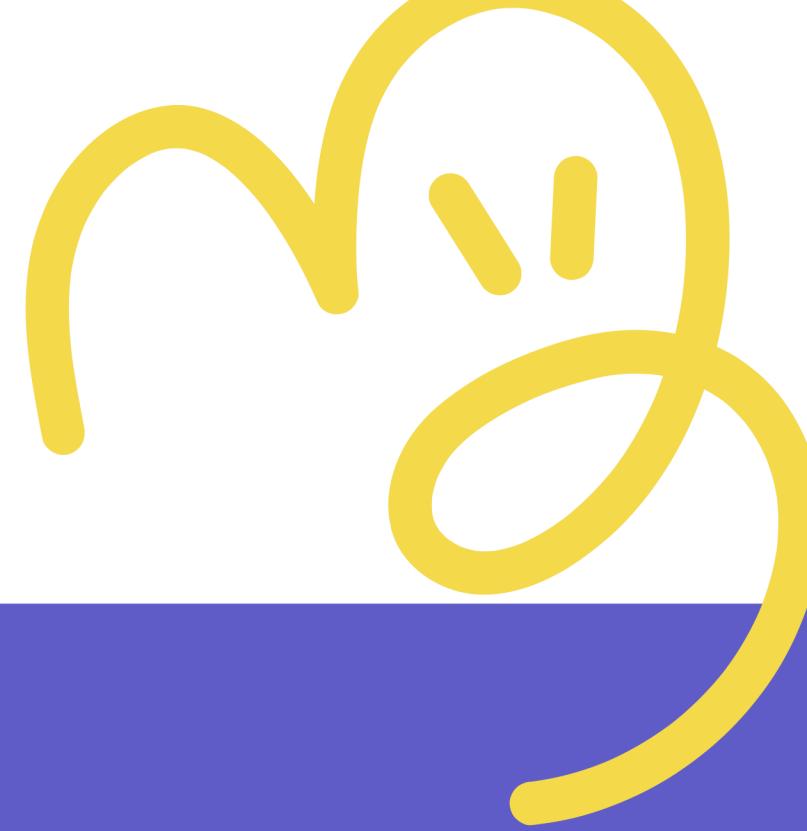




whale labs



Content



Project Overview	<u>03</u>	모델 학습	<u>08</u>
구성원 역할 분담	<u>04</u>	학습 결과	<u>09</u>
개발 환경	<u>05</u>	시스템 구현	<u>10</u>
시스템 구성도	<u>06</u>	Lessons learn	<u>11</u>
이미지 처리 프로세스	<u>07</u>		

Project Overview

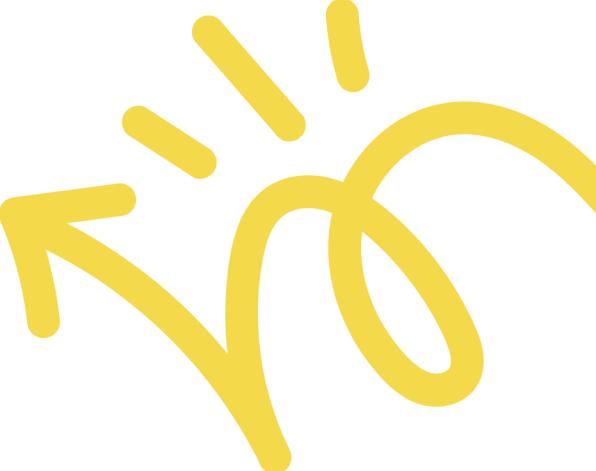
- introduction
- 주제 선정 및 배경



☒ [목차 페이지로 돌아가기](#)



introduction



유성호

- 데이터 수집
- 데이터 정제 가공



하재민

- 데이터 수집
- 데이터 정제 가공



김태엽

- Flask 서버 구축



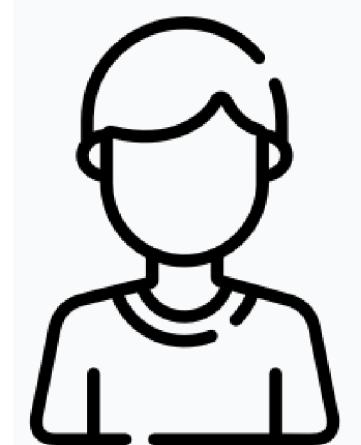
김석현

- 딥러닝 Model 구축 / 훈련



설성칠

- 결과 서비스 구현



모지원

- 딥러닝 Model 구축 / 훈련
- Chat 기능 구현



◀ [목차 페이지로 돌아가기](#)

주제 선정 및 배경



포토홈

흔획된 맹크고래 확인하는 해경

이종건 기자
[기자 페이지](#)

▲ □ ⌂ ⌂

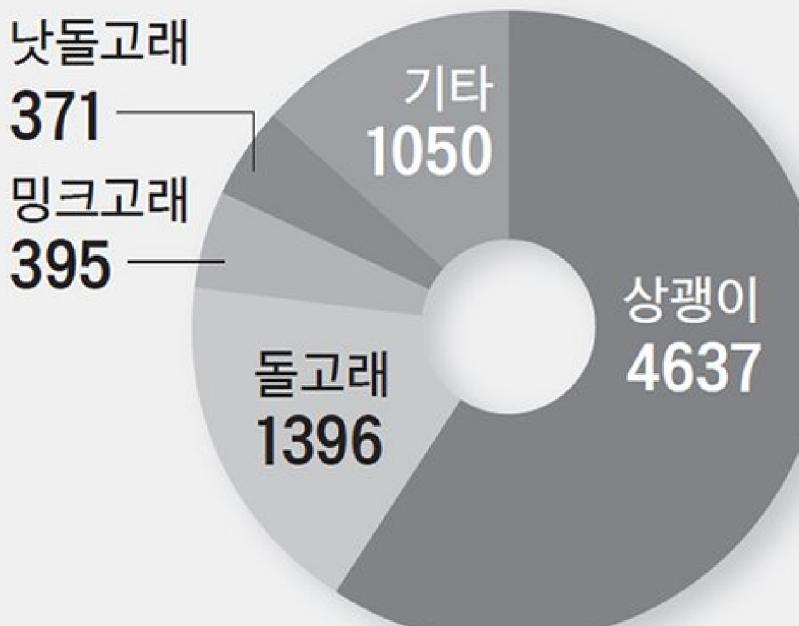
(동해=연합뉴스) 1일 오전 강원 동해시 목호항 북동방 1해리 해상에서 흔획된 맹크고래를 해경 직원이 확인하고 있다.

2023.5.1

[동해해경 제공.재판매 및 DB 금지]

momo@yna.co.kr

어떤 고래가 흔획되나 단위: 마리



※ 2015~2019년, 자료: 해양경찰청

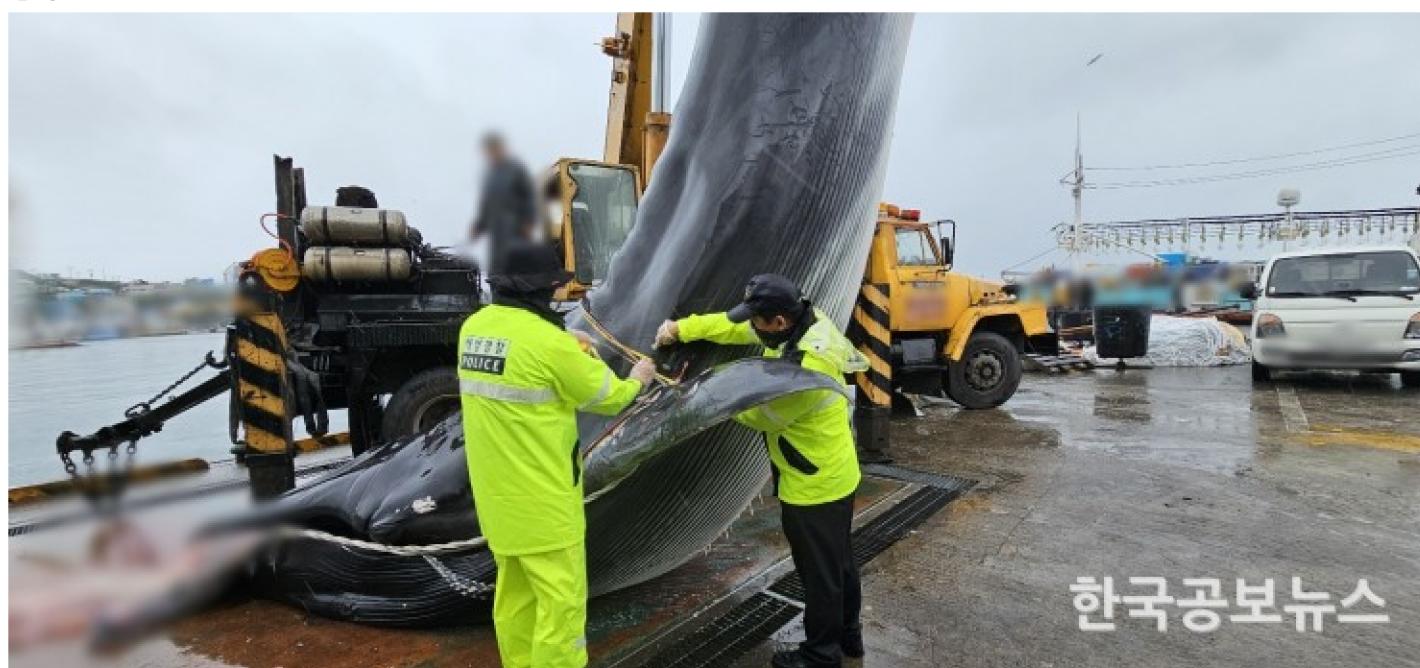
참여중심
뉴스통신
한국공보뉴스
www.kpnnews.org

포항해경, 보호종 '참고래' 흔획 신고접수

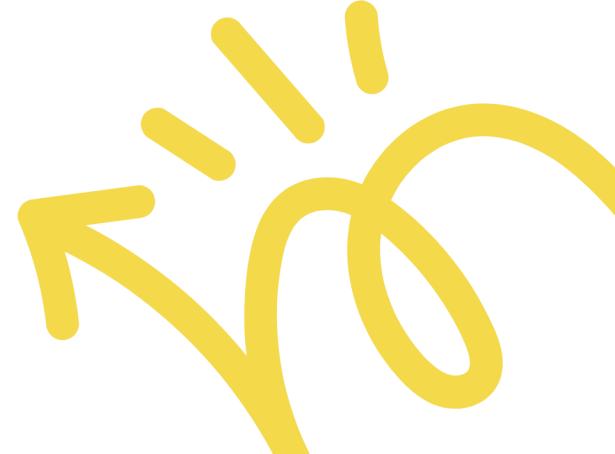
입력 : 2023-08-18 17:12:17

[한국공보뉴스/포항본부] 박동수 기자 = - 해양보호생물로 지정된 고래류는 위판 불가, 발견 시 해경에 신고

당부 -



한국공보뉴스



주제 선정 및 배경



• 상괭이(쇠돌고래과)



• 큰돌고래속(이빨고래소목 참돌고래과)



Development environment



◀ [목차 페이지로 돌아가기](#)



Development environment



◀ [목차 페이지로 돌아가기](#)



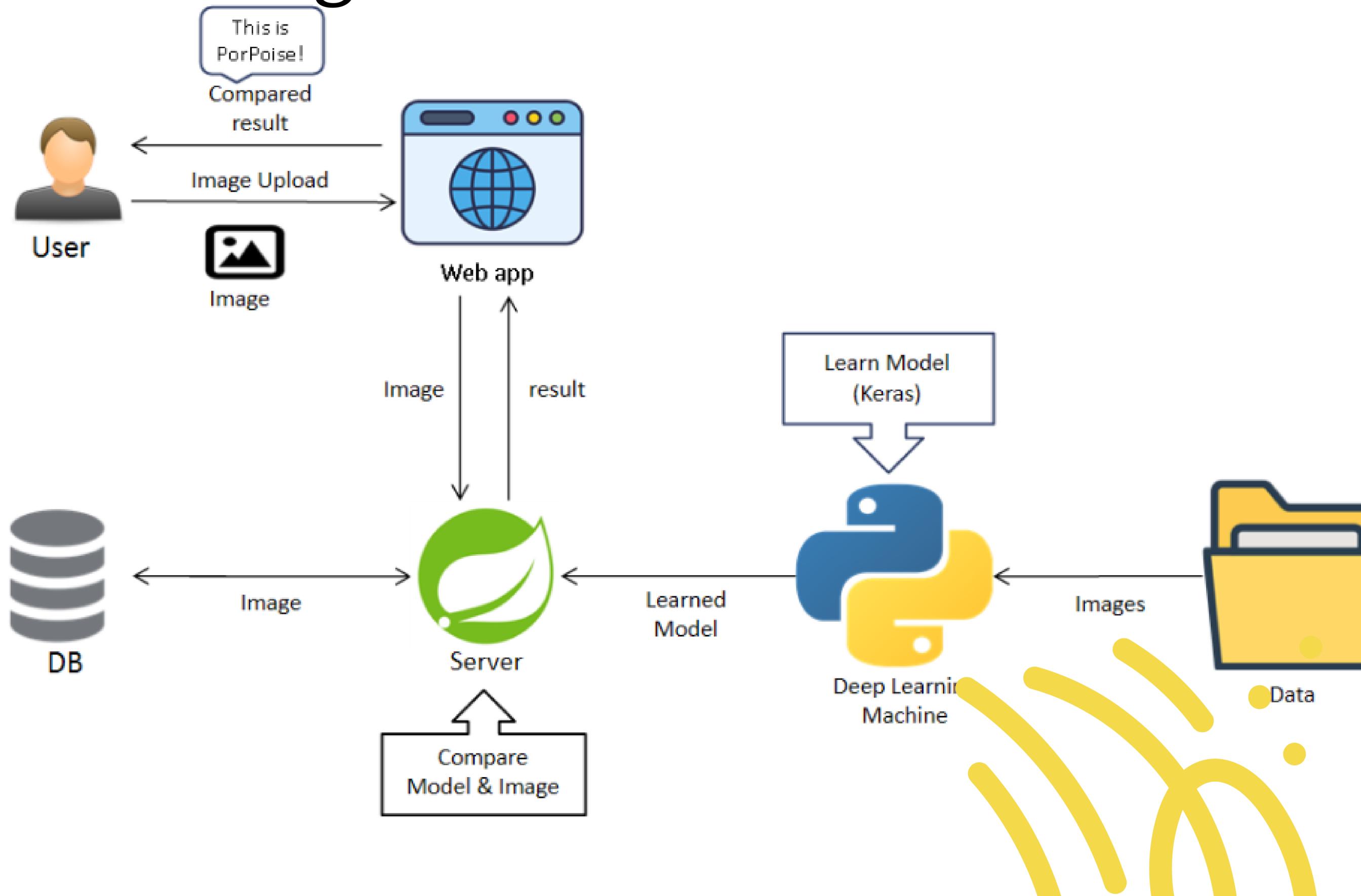
Project Performance and Results

- System Diagram
- Process

 [목차 페이지로 돌아가기](#)



System Diagram



Process

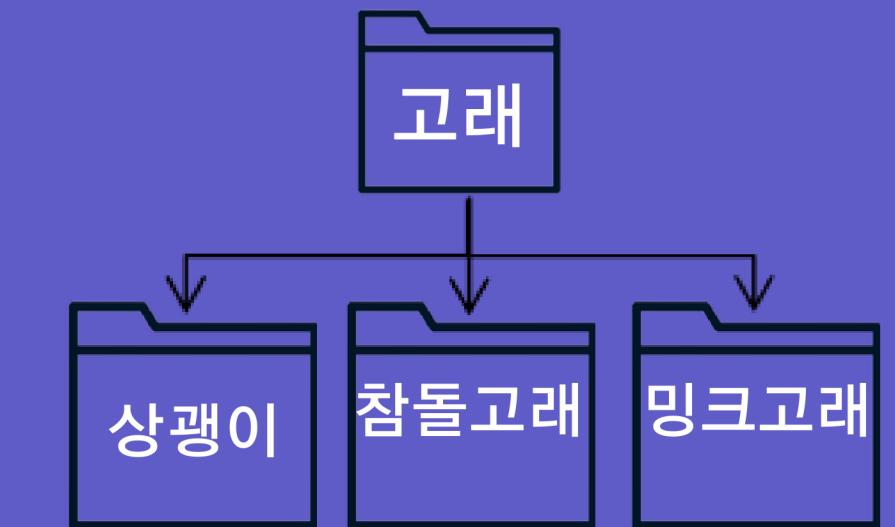
1. Crawling Image



2. Image Editing



3. DataSet



Crawling Image

```
# 검색쿼리
searchKey = input('검색할 키워드 입력 :')

# 폴더 생성
def createFolder(dir):
    try:
        if not os.path.exists(dir):
            os.makedirs(dir)
    except OSError:
        print('Error')

createFolder(f'train_dataset/{searchKey}')

driver = webdriver.Chrome()
driver.get('https://www.google.co.kr/imghp')

# 쿼리 검색 및 검색 버튼 클릭
elem = driver.find_element('name', 'q')
elem.send_keys(searchKey)
elem.send_keys(Keys.RETURN)

# 이미지 스크롤링
while True:
    driver.execute_script('window.scrollTo(0, document.body.scrollHeight);') # 브라우저 끝까지 스크롤
    time.sleep(1) # 쉬어주기
    try:
        button = driver.find_element(By.XPATH, '//*[@id="islmp"]/div/div/div/div[1]/div[2]/div[2]/input')
        button.click() # 스크롤을 내리다보면 '결과 더보기'가 있는 경우 버튼 클릭
        time.sleep(1)
    except:
        pass
    if driver.find_element(By.CLASS_NAME, 'OuJzKb.Yu2Dnd').text == '더 이상 표시할 콘텐츠가 없습니다.': # class 이름
        break

# 이미지 수집 및 저장
images = driver.find_elements(By.CSS_SELECTOR, ".rg_i.Q4LuWd") # 각 이미지들의 class
count = 1
for image in images:
    try:
        image.click()
        time.sleep(1)
        imgUrl = driver.find_element(By.XPATH,
            '//*[@id="Sva75c"]/div[2]/div[2]/div[2]/c-wiz/div/div/div/div[3]/div[1]/a/img[1]').get_attribute("src")
        imgUrl = imgUrl.replace('https', 'http') # https로 요청할 경우 보안 문제로 SSL에러가 남
        opener = urllib.request.build_opener()
        ...
```

```
# 초기 카운트 설정
count = 1

# 이미지를 저장할 디렉토리 경로 설정
save_directory = "C:\\\\Users\\\\it\\\\Desktop\\\\Python Work\\\\scraping\\\\train_dataset\\\\youtube Common Dolphin"

# 지정된 디렉토리가 없다면 생성
os.makedirs(save_directory, exist_ok=True)

while True:
    # 유튜브 화면의 좌표 설정 (예시 값, 실제 화면 크기와 위치에 맞게 수정)
    youtube_area = (223, 125, 1643, 918) # 좌상단 (x1, y1) 및 우하단 (x2, y2) 좌표

    # 유튜브 화면 영역만을 캡쳐
    screenshot = ImageGrab.grab(bbox=youtube_area)

    # 파일명 설정 및 카운트 증가
    file_name = f"screenshot_{count}.jpg" # JPEG 형식으로 저장
    file_path = os.path.join(save_directory, file_name)
    count += 1

    # 캡처한 이미지를 JPEG 형식으로 파일로 저장
    screenshot.save(file_path, format='JPEG', quality=95) # quality는 이미지 품질 조절 (1-95 범위)

    # 3초 대기
    time.sleep(0.5)
```



Image Editing

Raw Data

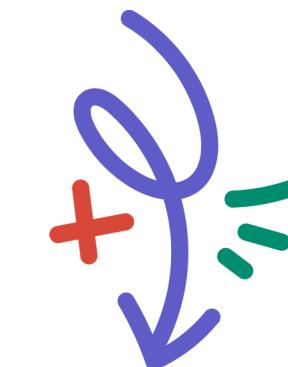
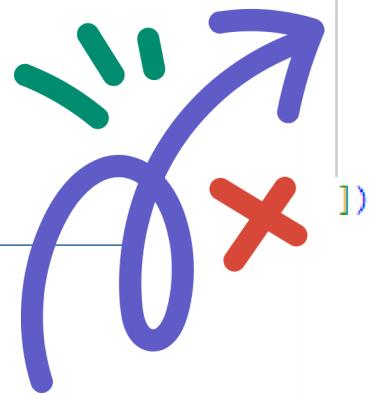
```
# 데이터셋 루트 폴더  
dataset_root = 'C:/Users/User/data/result/'  
  
# 모든 클래스 폴더 가져오기  
class_folders = [f for f in os.listdir(dataset_root) if os.path.isdir(os.path.join(dataset_root, f))]  
  
# 각 클래스 폴더 내의 이미지 수 계산  
total_images = 0  
for class_folder in class_folders:  
    class_path = os.path.join(dataset_root, class_folder)  
    images_in_class = [f for f in os.listdir(class_path) if f.endswith('.jpg') or f.endswith('.png')]  
    total_images += len(images_in_class)  
  
# 결과 출력  
print(f'클래스 폴더 수: {len(class_folders)}')  
print(f'총 이미지 수: {total_images}')
```

클래스 폴더 수: 3
총 이미지 수: 6133

상괭이 - 2,000 장
밍크 - 2000 장
참돌 - 2000장

Image Rotate / Horizontal / ColorJitter

```
# 이미지의 사이즈를 변환 후 텐서로 변환  
transforms = transforms.Compose([  
    transforms.Resize(64, 64),  
    transforms.RandomHorizontalFlip(),  
    transforms.RandomRotation(10),  
    transforms.ColorJitter(brightness=0.5, contrast=0.5, saturation=0.5, hue=0.5),  
    #transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)),  
    transforms.ToTensor(),  
])
```



Visualize

```
class_names = image_dataset.classes  
# 학습 결과 시각화  
train_losses = [] # 학습 손실값을 기록할 리스트  
train_accuracies = [] # 학습 정확도를 기록할 리스트  
test_losses = []  
test_accuracies = []  
save_interval = 1  
  
# 에포크마다 모델 저장을 위한 경로와 파일명 지정  
model_save_path = 'C:/Users/User/Desktop/model_result/models' # 저장할 디렉토리 경로  
model_name = 'model' # 저장할 모델 파일명 (에포크 번호가 자동으로 붙음)  
  
# 저장할 폴더가 없다면 생성  
os.makedirs(model_save_path, exist_ok=True)
```



Dataset



Run Epoch

```
#학습 수행(epoch = 50)
num_epochs = 20

for epoch in range(num_epochs):
    model.train()
    running_loss = 0.0
    correct = 0
    accuracy = 0
    total = 0
    test_total = 0
    test_loss = 0

    for data in train_loader:
        inputs, labels = data
        inputs, labels = inputs.to(device), labels.to(device)

        optimizer.zero_grad()
        outputs = model(inputs)
        loss = criterion(outputs, labels)
        loss.backward()
        optimizer.step()

        running_loss += loss.item()

        _, predicted = torch.max(outputs.data, 1)
        total += labels.size(0)
        correct += (predicted == labels).sum().item()
```

Cross-Entropy-Loss

```
#다중클래스 분류에서 사용하는 손실함수
criterion = nn.CrossEntropyLoss()
# 알고리즘(adam)
optimizer = optim.Adam(model.parameters(), lr=0.0001)

for param_tensor in model.state_dict():
    print(param_tensor, "\t", model.state_dict()[param_tensor].size())
```

CNN model

```
class CNNModel(nn.Module) :
    # 모델의 레이어
    def __init__(self, num_classes) :
        super(CNNModel, self).__init__()
        #4개의 합성곱, 폴링 레이어(pool), 완전연결레이어(fc1), 과적합 방지(dropout) 추가
        self.conv1 = nn.Conv2d(3, 32, kernel_size=3, stride=1, padding=1)
        self.bn1 = nn.BatchNorm2d(12)
        self.pool1 = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3, stride=1, padding=1)
        self.bn2 = nn.BatchNorm2d(48)
        self.pool2 = nn.MaxPool2d(2, 2)
        self.dropout1 = nn.Dropout(0.2)
        self.conv4 = nn.Conv2d(64, 64, kernel_size=3, stride=1, padding=1)
        self.bn4 = nn.BatchNorm2d(64)
        self.pool3 = nn.MaxPool2d(kernel_size=2, stride=2, padding=0)
        self.dropout2 = nn.Dropout(0.2)
        self.conv5 = nn.Conv2d(64, 128, kernel_size=3, stride=1, padding=0)
        self.bn5 = nn.BatchNorm2d(128)
        self.pool4 = nn.MaxPool2d(2, 2)
        self.fc1 = nn.Linear(64 * 64 * 4, num_classes) #이미지 사이즈가 128x128로 변경됨

    # 입력데이터의 연산 수행 후 출력
    def forward(self, input):
        output = self.pool3(F.relu(self.conv1(input)))
        #output = self.pool1(output)
        output = self.pool3(F.relu(self.conv2(output)))
        #output = self.pool2(output)
        output = F.relu(self.dropout1(output))
        #output = self.pool2(output)
        #output = F.relu(self.bn4(self.conv4(output)))
        #output = self.pool3(output)
        #output = F.relu(self.dropout2(output))
        #output = F.relu(self.bn5(self.conv5(output)))
        #output = self.pool4(output)

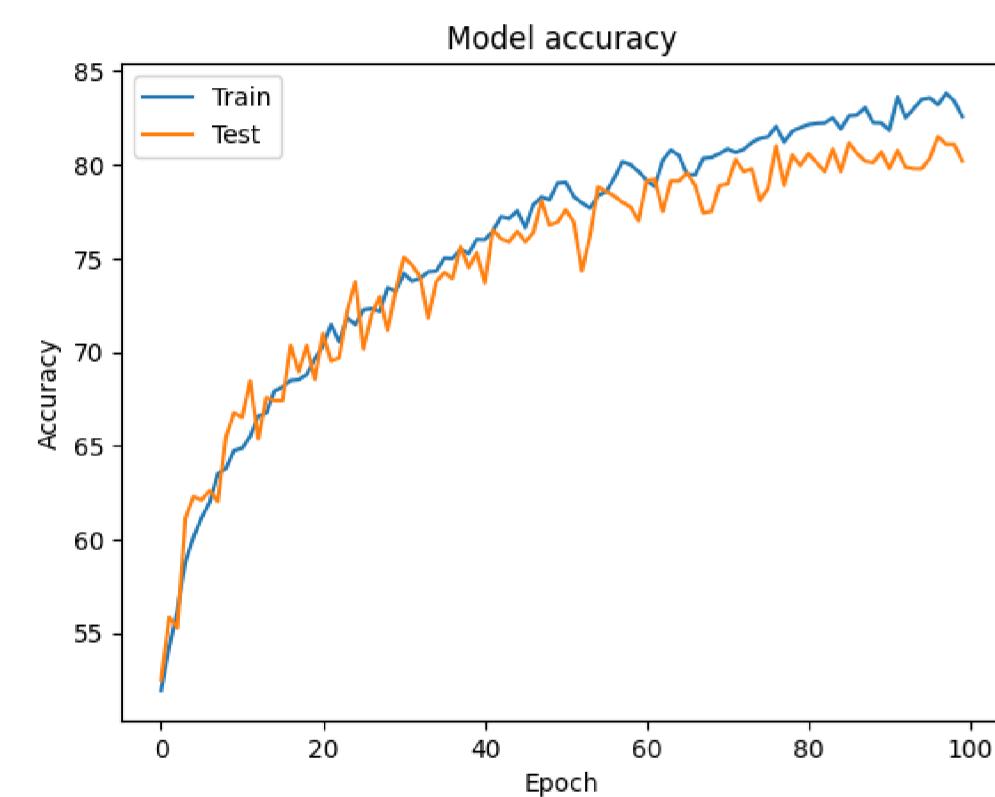
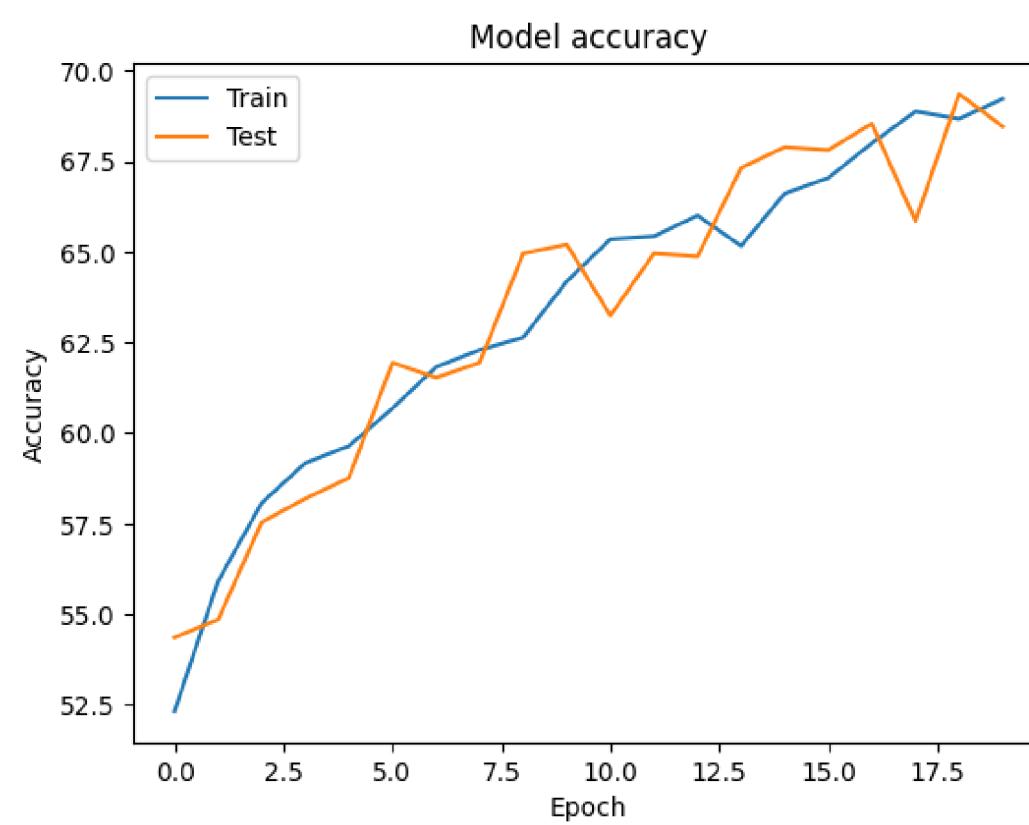
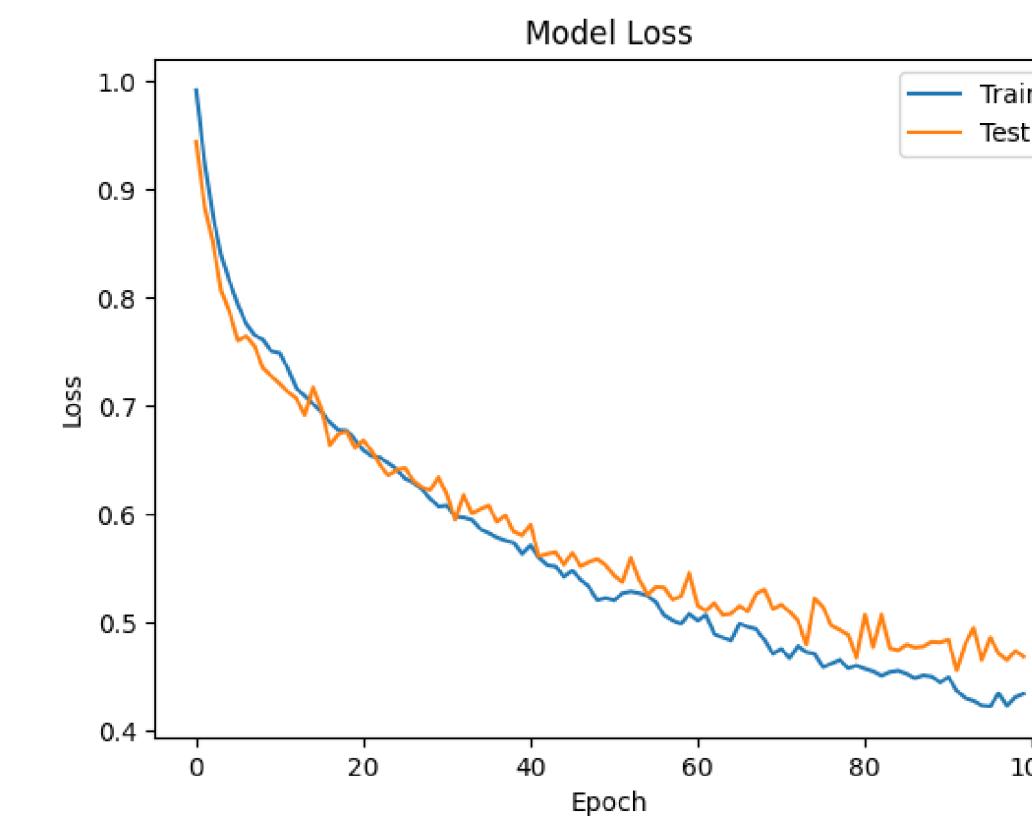
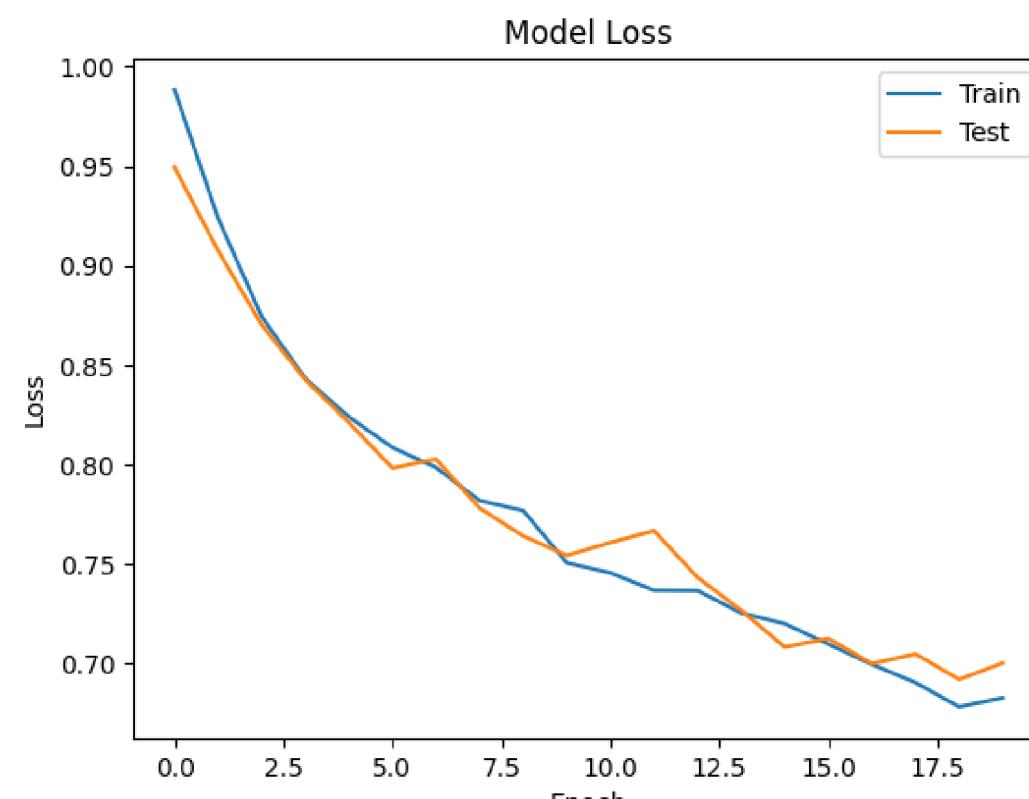
        # Global Average Pooling
        #output = F.adaptive_avg_pool2d(output, (1, 1))

        # 변경된 부분: 이전에는 128*128*128이었지만, 이제는 1280이 됩니다.
        output = output.view(-1, 64 * 64 * 4)
        output = self.fc1(output)

    return output
```



Result



Epoch 20

Epoch 100



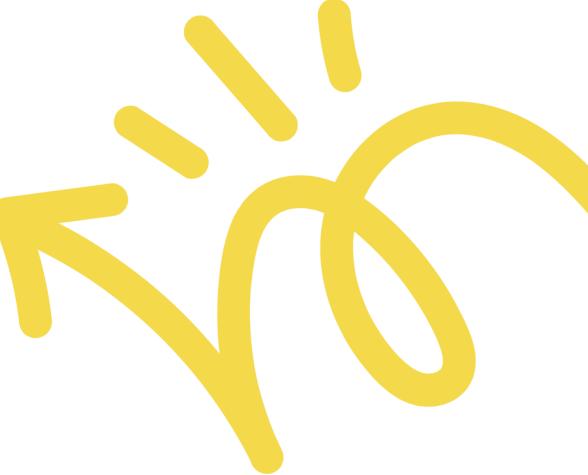
App Operation

- System Diagram
- Process

 [목차 페이지로 돌아가기](#)



Log in / Sign up



Log in



Whale Labs
for Android

Enter ID

Enter Password

New to Whale Labs? [Create an account](#)

Sign in



Whale Labs
for Android

Enter ID

Enter Name

Enter Cell Phone

Enter Password

Enter Password 2

Enter E-mail



Home / Result

Home

Whale Labs

Hello XXX!

[Home](#) [W.Chats](#)

Whale  select image

Place 

Width 7 M Weight 7 t

Date 2023/11/06

ETC

[Send](#)

image result

Whale Labs

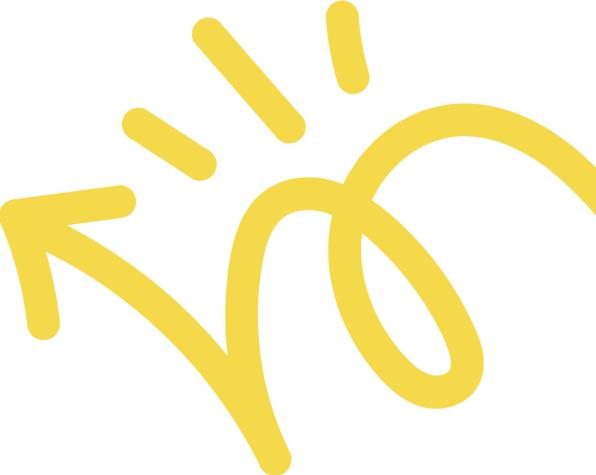
Hello XXX!

[Home](#) [W.Chats](#)



Whale Whale name

[more information](#)



Home / Result



massenger main

Whale Labs Hello XXX!

Home W.Chats

W.Chats

Search

Stells Stefword 5:30 PM
You sent a sticker

Samuel Joyce 4:00 PM
Samuel sent a sticker

Jenni Miranda 10:00 AM
I like this version 😊😊

Elva Barker 22 Aug
Elva sent an attachment

massenger chat list

Whale Labs Hello XXX!

Home W.Chats

< Jenni Miranda

Hi Anni, What's Up?

Oh, hello! All perfect 😊👍

Hi Anni, What's Up?

Oh, hello! All perfect 😊👍

Start Typing... +

This image displays two side-by-side screenshots of a messaging application interface. The left screenshot shows the 'Home' tab of the main screen, featuring a list of recent messages from users Stells Stefword, Samuel Joyce, Jenni Miranda, and Elva Barker. The right screenshot shows the 'W.Chats' tab of the main screen, which has been expanded into a full chat window for the conversation with Jenni Miranda. This chat window includes a header with the recipient's name and a message, followed by a back-and-forth exchange of messages and emojis. At the bottom of the chat window is a text input field with a placeholder and a '+' button for attachments.



Lessons Learn



[목차 페이지로 돌아가기](#)



Lessons Learn

