

# Busan science high school

# 2023 Ocean ICT Festival **2023 BOIF**

B QR 코드 영역 QR 삽입 후 테두리 삭제 26

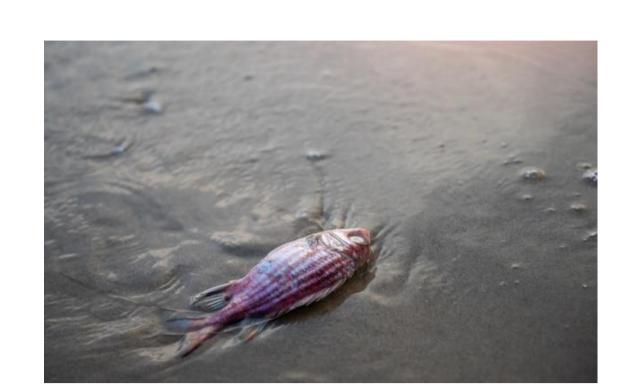
Youtube 영상 QR

# 해양생물정보제공

팀명: 해양 animal information 제공

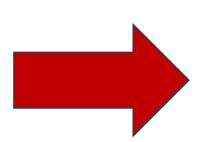
팀원: 2216 유상현 2220 황성빈

01





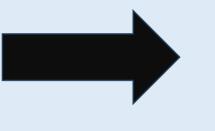
- 물고기를 잡았을 때 물고기의 <u>종류를 알지 못함</u>
- 2. **물고기에 대한 정보**도 빠르고 쉽게 알고 싶음



## 해양생물정보제공 프로그램 제작

02

사진 제공





#### 데이터 처리

#### ImageDataGenerator를 활용한 Image Data **Augmentation**

```
from PIL import Image
import os, glob, numpy as np
file_dir="augmented_images_"
categories=["godeungeo", "bangeo", "samchi", "myeolchi", "jeongaengee", "galchi"]
pixels=image_h*image_w*3
for idx, cat in enumerate(categories):
    label=[0 for i in range(nb_classes)]
     image_dir=file_dir+cat
     files=glob.glob(image_dir+"/*.jpeg")
     for i, f in enumerate(files):
          img=Image.open(f)
         img=img.convert("RGB")
img=img.resize((image_w, image_h))
         x.append(data)
         y.append(label)
x=np.array(x)
```

x\_train, x\_test, y\_train, y\_test=train\_test\_split(x,y)

### 학습

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (
     # Assuming you have ROB images in x_train and x_test
# Convert ROB images to grayscale and resize to (28, 28)
x_train_gray = np.array([cv2.cvtColor(cv2.resize(img.astype(np.uint8), (28, 28)), cv2.COLOR_RGB2GRAY) for img in x_train])
x_test_gray = np.array([cv2.cvtColor(cv2.resize(img.astype(np.uint8), (28, 28)), cv2.COLOR_RGB2GRAY) for img in x_test])
    # Reshape to (28, 28, 1)
x_train_gray = x_train_gray.reshape(-1, 28, 28, 1)
x_test_gray = x_test_gray.reshape(-1, 28, 28, 1)
model = Sequential()
model.add(Conv2D(75, (3, 3), strides=1, padding="same", activation="relu", input_shape=(28, 28, 1)))
model.add(GatchNormalization())
model.add(MaxPool2D((2, 2), strides=2, padding="same"))
model.add(Conv2D(50, (3, 3), strides=1, padding="same", activation="relu"))
model.add(Dropout(0, 2))
model.add(MaxPool2D((2, 2), strides=2, padding="same"))
model.add(MaxPool2D((2, 2), strides=2, padding="same"))
model.add(Conv2D(25, (3, 3), strides=1, padding="same", activation="relu"))
model.add(BatchNormalization())
  model.add(MaxPoolzD((2, 2), strides=2, padding="same"))
model.add(Flatten())
model.add(Dense(units=512, activation="relu"))
model.add(Dropout(0.3))
```

Call initializer instance with the dtype argument instead of passing it to the constructor	
Train on 44721 samples, validate on 14908 samples	
Epoch 1/20	
44721/44721 [====================================	
Epoch 2/20	
44721/44721 [====================================	
Epoch 3/20	
44721/44721 [====================================	
Epoch 4/20	
44721/44721 [====================================	
Epoch 5/20	
44721/44721 [====================================	
Epoch 6/20	
44721/44721 [====================================	
Epoch 7/20	
4472//44721 [============] - 370s 8ms/sample - loss: 1.7925 - acc: 0.1665 - val_loss: 2.1182 - val_acc: 0.1637	
Epoch 8/20	

### 테스트

```
from tensorflow.keras.preprocessing.image import load_img, img_to_array
# Assuming you have the test image in 'myeolchi001.jpeg'
test_image_file = "jeongaengee001.jpeg"
img = load_img(test_image_file, target_size=(28, 28))
img_array = img_to_array(img)
# Convert the image to grayscale
gray_img = cv2.cvtColor(img_array.astype(np.uint8), cv2.CoLOR_RGB2GRAY)
# Reshape to (28, 28, 1)
test_image = gray_img.reshape(1, 28, 28, 1) / 255.0
prediction=model.predict(test_image)
for i in prediction
   pre_ans=i.argmax()
    pre_ans_str=''
    if pre_ans==0: pre_ans_str="고등어'
    elif pre_ans==1: pre_ans_str="방어"
    elif pre_ans==2: pre_ans_str="삼치"
    elif pre_ans==3: pre_ans_str="멸치"
   elif pre_ans==4: pre_ans_str="전갱이"
    else: pre_ans_str="갈치"
print(pre_ans_str)
```

동물계 척삭동물문 조기어강 극기상목 전갱이과 전갱이속 전갱이 서식지 : 연안의 중층과 저층에 서식 천적 : 고등어 등의 어류 먹이 : 젓새우, 소형새우 등 대형 플랑크톤과 작은 어류, 오징어류 등 몸 길이가 40cm 가량이며, 등은 암청색이고 배는 은백색이다



전갱이

과가 같은 동물은 잘 구분을 못하는 모습을 보인다.

이름 제공





# 해양 생물 정보

```
f mackerel():
fish_data = {
  "고등어": {
  "조롱씨: "대평양 고등어(참고등어), 망치고등어(접고등어), 대서양 고등어",
  "태평양 고등어(참고등어)": {
  "학명": "Scomber japonicus",
  "불리는 이름": {
  "아름 ": {
  "어른 개체": "고등어",
  "어린 개체": "고도리",
  },
}
                },
"영어권 국가": "태평양고등어(Pacific mackerel), 일본고등어(Japanese mackerel), 처브고등어(chub mackerel)",
            "과": "고등어과",
"속": "고등어속",
"종": "참고등어",
"몽길이": "40cm 이상",
           "색": {
 "등": "녹색+검은색 물결무늬",
 "배": "은백색",
             ;,
"서식지": "수온이 10~20도인 맑은 온난해(바다의 표층과 중층에 서식){봄과 여름에는 얕은 곳, 가을에는 깊은 곳으로 이동}",
          },
"어린 개체": "동물성 플랑크톤",
```

years = [1970, 1975, 1980, 1980, 1990, 1990, 2000, 2010, 2015, 2020]
godeungeo = [38256, 70123, 62690, 68479, 96297, 200442, 145908, 135596, 99534, 140623, 115260] # 고등어 잡이랑
galchi = [69082, 120078, 119980, 127606, 103736, 94596, 81050, 60086, 59242, 41049, 54481] # 갈치 잡이랑
samchi = [25036, 25956, 12395, 4393, 5301, 6544, 19883, 4319, 2564, 574, 757] # 삼치 잡이랑
myeolchi = [54047, 175451, 169657, 143512, 130192, 230697, 201192, 249001, 249636, 211574, 210943] # 별치 잡이랑
bangeo = [72142, 37238, 48490, 42879, 74172, 200897, 228509, 189126, 159130, 155743, 87024] # 방어 잡이랑 jeongaengi = [883, 6563, 565, 15702, 16586, 12269, 19510, 42608, 19305, 43362, 21294] # 전쟁이 잡이량 plt.plot(years, samchi, marker='o', label='Samchi' plt.plot(years, myeolchi, marker='o', label='Myeolchi')
plt.plot(years, bangeo, marker='o', label='Bangeo') plt.plot(years, jeongaengi, marker='o', label='Jeo plt.text(years[i], godeungeo[i], f'Godeungeo: {godeungeo[i]}', fontsize=10, ha='center', va='bottom') pit.text(years[i], galchi[i], f'Galchi: {galchi[i]}', fontsize=10, ha='center', va='bottom')
pit.text(years[i], samchi[i], f'Samchi: {samchi[i]}', fontsize=10, ha='center', va='bottom')
pit.text(years[i], myeolchi[i], f'Myeolchi: {myeolchi[i]}', fontsize=10, ha='center', va='bottom') pit.text(years[i], bangeo[i], f'Bangeo: {bangeo[i]}', fontsize=10, ha='center', va='bottom')
pit.text(years[i], Jeongaengi[i], f'Jeongaengi: {Jeongaengi[i]}', fontsize=10, ha='center', va='bottom') plt.title("Fish Species-wise Annual Catch Trends") plt.ylabel("Catch Amount")

영 발풍 위기 중할 때 "바다 거북":(
"바다 거북은 7종 모두가 멸종위기 종으로 선정된 만큼 위기에 처한 동물이다.",
"바다 가북은 7종 모두가 멸종위기 종으로 선정된 만큼 위기에 처한 동물이다.",
"플라스틱 쓰레기로 인한 해양오렴, 산란장이 위치한 연안의 파괴, 무분별한 남희, 바다의 온도 상승이 1억 5천만년동인 "붉은바다거북, 매부리바다거북, 푸른바다거북, 장수거북 등이 한국에서 볼 수 있는 바다 거북이다.",
"기후위기에 무감각했던 일상을 잠시나마 돌아보는 건 어떨까?" ),
"바다 물법":{
"백형도를 비롯해 가로림만과 전국 동해·서해·남해 일원에 분포하는 바다 물법들이 생명을 위협받고 있다.",
"산업개발로 인한 환경오염으로 희생되고, 관광지 개발과 어민들의 어류 남쪽으로 인한 먹이 부족 등으로 늘 위험에 노출
"기후위기에 무감각했던 일상을 잠시나마 돌아보는 건 어떨까?" "수 천만년 이상 바다를 지배해 온 고래의 목숨 또한 위협 받고 있다.", "248종의 종들 중 20개의 종이 멸종 위기 종으로 선정되었고 전세계에 148만 마리의 고래만이 남아있다.", "커다렸고 신비한 존재인 고래의 생명이 해당 석유 시추 및 파괴적인 어덤 등으로 인해 꺼져가고 있다.", "기후위기에 무감각했던 일상을 잡시나마 돌아보는 건 어떨까?" 이 '',
"바다의 말이라 불리는 해마, 50종의 해마중 10종이 한국에서 발견된다.",
"바다의 말이라 불리는 해마, 50종의 해마중 10종이 한국에서 발견된다.",
"인간들의 무분별한 남희과 해양환경의 오염에 의한 해마의 서식지 파괴에 의해 해마 또한 그 목숨을 위험받고 있다.",
"바다의 포세이돈 해마의 목숨을 지켜주자",
"기후위기에 무감각했던 일상을 잠시나마 돌아보는 건 어떨까?"

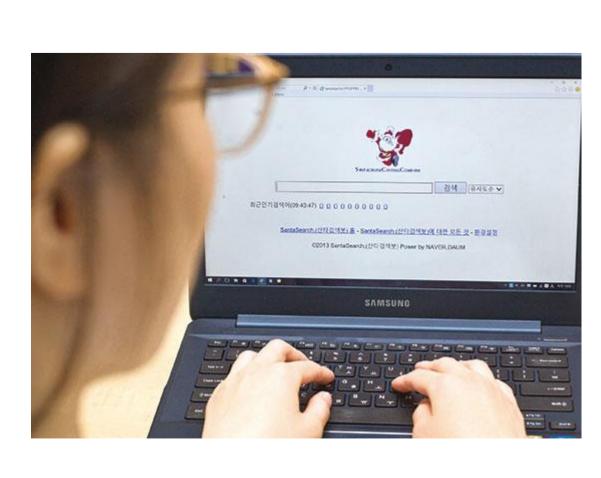
#### 정보 제공

```
if __name__ == "_main_":
fish_name = input("물고기 이름을 입력하세요(고등어, 갈치, 삼치, 멸치, 방어, 전쟁이의 정보가 준비되어 있습니다): ")
fish_info = get_fish_info(fish_name)
                print(f"{key}: {value}")
```

```
테스트
물고기 이름을 입력하세요(고등어, 갈치, 삼치, 멸치, 방어, 전갱이의 정보가 준비되어 있습니다 해양 멸종 위기 동물의 정보가 궁금하다면 해양 멸종 위기 동물이라 검색해 주세요): 삼치
학명: Trichiurus lepturus
불리는 이름:
한국: 삼치
서양: Spanish mackerels or seerfishes
계: 동물계
강: 조기어강
목: 고등어목
속: 삼치속
종: 삼치
무게: 5kg 전후
생김새: 고등어 보다 크고 날렵하게 생김
서식지: 수심 50~150m의 표해수대에 서식
산란기: 4~6월
분포: 서해와 남해, 동해를 비록한 일본, 하와이, 호주, 연해주 근해에 서식
어른 개체: 멸치, 까나리, 정어리, 전갱이
어린 개체: 갑각류, 어류의 치어
수명: 7년
성장 속도: 생후 1년후 성장이 거의 완료됨 이후 2년안에 모든 성장이 완료됨
천적: 상어
철(맛있는 시기): 10~2월
영양: 비타민B와 D가 다량 함유됨 칼슘도 풍부함
```

사진만 찍어서 올려도 해양 생물의 정보 제공 => 빠른 정보 제공





이름만 넣어도 해양 생물의 정보 제공 => 빠른 정보 제공