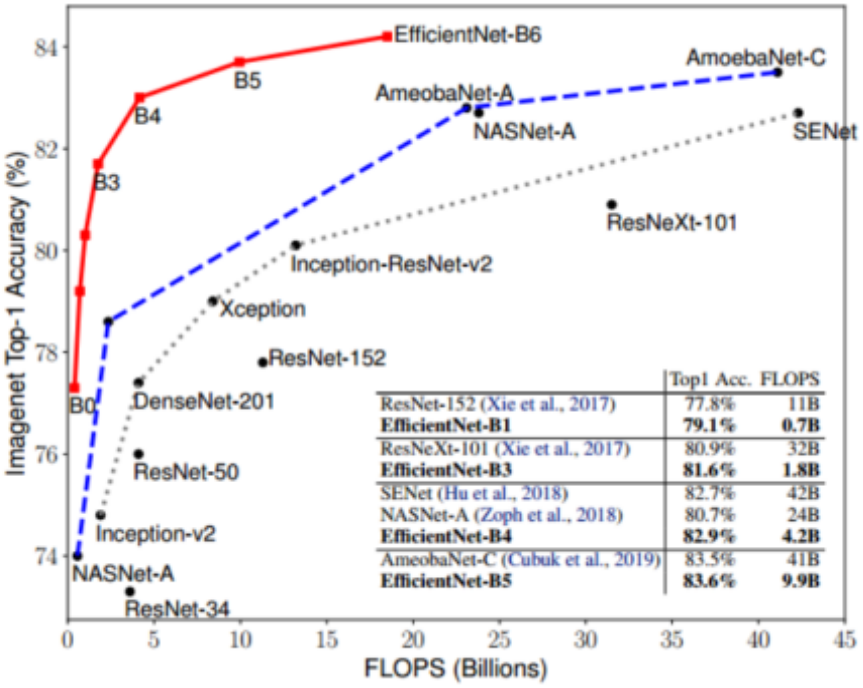


그림 **7. Model Scaling** 종류

EfficientNet은 이 3가지 종류의 scaling을 여러 차례 조합하고 적합한 조합을 찾아내어 적용한 모델이다. EfficientNet은 이러한 조합을 compound scaling이라고 부른다. 그림 8은 EfficientNet과 다른 모델의 성능을 비교한 그림이다.



	Top1 Acc.	FLOPS
ResNet-152 (Xie et al., 2017)	77.8%	11B
EfficientNet-B1	**79.1%**	**0.7B**
ResNeXt-101 (Xie et al., 2017)	80.9%	32B
EfficientNet-B3	**81.6%**	**1.8B**
SENet (Hu et al., 2018)	82.7%	42B
NASNet-A (Zoph et al., 2018)	80.7%	24B
EfficientNet-B4	**82.9%**	**4.2B**
AmeobaNet-C (Cubuk et al., 2019)	83.5%	41B
EfficientNet-B5	**83.6%**	**9.9B**

그림 **8. EfficientNet** 성능 비교 차트

그래프에서 x축의 FLOPS는 연산량을 의미하고 y축은 각 모델의 정확도를 의미한다. 그래프를 보면 efficientNet이 다른 모델들에 비해 적은 연산량으로 높은 정확도를 보여준다는 것을 알 수 있다.

다음으로 NFNet은 2021년에 발표된 모델로 기존 모델에서 사용하던 배치 정규화 과정을 없앤 모델이다. 배치 정규화란 활성화 함수의 출력값이나 결과값을 정규화하는 것을 말한다. 이러한 배치 정규화 과정을 거치면 학습을 효율적이고 빠르게 진행할 수 있다는 장점이 있어 많은 모델에 채택되어 사용되고 있었다. 하지만 미세 조정(fine-tuning)이 필요한 다른 하이퍼파라미터 필요, 계산 비용 증가(연산량 증가), 분산 학습에서 에러 발생 가능성 등의 단점도 가지고 있다. 그래서 NFNet은 배치 정규화 대신 드롭아웃과 Adaptive Gradient Clipping(AGC)을 사용한 모델이다. Gradient Clipping는 기울기(gradient)가 특정 임계 값을 초과하지 않도록 하여 모델 학습을 안정화시키는 방법으로 Adaptive Gradient Clipping는 이 방법의 종류 중 하나이다. 그림 9는 NFNet의 성능을 보여주는 그래프이다.

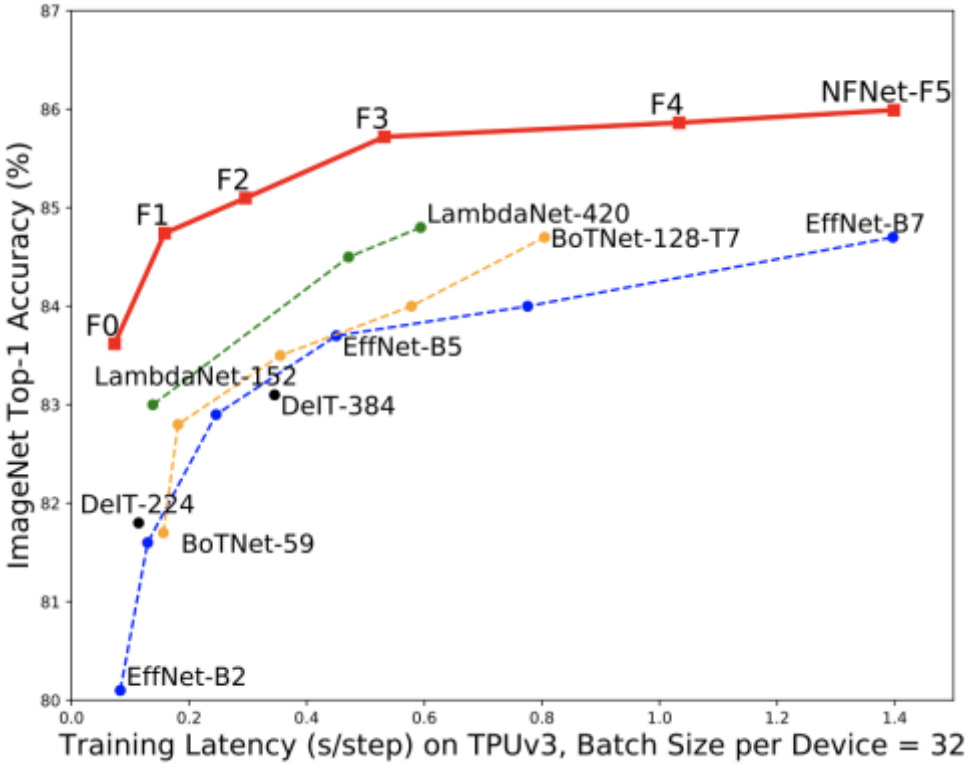


그림 **9. NFNet** 성능 비교 차트

EfficientNet 2가지와 NFNet 모델에 앙상블 기법을 적용하여 분류 모델을 설계하였다. 이미지를 Input하면 학습된 모델로 각각 통과하여 확률값을 얻는다. 각 모델의 정확도를 weight로 활용하여 확률값과 연산을 진행 후 나온 출력값을 SoftMax 함수를 통해 간단히 만들어 가장 높은 값을 최종 결과값으로 선정한다.

ii) 길이 측정 모델

openCV를 통해 대부분의 기능을 구현했다. 길이 측정의 전체적인 흐름은 다음과 같다. 기준물체인 카드와 물고기가 같이 찍힌 이미지 파일을 웹을 통해 입력받는다. 카드는 대부분 사람들이 흔히 소지하고 있는 물품이고, 규격이 통일되어 있기 때문에 기준물체로 선정하였다. 이후, rembg.bg를 통해 입력받은 파일에서 카드와 물고기를 제외한 배경 부분을 제거한다. 더 정확한 물체 인식을 위해 필요하다. 2가지 물체(물고기, 카드)가 인식되었다면 각각의 물체에 사격형 contour를 그린다. 2개의 contour가 생겼을 경우, 각 contour의 픽셀 수를 계산할 수 있다. 이때, 카드의 규격인 85.60mm x 53.98mm를 이용한다. 신용카드의 가로와 세로의 비율은 약 1.585이다. 2가지 물체의 가로세로 비율을 구하여 카드의 비율과 근사하면 카드로 인식하였다. 물체를 인식하여 contour를 그릴 때, 가로세로의 오차가 발생할 수 있으므로 오차범위를 ±5%로 두었다. 즉, 가로 세로 비율이 1.506보다 크고 1.665보다 작다면 카드로 인식한다. 이후, 카드의 길이를 통하여 cm당 픽셀 수를 계산한다. 최종적으로 cm당 픽셀 수를 이용하여 구하고자 하는 물고기의 길이를 측정한다.

위와 같은 과정으로 카드를 통해 물고기의 길이를 측정한다. 카드의 비율을 통해 카드를 인식한다면 잘못 인식할 경우가 있다고 생각할지도 모른다. 하지만, 대부분 물고기들은

카드와 달리 세로 길이보다 가로 길이가 더 길기 때문에 잘못 인식할 확률은 매우 적다고 판단하였다. 그리고 길이 측정을 위해 몇 가지 가정을 두었다. 어떤 가정을 두었는지는 뒤에 향후 개선 사항에서 다시 말하겠다.

iii) 무게 예측 모델

무게 예측은 선형회귀를 이용했다. 일반적으로 물고기의 무게와 길이는 비례관계로, 길이가 길면 무게가 많이 나갈 것이라고 예측 가능하다. 이를 토대로 물고기의 무게와 길이에 대한 임의의 선형식 W = aL + b을 세울 수 있다. 여기서 W는 물고기의 무게, L은 물고기의 길이이고 a와 b는 각각 가중치와 절편이다.

어종별 무게와 길이 데이터를 이용해 평균제곱오차를 최소화하는 가중치와 절편을 찾고, 이 값을 저장하여 주어진 길이에 대한 무게를 예측한다.

3.3. 모델 학습

각 어종당 1,000장씩 총 10,000장의 data를 통해 모델 학습 및 평가를수행했다. train data와 test data의 비율은 7:3으로 두었다. 즉, 어종당 700장은 학습을 위해 사용되었고, 300장은 평가를 위해 사용되었다. 그리고 학습을 위해 모든 이미지 파일들의 크기를 (224, 224)로 resize했다. 학습 및 평가를 위한 전체 코드는 Github에서 확인할 수 있다.

i) 학습 방법

하나의 model을 만들기 위해 3가지 trained model을 사용하는 앙상블 기법을 사용했다. 각각의 모델의 종류는 efficientnet_b0 / efficientnet_b4 / nfnet_10 이다. batch size는 32, learning rate는 0.0001, epoch는 5로 설정하여 각 model을 학습하였다. 학습 종료 후 각 모델의 accuracy는 순서대로 [99.84, 99.71, 99.6] 이다. 각 model의 정확도를 weight로 두어 최종 결과를 판단하였다.

batch size와 learning rate, epoch를 각각 32, 0.0001, 5로 설정한 이유는 다음과 같다. 우선 batch size의 경우 일반적으로 32, 64로 설정하는 것이 성능에 좋다고 알려져 있다. 그리고 좋은 학습률은 일반적으로 매우 낮은 학습률 0.00001부터 10까지 조정하여 찾을 수 있다. Loss가 크면 학습률을 줄이고, 학습 시간이 매우 오래 걸리면 학습률을 크게 하는 방식을 통해 0.0001이 적합하다고 판단하여 해당 학습률로 설정하였다. 마지막으로 epoch는 5로 설정하였을 때, 3가지 모델 모두가 99.5% 이상의 정확도를 가지게 되었다. 99.5% 이상의 정확도는 미리 정해둔 기준보다 높기 때문에 epoch를 5로 설정하였다.

ii) 평가 기법

앞서 학습한 앙상블 model에 각 어종당 300장의 test date를 적용한 결과는 그림 10과 같다. label 부분은 Mac으로 작업을 수행하여 한글 깨짐 현상이 나타났다.

그림 **10. test data**에 **ensemble model**을 적용한 결과

3.4. 인터페이스

실제로 동작을 확인하기 위해서는 시스템이 올라가야 해서 웹 화면을 구성하기로 하였다. 기본적으로 연구실에서 사용하고 있는 웹 화면을 물고기라는 테마에 맞게 배경 화면이나 폰트 색들을 수정하였다. 분류 모델이나 길이, 무게 예측 모델들은 gradio를 이용하여 동작하도록 설계하였다.

3.5. 수정사항

i) 데이터 수집

국립수산과학원을 통해 데이터를 수집하려고 했었다. 하지만, 사익을 위한 목적으로 데이터가 제공될 수 없다는 답변을 들어서 크롤링을 통해 데이터 수집을 진행하였다.

ii) 분류 모델

분류 모델의 경우 앞서 말한 바와 같이 AlextNet, Xception, VGGNet 대신 EfficientNet과 NFNet을 사용하기로 변경하였다.

4. 연구 결과 분석 및 평가

4.1. 결과

i) 분류 모델

그림 11은 실제 모델이 웹에서 동작하는 사진이다.

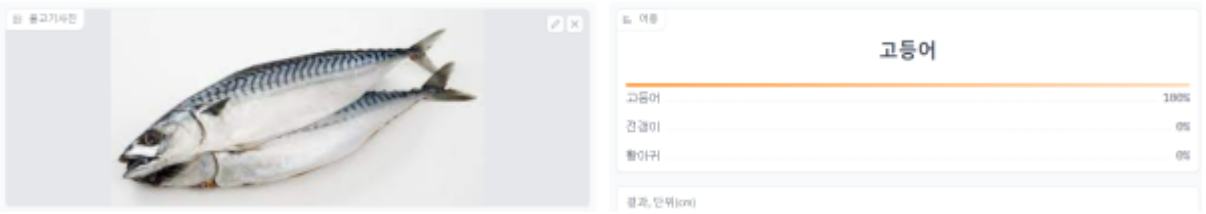


그림 **11.** 분류 결과

ii) 길이 측정 및 무게 예측

그림 **12.** 길이 및 무게 예측 결과

iii) 인터페이스

그림 **13.** 인터페이스 전체 모습

그림 **14.** 사용법 및 주의사항 화면

4.2. 평가

i) 분류 모델

테스트는 웹 사이트의 이미지를 통해 진행하였으며 도루묵, 참조기, 삼치를 제외한 나머지 어종은 모두 **20**장의 다른 사진들을 사용하였다.

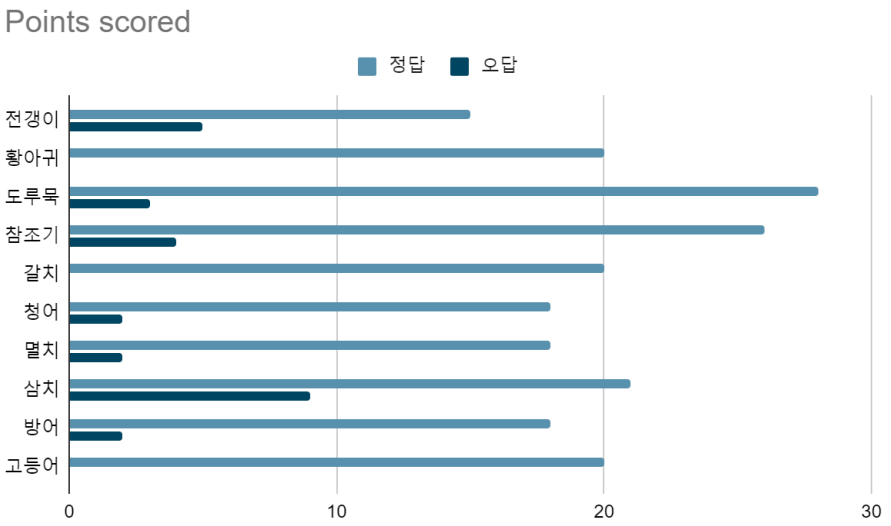


그림 **15.** 물고기 종류 예측 결과 차트

테스트를 진행하여 본 결과 10가지 어종에 대해서 대부분 정확하게 측정하였다. 하지만, 삼치의 경우 사진에 따라 고등어나 갈치 등으로 잘못 판별하는 경우가 발생하였다. 그리고 다음과 같이 사진에 여러 마리의 어류가 있는 경우 갈치로 오인하는 경우가 있었다.

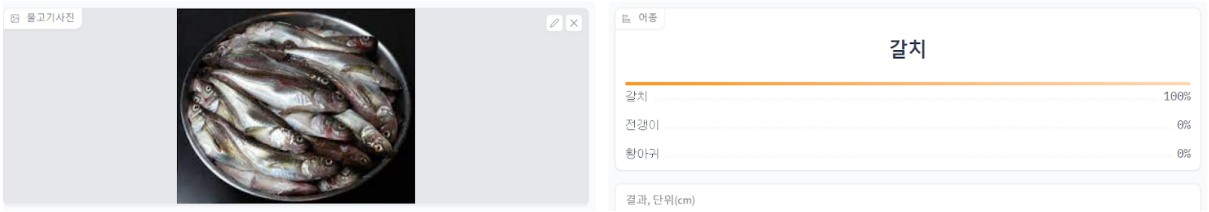


그림 **16.** 여러 마리의 물고기 예측 실패

ii) 길이 측정 및 무게 예측

10가지 종류의 물고기 중 쉽고 저렴하게 구입할 수 있는 물고기를 조사해보았다. 그 결과, 참조기가 가장 적합하다고 판단했다. 이후, 참조기 10마리를 실제로 구입하여 길이와 무게 측정을 한 뒤, 길이 측정 및 무게 예측 기능을 통해 얻은 결과값과 비교해보았다.

표 **3.** 실제 물고기 측정 **Test Data :** 파란색 오차율은 실측값보다 예측값이 낮은 것을**,** 빨간색 오차율은 실측값보다 예측값이 높은 것을 의미한다**.**

물고기 번호	길이**(cm)**			무게**(g)**		
	프로그램	실측 값	오차율**(%)**	프로그램	실측 값	오차율**(%)**
1	20.7	21	-1.44	97.8	83	17.8
2	20.4	21	-2.94	92.5	84	10.1
3	20.3	20.7	-1.47	91.7	84	9.2
4	20.3	21.5	-5.91	90.5	86	5.2
5	20.5	21	-2.44	95	81	17.3
6	20.3	21.3	-4.92	91.1	91	0.1
7	19.7	20.8	-5.58	79.7	89	-11.7
8	20.2	20.5	-1.48	89	80	11.3
9	19.5	20.5	-5.12	75.4	82	-8.7
10	19.5	20.3	- 4.10	76.8	84	-9.4

표 3의 길이 부분을 보면 참조기 10마리 모두가 실측값보다 프로그램을 통해 예측한 값이 더 낮다는 것을 알 수 있다. 이러한 결과가 나온 이유는 rembg를 통해 배경 화면을 제거할 때 생긴 것으로 판단된다. 물고기의 꼬리 끝부분은 뚜렷하지 않아 배경 화면과 구분이 잘 안되어서 잘렸을 수 있기 때문이다.

표의 무게 부분을 보면 7마리의 참조기에서 실측값보다 프로그램을 통해 예측한 값이 더 높은 것을 확인할 수 있고, 길이와 달리 오차율도 더 크다. 이러한 결과는 물고기의 자연 감량이 원인으로 파악된다. 실제 사용한 무게-길이 데이터는 어획한 후, 죽은 지 얼마 되지 않은 물고기를 측정하였다. 반면, test할 물고기를 구할 때는 잡은 지 얼마 안 된 활어를 구하기 어려웠다. 따라서, 마트에서 구입한 물고기를 이용하여 측정했는데, test에 사용된 물고기는 이미 유통 과정에서 건조되어 무게가 감량되었다. 따라서, 예측값보다 실측값이 더 낮게 나온 것으로 판단된다. 또, 비늘이 이미 제거되어 있었는데 이러한 이유도 오차에 영향을 줬다.

구입한 참조기들은 마트에서 소비자들에게 팔기 위한 상품이었기 때문에 길이와 무게가 거의 비슷하다. 다양한 길이를 가진 활어들을 구할 수 있었다면, test에 신뢰도가 올라갔을 것이다.

5. 결론 및 향후 연구 방향

5.1. 결론

현재까지의 결과로 보아 최초 계획했던 시스템 기능의 구현은 완료되었다. 하지만 각 기능의 성능 면에선 개선해야 할 점이 많이 있다고 생각한다. 비록, 본 과제에서는 시간과 자원의 한계로 인해 완벽한 기능의 서비스를 구현하지 못했다. 하지만 데이터 수집 및 전처리 과정, 모델 학습 및 평가, 다양한 Libraries(openCV, gradio 등)의 사용까지 딥러닝에 대한 전반적인 과정을 모두 경험해보았다. 또한, 현재 시장에는 물고기의 분류 또는 길이 추정 각각에 대한 서비스만 존재하고 무게 예측에 대한 서비스는 전무하다. 이러한 점을 생각해보면, '어류 판별과 무게 예측 시스템'의 초석으로서 의의가 있다.

시스템을 개발하며 아쉬웠던 점들이 존재한다. 시스템의 특정 부분의 기능을 어떠한 방법으로 개선할지를 알고 있기 때문이다. 추후에 시간과 자원에 대한 지원이 있다면 해당 시스템에 대한 개선점을 수정하여 더 좋은 서비스를 만들 수 있을 것으로 기대된다. 향후 개선 사항에 대해서 알아보자.

5.2. 향후 연구 방향

i) 분류

분류할 때, 단일 어종의 경우 판별이 잘 되었으나 여러 어종이 같이 있다면 어류 판별이 힘들었다. 먼저 물체 인식을 진행한 후, 개별적인 판별이 이루어지도록 시스템이 수정된다면 여러 마리의 물고기에 대한 판별이 가능해질 것이다. 그리고 단일 어종의 경우 학습 데이터 셋을 늘리고 Resolution Scaling을 진행한다면 정확도가 향상될 것으로 보인다.

ii) 길이 측정

길이 측정을 위해 여러 가지 가정이 필요하다. 각 가정을 없앤다면 더 좋은 서비스가 될 것이다. 따라서, 각 가정에 대한 설명과 개선점을 생각해보았다. 3가지 가정을 살펴보자.

a) 물고기를 잘 인식하도록 바닥의 패턴을 단순화해야 한다.
b) 물고기와 카드 2가지 물체만 인식되어야 한다.

c) 물체와 수직으로 사진을 찍어야 정확한 길이가 측정된다.

다음과 같이 3가지 가정이 있다. a, b는 모두 하나의 문제점으로 인해 생긴 가정들이다. 카드를 카드로 인식하고, 물고기를 물고기로 인식한다면 해결될 문제이다. 현재 사용하고 있는 방법은 물체 2개만을 인식하고, 해당 물체 중 가로 세로의 비율이 카드 비율의 오차범위 내에 있다면 카드로 인식하여 cm당 pixel을 구하는 것이다. 물고기와 카드를 잘 인식하기 위해 바닥의 패턴이 단순화되어야 하고, 카드 이외에 나머지 물체를 물고기로 가정하기 때문에 b 가정이 필요하다. 따라서, 이러한 가정들을 없애고 더 좋은 서비스를 제공하기 위해서는 카드를 카드로 인식하고, 물고기를 물고기로 인식하면 된다. 이를 위해 생각한 개선 방법은 YOLO(You Only Look Once) 등을 이용하여 카드와 물고기를 학습시켜서 인식하도록 하는 것이다. 이런 방법을 사용한다면 a, b 가정들이 필요 없게 되어 더 나은 성능을 가진 서비스가 제공될 수 있다.

다음 가정은 물체와 수직으로 사진을 찍어야 정확한 길이가 측정된다는 것이다. 카메라의 각도에 따라 cm당 pixel이 달라지고 이는 부정확한 길이 측정으로 이어지기 때문에 생긴 가정이다. 이러한 가정을 없애기 위해선 영상의 기하학적 해석을 통해 정확한 길이 측정을 하는 것이다. 하지만 이는 아직 활발히 연구되지 않는 영역이라 접근이 어려웠다. 현재 생각나는 간단한 해결책은 사용자에게 수직 촬영을 요구하는 것이다.

추가로 기준 물체의 종류를 늘리는 것도 서비스 편의성 향상을 도모한다. 현재 서비스의 기준 물체는 카드 1가지만 존재한다. 만약 사용자가 카드를 들고 있지 않다면 서비스 이용이 제한되는 것이다. 따라서, 사용자들이 흔히 가지고 다니는 물체들을 여러 가지 추가한다면 편의성이 향상될 수 있다.

iii) 무게 예측

무게 예측 모델에서 데이터의 개수는 매우 중요하다. 단순 선형 회귀 모델을 사용했기 때문에 데이터 개수에 따라 정확도가 크게 차이 났다. 방어의 경우 9개의 데이터를 사용했는데 기능 구현은 가능했다. 하지만, 선형 회귀 모델을 그려보았을 때 데이터 개수가 많은 참조기와 다른 양상을 보인다.

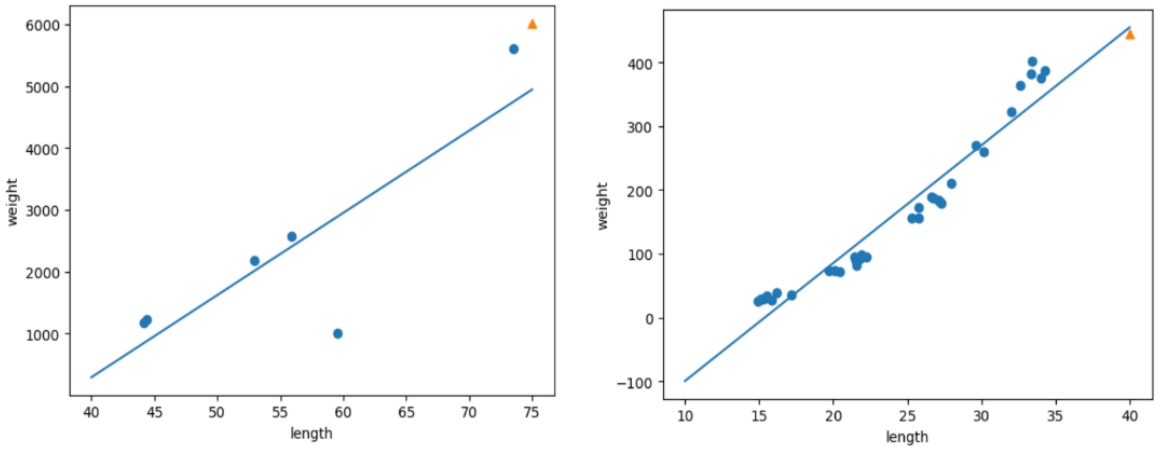


그림 **17.** 선형 회귀 모델 **(**방어**,** 참조기**)**

정확도를 향상시키기 위해 데이터 개수를 늘려야 한다. 일반적으로는 시스템을 구축하기 전에 개발자가 직접 데이터를 수집한다. 하지만, 우리 시스템의 특성상 사용자에게 물고기 사진을 입력받는다. 즉, 대부분의 상황에서 사용자에게도 물고기 데이터가 있다.

낚시, 수산물 시장 등에서 물고기를 앞에 두고 시스템을 사용한다면 길이와 무게에 대한 정보를 실제로 얻을 수 있다. 이때, 사용자에게 무게 정보를 요구하여 선형 회귀 모델을 업데이트시킬 수 있다. 다양한 길이와 무게 데이터를 얻을 수 있는 것이다. 또한, 사용자가 늘어나며 데이터가 추가될수록 정확도가 향상된다는 장점이 있다. 따라서, 향후에 개발자가 데이터를 추가하는 방법과 사용자에게 데이터를 요구하는 방법을 둘 다 사용하여 정확도를 개선할 것이다.

6. 참고 문헌

[1] S. Lee, J. Yang, H. Cha, S. Yoon, D. Jang, Y. Chun, “Age and growth of the sandfish, Arctoscopus japonicus in the East Sea of Korea”, *J Korean Soc Fish Ocean Technol,* Vol. 44, No. 4, pp. 312-322, Dec. 2008. (in Korean)

[2] I. Yeon, D. Lee, J. Lee, K. Choi, B. Hong, J. Kim, Y. Kim, “Long-term changes in the small yellow croaker, Larimichthys polyactis, population in the Yellow and East China Seas”, *J Korean Soc Fish Ocean Technol,* Vol. 46, No. 4, pp. 392-405, 2010. (in Korean)

[3] D. Chang, J. Yoo, B. Kim, S. Lee, D. Kwon, J. Koo, G. Ahn, I. Oh, “A characteristics on the forming of fishing ground and population ecological study of Yellow tail, Seriola quinqueradiata, in the coastal waters off Gim-nyeong of Jeju Island, Korea”, *J Korean Soc Fish Ocean Technol,* Vol. 46, No. 4, pp. 406-415, 2010. (in Korean)

[4] Mingxing Tan, Quoc V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks” 28 May 2019

[5] Andrew Brock, Soham De, Samuel L. Smith, Karen Simonyan, “High-Performance Large-Scale Image Recognition Without Normalization” 11 Feb 2021

[6] NFNet 논문리뷰[Online].

https://jeonghwarr.github.io/review/nfnet___high_performance_large_scale_image_recognition_without_normalization/ (downloaded 2022, Sep, 28)

[7] EfficientNet 논문리뷰[Online]. https://hoya012.github.io/blog/EfficientNet-review/ (downloaded 2022, Sep, 28)