

# 카피 (Categorized Fish)

「어민들을 위한 병어 자동 분류 프로그램」

분 반 : 01분반  
과목명 : 산학SW프로젝트  
교수명 : 나인섭  
이 름 : 임대인  
학 번 : 20154300

# 목 차

## 1. 프로그램 소개

주제 .....	3
개발 동기 및 목적 .....	3
기대효과 및 활용방안 .....	3
기존 유사 연구 .....	4

## 2. 개발 내용

개발 계획 .....	4
개발 환경 .....	4
개발 내용 .....	4
기본 설계 .....	4

## 3. 세부 내용

카피 시스템 구성도 .....	5
카피 데이터 수집 및 전처리 .....	5
카피 물고기 분류 모델 생성 .....	6
카피 분류 모델과 웹 사이트 연동 .....	6

## 4. 결과 및 고찰

결과 분석 및 고찰 .....	6
참고문헌 .....	7

## 1. 프로그램 소개

### 1-1) 주제

어민들을 위한 병어 자동 분류 프로그램

### 1-2) 개발 동기 및 목적

#### □ 개발동기

수산물에 대한 전염병 발생이 5년새 4배이상 증가하는 등 병어에 대한 피해가 증가하고 있다. 정부가 예산을 들여 예방이나 치료에 힘을 쓰고 있지만 뚜렷한 효과가 나타나지 않았는데 이런 문제에 도움이 될 만한 솔루션을 제공하고자 개발을 시작하게 되었다.

#### □ 개발목적

- 양식업을 하는 어민들의 병어로 인한 피해를 최소화
- 물고기의 병의 경우 바이러스로 전파되어 전염성이 있기 때문에 빠른 발견을 통해 전염을 예방

### 1-3) 기대효과 및 활용방안

#### □ 기대효과

- 사람이 많은 수의 물고기를 관찰하고 병어를 판단하는 것은 현실적으로 어려운데 이를 자동화하여 편의성 증대
- 양식업을 한다고 해서 물고기 병에 대해 잘 알 수 없는데 자동화 시스템을 통해 어떤 병인지 파악하고 정보를 제공하여 적절한 대처 가능

#### □ 활용방안

- 양식업을 하는 어민들의 어장에 설치하여 병어 분류 서비스
- 병어뿐 아니라 물고기 종도 분류를 해주어 낚시꾼과 같은 일반인들에게도 어종 분류 서비스를 제공

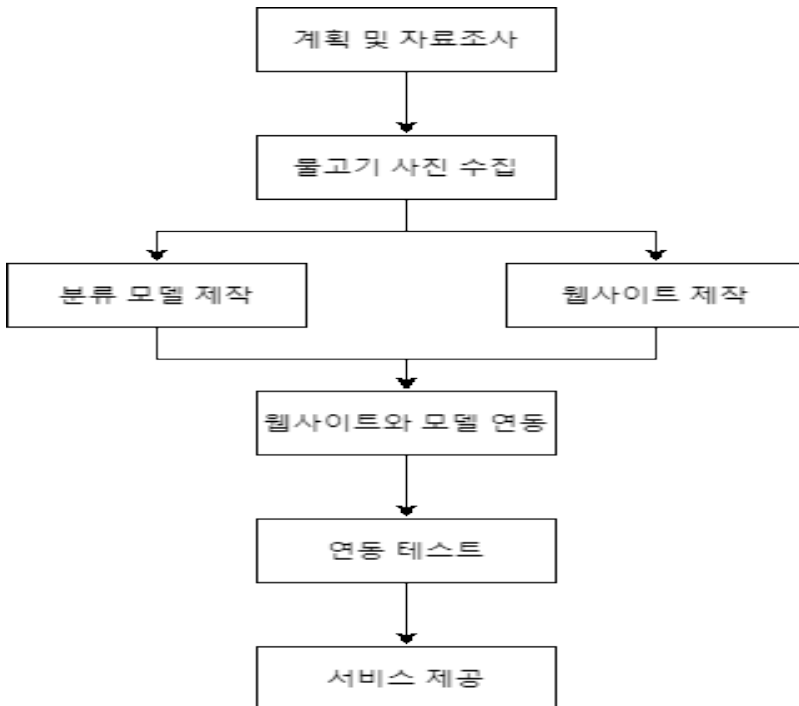
### 1-4) 기존 유사 프로그램



- 물고기의 크기에 따라 분류를 해주는 경사식 어류 선별기 (좌측 이미지)
- 양식 어류의 관리를 위한 수온 조절, 먹이 자동 도포 기술

## 2. 개발 내용

### 2-1) 개발 계획



### 2-2) 개발 환경

- OS : Microsoft Windows 10 Education (version: 1903, build: 18362)
- 웹 : HEROKU (웹 배포, version : 7.39.5), Django (웹 서버, version: 2.0.13)
- 주요 개발언어 : Python IDLE (version: 3.7)
- 소스편집기 : VisualStudio Code (version: 1.38.1)

### 2-3) 개발 내용

- 물고기별 대표 이미지를 사용하여 합성 이미지 종류별 30장 생성
- 구글 사의 오픈소스 딥러닝 모델인 Inception을 사용하여 모델을 생성
- 웹 사이트를 통해 사진을 업로드하여 물고기를 분류할 수 있게 웹 사이트 구현
- Keras API를 사용하여 사진마다 25장씩 이미지 부풀리기

### 2-4) 기본 설계

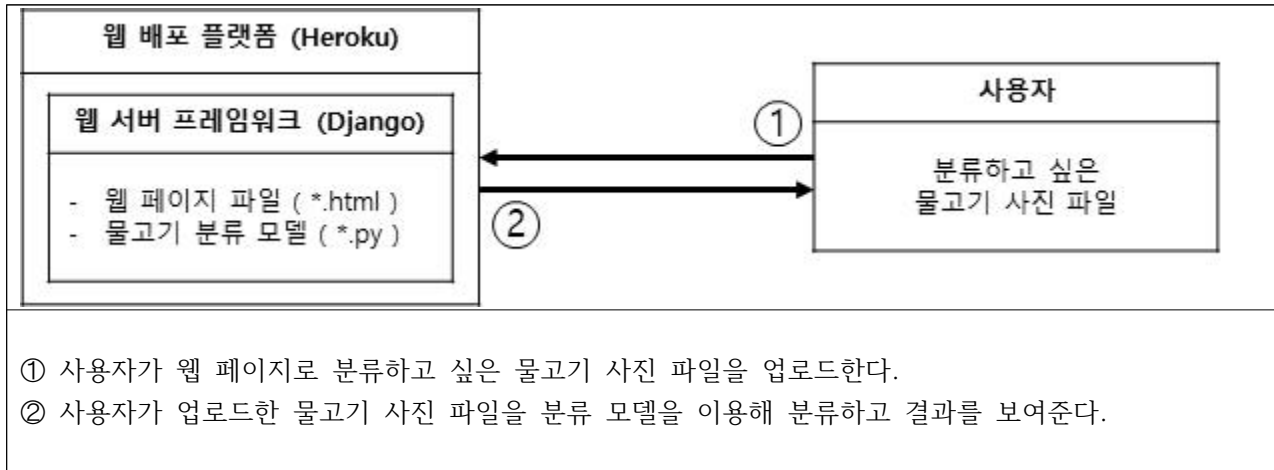
1. 데이터 수집 : 병어의 이미지 데이터는 절대적으로 부족, 그렇기 때문에 대표 병어 이미지를 지정한 뒤 해당 이미지의 상처 부분을 사용해 합성이미지를 물고기별 정상, 비정상 30장을 생성한다.
2. 데이터 전처리 : 텐서플로우의 Keras API를 사용해 이미지를 회전, 밀기, 뒤집기 등을 적절히 하여 이미지당 25장의 이미지를 부풀려 4500장((30\*3\*2)\*25)의 이미지를 학습데이터로 사용한다.
3. 모델 생성 및 웹사이트 구현 : 구글의 오픈소스 모델인 Inception v3 모델을 사용하여 4500장의 이미지를 학습시켜 모델을 만든다. 그리고 Django 프레임워크를 사용하여 웹페이지를 구현한다.
4. 모델과 웹사이트 연동 : 웹사이트에서 사진을 업로드하면 만들어둔 모델을 사용하여 해당 사진의 물고기가 병어인지 정상 물고기인지 판단하여 나타내도록 연동한다.
5. 최종 테스트 및 배포 : 각 어종별 테스트를 진행한 후 웹사이트를 누구나 사용가능하게 배포한다.

### 3. 세부 내용

(코드는 Github에 업로드 :

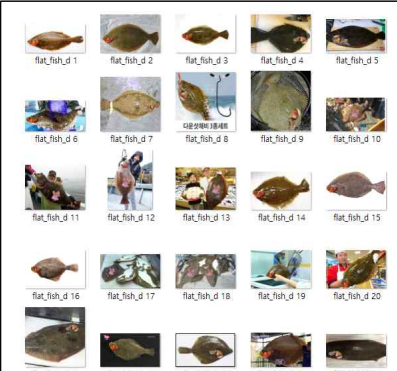
[https://github.com/dnfwlxl1/CategorizedFish\\_ChosunUniv\\_fallsemesterProject](https://github.com/dnfwlxl1/CategorizedFish_ChosunUniv_fallsemesterProject))

#### 3-1) 카피 시스템 흐름도




#### 3-2) 카피 데이터 수집 및 전처리

##### □ 수집 방법



- 대표 이미지에서 상처를 가져와 정상 물고기 사진에 합성
- 광어, 송어, 볼락에 대해 총 90장을 합성  
(정상 물고기 90장, 합성 물고기 90장, 총 180장)

##### □ 데이터 부풀리기



- 하나의 이미지를 왜곡, 뒤집기, 반전 등 처리를 하여 25장씩 이미지를 생성 (왼쪽처럼 하나의 사진이 여러 모습으로 부풀려짐)

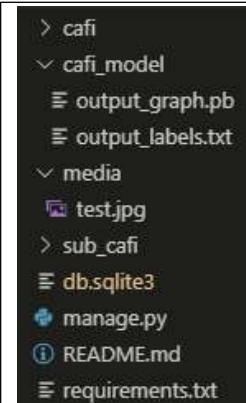
### 3-3) 카피 물고기 분류 모델 생성

(<https://gist.github.com/solaris33/>) - 구글의 Inception v3을 한글로 주석 처리하여 학습 예제를 제공

<pre>parser.add_argument(     '--train_batch_size',     type=int,     default=100,     help='How many images to train on at a time.' )</pre>	- 하나의 이미지를 한번에 얼마나 학습에 넘겨주는지를 설정하는 것으로 N = 100이 성능, 속도 면에서 가장 좋은 모습을 보여 100으로 설정
<pre>parser.add_argument(     '--how_many_training_steps',     type=int,     default=1000,     help='How many training steps to run before ending.' )</pre>	- 몇 단계에 거쳐 학습을 하는지 설정하는 것으로 1000번 이상으로 해보았지만 성능면에서 좋아지는 것이 없어 속도, 성능면에서 가장 좋았던 1000으로 설정

그 밖의 파라미터들은 설정값을 변경함에도 크게 의미가 없어 디폴트 값으로 두고 학습을 진행하였다. 진행 순서는 먼저 물고기의 사진들을 분류하고 싶은 것 별로 하나의 폴더에 담은 뒤에 실행을 하여야한다. 실행하면 각 폴더 명으로 라벨링을 한 뒤 이미지를 하나하나 학습한다. 그 뒤 미리 설정한 경로에 라벨링한 'output\_labels.txt'와 학습이 완료된 모델인 'output\_graph.pb' 파일이 생성된다.

### 3-4) 카피 분류 모델과 웹 페이지 연동

	- 사용자가 업로드한 이미지는 웹 서버의 동적 저장 폴더인 'media'에 분류 모델은 웹 서버의 정적 저장 폴더인 'static'에 저장하여 웹 페이지에서 사용자의 행동에 따라 분류를 하고 결과를 보여주게 구현하였다.
--	--

## 4. 결과 및 고찰

### 4-1) 결과 분석 및 고찰

4500장의 이미지로 만든 모델의 평균 정확도는 91%로 높은 수치를 기록하였다. 하지만 데이터 수집 과정에서 병어 이미지가 한, 두 장을 제외하고는 전혀 존재하지 않아 모두 합성으로 가상의 병어를 만들어 학습하고 테스트하였기 때문에 신뢰 성있는 결과라고 보기는 어렵다. 하지만 해당 모델에서 병어 분류 기준으로 한 '스쿠티카'병의 경우 물고기의 외부에 아가미 쪽이나 지느러미 쪽에 생긴다는 특징이 있으며 몸이 벗겨져 빨갭게 살점이 보이는 특징을 가지고 있기 때문에 상처의 모양만 추가적으로 수집하여 합성을 한다면 실제 병어들의 모습과 상당히 유사할 것이라고 생각한다. 프로젝트 시작단계에서 교수님께서 해당 연구를 하고있는 지인이 있고 해양생태관련 정부의 부서에서도 데이터를 가지고 있을 것이라고 하였으니 협력하여 개발을 진행한다면 해양 생태, 양식에 큰 도움이 될 것이라고 생각한다. 여기에 병어가 있던 환경의 데이터나 전염된 수와 같은 부가적인 데이터를 수집한다면 병의 전염성, 병을 유발하는 환경을 찾아내는 등 의학, 환경 개선에도 도움이 될 것이라고 생각한다.

프로젝트 진행간 계획한 대로 되지 않은 부분이 있어 아쉬웠는데 바로 웹 사이트의 배포이다. 기획했던 Heroku 플랫폼의 경우 무료 서비스는 일정 이상의 파일을 저장할 수 없는 제한이 있는데 제작한 분류 모델이 초과해버려 웹 배포를 진행하지 못하여 로컬 환경에서만 작동하였다. 아마존 S3, Cloudinary와 같은 차선책이 있다고 하니 수정을 통해 웹 배포까지 진행하고자 한다.

#### 4-2) 참고 문헌

(<http://solarisailab.com/archives/1422>) - Inception v3를 사용해 고양이 분류하기 예제

(<https://docs.djangoproject.com/ko/3.0/intro/tutorial01/>) - Django로 앱 시작하기 예제