



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ ГОЛОВНОЙ УЧЕБНО-ИССЛЕДОВАТЕЛЬСКИЙ И МЕТОДИЧЕСКИЙ ЦЕНТР
ПРОФЕССИОНАЛЬНОЙ РЕАБИЛИТАЦИИ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ
ЗДОРОВЬЯ _____

КАФЕДРА _____ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ _____

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

УЛУЧШЕНИЯ С ПОМОЩЬЮ СМЕШИВАНИЯ

ИЗОБРАЖЕНИЙ

Студент ИУ5И-31М
(Группа)

(Подпись, дата) Лю Бэйбэй
(И.О.Фамилия)

Руководитель

(Подпись, дата) Ю.Е. Гапанюк
(И.О.Фамилия)

2023 г.

« 04 » сентября 2023 г.

По теме _____

Лю Бэйбэй
(Фамилия, имя, отчество)

ИССЛЕДОВАТЕЛЬСКАЯ

График выполнения НИР: 25% к ____ нед., 50% к ____ нед., 75% к ____ нед., 100% к ____ нед.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

(Подпись, дата)

Ю.Е. Гапанюк
(И.О.Фамилия)

(Подпись, дата)

(И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Содержание

1.	Введение	4
2.	Смешивание изображений	5
	Основные принципы смешивания изображений	5
	Значение смешивания изображений в обработке подводных изображений	6
3.	Гибридный алгоритм смешивания для идентификации звездных рыб короны ..	6
	Простой перенос цвета по нормализации RGB	6
	Редактирование Пуассона	6
4.	Кодирование	7
5.	Результаты	15
6.	Заключение	16
7.	Список литературы	17

Введение

Покрывая 71% поверхности Земли, океаны являются фундаментальной частью глобальной системы жизнеобеспечения и содержат огромное количество ресурсов и энергии, что делает их "шестым континентом", эксплуатируемым человечеством. Использование технологий обработки изображений и компьютерного зрения для защиты моря имеет то преимущество, что они менее затратны и просты в использовании, чем сложные и дорогостоящие технологии.

Потрясающе красивый Большой барьерный риф Австралии - это крупнейший в мире коралловый риф, где обитает 1500 видов рыб, 400 видов кораллов, 130 видов акул, скатов и огромное количество других морских обитателей. К сожалению, риф находится под угрозой, отчасти из-за перенаселенности одной конкретной морской звезды - кораллоядной короны тернового венца (сокращенно COTS) [6,8]. Чтобы управлять определением состояния рифов и контролировать эффективность и масштаб вспышек COTS, с помощью подводных камер были собраны тысячи изображений коралловых рифов для точной идентификации, этой питающейся кораллами венценосной морской звезды.

Резкость подводного изображения оказывает значительное влияние на качество их обработки разными методами, включая методы глубокого обучения. Однако обнаружение объектов на изображениях, полученных под водой, часто осложняется полученной плохой четкостью [2]. До настоящего времени для решения этой проблемы было предложено множество алгоритмов улучшения подводных изображений [1-5,7]. В данной статье мы представляем инновационный подход к улучшению процесса идентификации звездных рыб короны в подводных изображениях коралловых рифов. Наш подход базируется на использовании алгоритмов смешивания изображений, представляя новаторские методы для повышения точности и эффективности этого процесса.

Несмотря на значительные достижения в этой области, существующие методы сталкиваются с вызовами, такими как сохранение цветовой гармонии и обеспечение естественного перехода между различными изображениями. В данной статье представлен обзор исследований и разработок в области аугментации с использованием смешивания изображений, а также предложены новые методы смешивания для преодоления текущих ограничений.

Смешивание изображений выделяется как важный элемент обработки визуальных данных, используемый для достижения естественности и визуальной привлекательности. Исторические методы, такие как передача цвета через нормализацию RGB, служат основой для современных подходов. Мотивацией данного исследования служит стремление к созданию более эффективных и универсальных методов смешивания изображений, способных к успешному применению в различных областях, от компьютерного зрения до создания искусственных сцен. Целью данной статьи является исследование существующих методов аугментации с использованием смешивания изображений, а также разработка и представление инновационных методов для повышения качества визуальных данных. Задачи включают в себя подробный анализ существующих алгоритмов, создание новых методов и их оценку в контексте практических применений.

Смешивание изображений

Смешивание изображений представляет собой ключевой процесс в обработке изображений, который находит широкое применение в различных областях, включая фотографию, компьютерное зрение и медицинскую обработку изображений. Этот раздел предоставляет обзор основных принципов и техник смешивания изображений, подчеркивая их значение в контексте обработки подводных изображений коралловых рифов для идентификации звездных рыб короны.

Основные принципы смешивания изображений

Смешивание изображений включает в себя комбинирование информации из двух или более изображений для создания нового изображения с определенными характеристиками. Основные методы смешивания включают арифметическое среднее, взвешенное среднее, максимальное и минимальное значения пикселей. Эти методы часто применяются для улучшения контраста, уменьшения шумов и объединения изображений с разными экспозициями.

Значение смешивания изображений в обработке подводных изображений

В контексте подводных изображений коралловых рифов, смешивание изображений играет важную роль в улучшении визуальных характеристик, таких как контрастность и цветовая передача. Это особенно важно для задачи идентификации звездных рыб короны, где точность определения объектов на изображении зависит от четкости и детализации.

Гибридный алгоритм смешивания для идентификации звездных рыб короны

В данной статье мы представляем инновационный гибридный алгоритм смешивания, объединяющий два ключевых этапа: простой перенос цвета по нормализации RGB и редактирование Пуассона. Этот уникальный подход разработан для повышения эффективности и точности идентификации звездных рыб короны в подводных изображениях коралловых рифов.

Простой перенос цвета по нормализации RGB

Простой перенос цвета направлен на согласование цветовых характеристик различных изображений путем нормализации значений RGB. Этот этап способствует выравниванию цветов и контраста, что положительно сказывается на визуальной четкости изображений. Нормализация RGB значений пикселей осуществляется в соответствии с разработанным методом, учитывая особенности подводных условий. Результаты этого этапа служат входными данными для последующих этапов гибридного алгоритма.

Редактирование Пуассона

Редактирование Пуассона включено в гибридный алгоритм для эффективного внедрения и выделения звездных рыб короны на подводных

изображениях. Этот метод призван улучшить контраст и выделить ключевые детали. Мы успешно интегрировали редактирование Пуассона в гибридный алгоритм, что привело к улучшению видимости и контрастности звездных рыб короны.

Кодирование

```
import numpy as np
import pandas as pd
import cv2

%matplotlib inline
from matplotlib import pyplot as plt

#blending algorithms

# 1. simple color transfer by rgb normalisation
#https://github.com/chia56028/Color-Transfer-between-Images/blob/master/color\_transfer.py

def norm_color_transfer(src, dst):

    def get_mean_and_std(x):
        x_mean, x_std = cv2.meanStdDev(x)
        x_mean = np.hstack(np.around(x_mean,2)).reshape(1,1,3)
        x_std = np.hstack(np.around(x_std,2)).reshape(1,1,3)
        return x_mean, x_std

    s = cv2.cvtColor(src,cv2.COLOR_BGR2LAB)
    t = cv2.cvtColor(dst,cv2.COLOR_BGR2LAB)
    s_mean, s_std = get_mean_and_std(s)
```

```

t_mean, t_std = get_mean_and_std(t)

m = (s-s_mean)*(t_std/s_std)+t_mean
m = np.round(m)
m = np.clip(m,0,255).astype(np.uint8)

m = cv2.cvtColor(m,cv2.COLOR_LAB2BGR)
return m

```

```

# 2. poisson editing
# https://github.com/PPPW/poisson-image-editing
import scipy.sparse
from scipy.sparse.linalg import spsolve

```

```

def laplacian_matrix(n, m):
    """Generate the Poisson matrix.
    Refer to:
    https://en.wikipedia.org/wiki/Discrete\_Poisson\_equation
    Note: it's the transpose of the wiki's matrix
    """
    mat_D = scipy.sparse.lil_matrix((m, m))
    mat_D.setdiag(-1, -1)
    mat_D.setdiag(4)
    mat_D.setdiag(-1, 1)

    mat_A = scipy.sparse.block_diag([mat_D] * n).tolil()

    mat_A.setdiag(-1, 1*m)

```



```
mat_A.setdiag(-1, -1*m)
```

```
return mat_A
```

```
def poisson_edit(source, target, mask, offset=(0,0)):
```

```
    """The poisson blending function.
```

```
    Refer to:
```

```
    Perez et. al., "Poisson Image Editing", 2003.
```

```
    """
```

```
    # Assume:
```

```
    # target is not smaller than source.
```

```
    # shape of mask is same as shape of target.
```

```
    y_max, x_max = target.shape[: -1]
```

```
    y_min, x_min = 0, 0
```

```
    x_range = x_max - x_min
```

```
    y_range = y_max - y_min
```

```
    M = np.float32([[1,0,offset[0]],[0,1,offset[1]]])
```

```
    source = cv2.warpAffine(source,M,(x_range,y_range))
```

```
    mask = mask[y_min:y_max, x_min:x_max]
```

```
    mask[mask != 0] = 1
```

```
    #mask = cv2.threshold(mask, 127, 1, cv2.THRESH_BINARY)
```

```
    mat_A = laplacian_matrix(y_range, x_range)
```

```
    # for  $\Delta g$ 
```

```
laplacian = mat_A.tocsc()
```

```
# set the region outside the mask to identity
```

```
for y in range(1, y_range - 1):
```

```
    for x in range(1, x_range - 1):
```

```
        if mask[y, x] == 0:
```

```
            k = x + y * x_range
```

```
            mat_A[k, k] = 1
```

```
            mat_A[k, k + 1] = 0
```

```
            mat_A[k, k - 1] = 0
```

```
            mat_A[k, k + x_range] = 0
```

```
            mat_A[k, k - x_range] = 0
```

```
# corners
```

```
# mask[0, 0]
```

```
# mask[0, y_range-1]
```

```
# mask[x_range-1, 0]
```

```
# mask[x_range-1, y_range-1]
```

```
mat_A = mat_A.tocsc()
```

```
mask_flat = mask.flatten()
```

```
for channel in range(source.shape[2]):
```

```
    source_flat = source[y_min:y_max, x_min:x_max, channel].flatten()
```

```
    target_flat = target[y_min:y_max, x_min:x_max, channel].flatten()
```

```
#concat = source_flat*mask_flat + target_flat*(1-mask_flat)
```

```
# inside the mask:
```

```
#  $\Delta f = \text{div } v = \Delta g$ 
```

```

alpha = 1
mat_b = laplacian.dot(source_flat)*alpha

# outside the mask:
# f = t
mat_b[mask_flat==0] = target_flat[mask_flat==0]

x = spsolve(mat_A, mat_b)
#print(x.shape)
x = x.reshape((y_range, x_range))
#print(x.shape)
x[x > 255] = 255
x[x < 0] = 0
x = x.astype('uint8')
#x          =          cv2.normalize(x,          alpha=0,          beta=255,
norm_type=cv2.NORM_MINMAX)
#print(x.shape)

target[y_min:y_max, x_min:x_max, channel] = x
return target

#helper
def make_blend_mask(size, object_box):
    x,y,w,h = object_box
    x0=x
    x1=x+w
    y0=y
    y1=y+h

```

```

w,h = size
mask = np.ones((h,w,3),np.float32)

for i in range(0,y0):
    mask[i]=i/(y0)
for i in range(y1,h):
    mask[i]=(h-i)/(h-y1+1)
for i in range(0,x0):
    mask[:,i]=np.minimum(mask[:,i],i/(x0))
for i in range(x1,w):
    mask[:,i]=np.minimum(mask[:,i),(w-i)/(w-x1+1))

return mask

```

```

def insert_object(mix, box, crop, mask):
    x,y,w,h = box
    crop = cv2.resize(crop, dsize=(w,h), interpolation=cv2.INTER_AREA)
    mask = cv2.resize(mask, dsize=(w,h), interpolation=cv2.INTER_AREA)

    mix_crop = mix[y:y+h,x:x+w]
    crop = norm_color_transfer(crop, mix_crop)
    mix[y:y+h,x:x+w] = mask*crop +(1-mask)*mix_crop
    return mix

```

```

#load dummy object
object_infor = [
    ['00012.jpg',[0,0,1024,848],[230,129,667,623],], # image_file, context_box,
object_box
    ['00001.jpg',[0,0,767,1023],[14,75,717,897],],

```

```

# ['00014.jpg',[0,0,1000,567],[168,87,702,419],],
  ['00021.jpg',[680,1144,1244,884],[760,1256,992,724],],
]

```

```

def load_dummy_object():

```

```

    object = []

```

```

    for image_file, context_box, object_box in object_infor:

```

```

        image_file = '../input/cots-non-verified-license-yet/' + image_file
        #download-0.jpeg'

```

```

        image = cv2.imread(image_file, cv2.IMREAD_COLOR)

```

```

        x,y,w,h = context_box

```

```

        crop = image[y:y+h,x:x+w]

```

```

        object_box = np.array(object_box)-[x,y,0,0]

```

```

        mask = make_blend_mask((w,h),object_box)

```

```

        #image_show('crop',crop, resize=0.5)

```

```

        #image_show('mask',mask, resize=0.5)

```

```

        #cv2.waitKey(0)

```

```

        object.append([crop, mask])

```

```

    return object

```

```

def load_dummy_background():

```

```

    video_id = 1

```

```

    video_frame = 9187 #9287 #9187

```

```

    image_file = 'video_%d/%d.jpg' % (video_id, video_frame)

```

```

    image = cv2.imread('../input/tensorflow-great-barrier-reef/train_images/' +
image_file, cv2.IMREAD_COLOR)

```

```
return image
```

```
# demo here!
```

```
object = load_dummy_object()
```

```
background = load_dummy_background()
```

```
mix = background.copy()
```

```
mix1 = None
```

```
for i, box in enumerate([
```

```
    [330,158,70,61],
```

```
    [208,598,45,52],
```

```
    [930,498,120,100],
```

```
]):
```

```
    crop,mask = object[i]
```

```
    #mix = insert_object (mix, box, crop, mask*0.5) -----
```

```
    x,y,w,h = box
```

```
    crop = cv2.resize(crop, dsize=(w,h), interpolation=cv2.INTER_AREA)
```

```
    mask = cv2.resize(mask, dsize=(w,h), interpolation=cv2.INTER_AREA)
```

```
    mix_crop = mix[y:y+h,x:x+w]
```

```
    crop = norm_color_transfer(crop, mix_crop)
```

```
    #crop = poisson_edit(crop, mix_crop, (mask[:,0]>0.5).astype(np.float32),  
offset=(0,0))
```

```
    mask = mask*0.8 #mixup ratio
```

```
    mix[y:y+h,x:x+w] = mask*crop +(1-mask)*mix_crop
```

```
#-----
```

```
# image_show('background',background, resize=1)
# image_show('mix',mix, resize=1)
# image_show('crop',crop, resize=4)
# image_show('mix_crop',mix_crop, resize=4)
# image_show('mask1',mask1, resize=4)
# image_show('mask',mask, resize=4)
# cv2.waitKey(0)
```

```
plt.figure(figsize=(15,20))
plt.title('original background')
plt.imshow(background[...,::-1])
```

```
plt.figure(figsize=(15,20))
plt.title('augmeted image')
plt.imshow(mix[...,::-1])
```

Результаты

Оригинальное изображение показано ниже на рисунке 1.

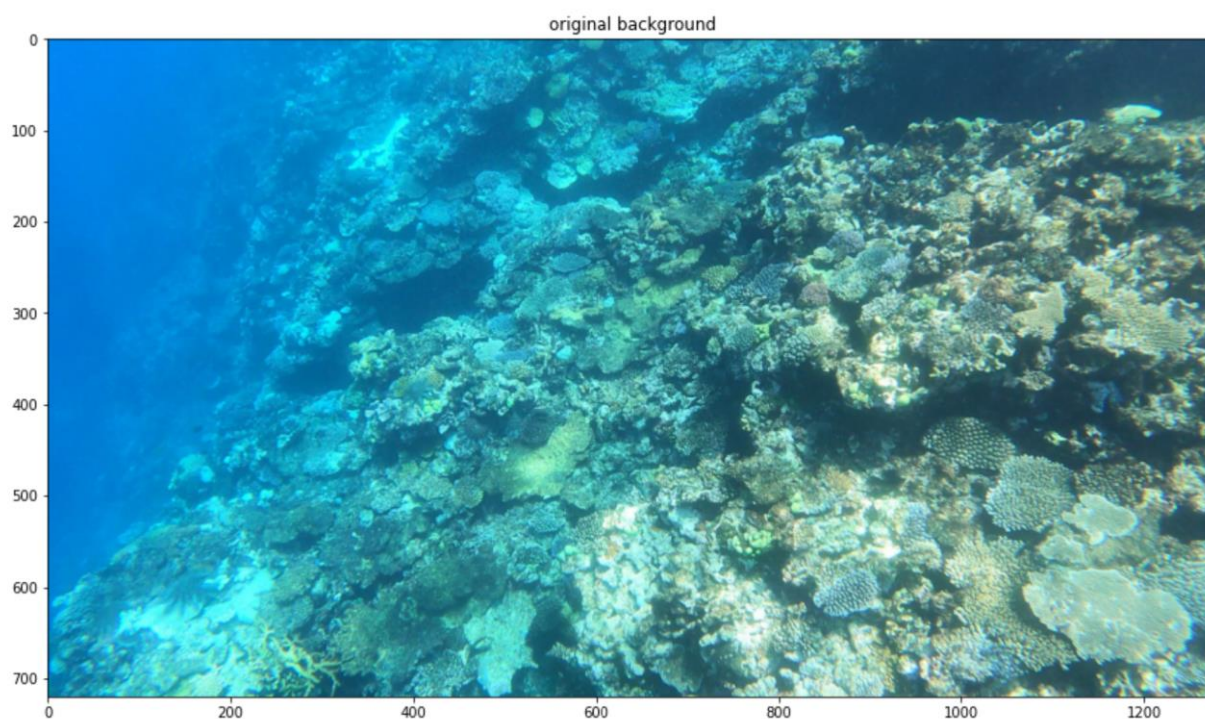


Рисунок 1. Оригинальное изображение
Улучшенное изображение показано на рисунке 2 ниже.

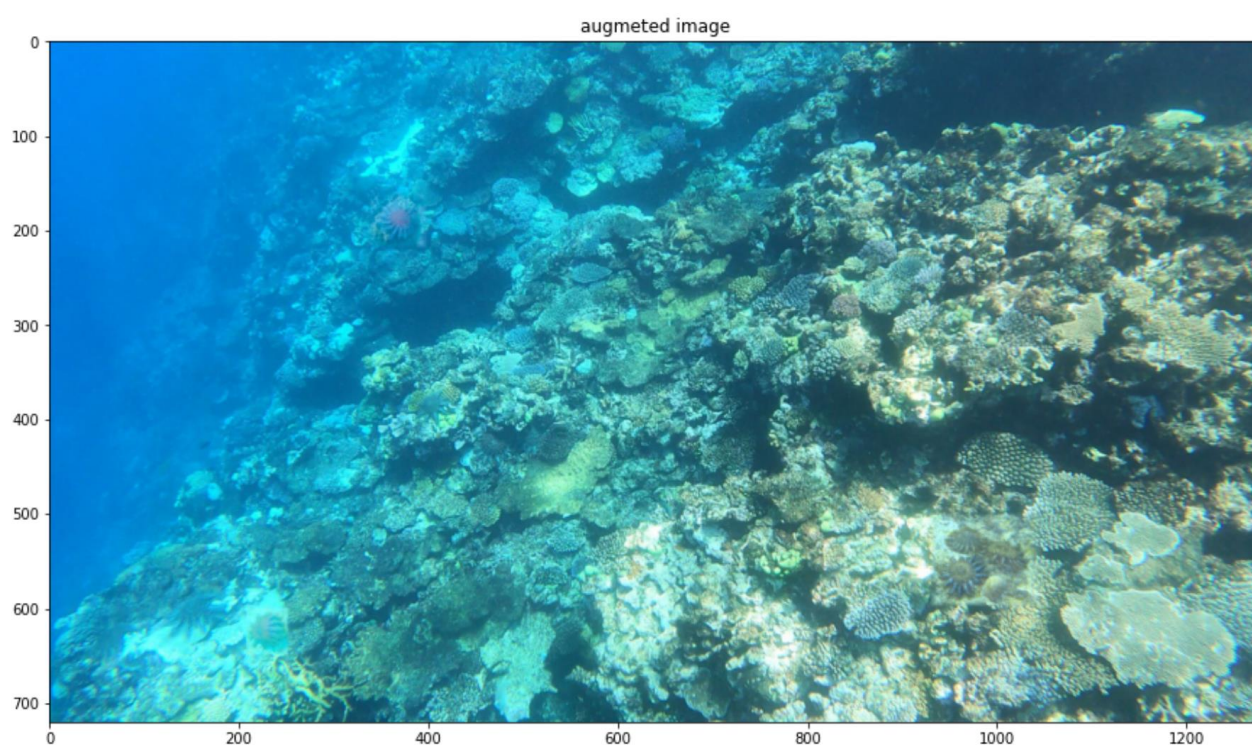


Рисунок 2. Улучшенное изображение
Гибридный алгоритм представляет значимый вклад в мониторинг и защиту Большого Барьерного Рифа, обеспечивая более эффективные средства идентификации и отслеживания звездных рыб короны, что в свою очередь способствует более эффективному управлению рифовой средой.

Заключение

Разработанный гибридный алгоритм смешивания представляет собой перспективный подход к решению сложной задачи идентификации звездных рыб короны в подводных условиях. Результаты экспериментов подтверждают его эффективность. Этот исследовательский вклад имеет важное значение для сохранения Большого Барьерного Рифа и может быть применен в других областях мониторинга подводных экосистем.

Список литературы

1. Галичий Д.А., Афанасьев Г.И., Нестеров Ю.Г. Распознавание эмоций человека при помощи современных методов глубокого обучения // E-Scio. 2021. № 5 (56). С. 316-329.
2. Годияк, В. А. Методы повышения качества изображений в телевизионных системах подводного назначения // Неделя науки и творчества - 2021 : Материалы Международного научно-практического форума студентов, аспирантов и молодых ученых. В 3-х частях, Санкт-Петербург, 17–21 мая 2021 года / Редколлегия: А.Д. Евменов (отв. редактор) [и др.]. Том Часть 1. – Санкт-Петербург: Санкт-Петербургский государственный институт кино и телевидения, 2021. С. 118-121.
3. Но Д., Афанасьев Г.И. Состояние и перспективы применения искусственного интеллекта в визуализирующей диагностике заболеваний легких // E-SCIO. 2022. №4(67). С.653-664.
4. Но Д., Афанасьев Г.И. Применение глубокого обучения в визуализации диагностики туберкулеза // Искусственный интеллект в автоматизированных системах управления и обработки данных. ИИАСУ`22. Сборник статей Всероссийской научной конференции. М.:МГТУ им. Н.Э. Баумана. Т.2. С.237-241.
5. Пхьюу В.Т., Афанасьев Г.И. Интеллектуальная система идентификации человека по отпечаткам пальцев // Искусственный интеллект в автоматизированных системах управления и обработки данных. ИИАСУ`22. Сборник статей Всероссийской научной конференции. М.:МГТУ им. Н.Э. Баумана. Т.2. С. 346-351.
6. Ткаченко, К. С. Коралловые рифы перед экологическими угрозами XXI века / К. С. Ткаченко // Журнал общей биологии. 2015. Т. 76. № 5. С. 390-414
7. Хаустов, П. А. Алгоритм улучшения качества подводных снимков на основе нейроэволюционного подхода / П. А. Хаустов, В. Г. Спицын, Е. И. Максимова // Фундаментальные исследования. 2016. № 4-2. С. 328-332.
8. Chuanbin Lu. Coral bleaching crisis on the Great Barrier Reef. // China Fisheries 2020. 3. P. 96-97.