

Application of Artificial Intelligence of Things on Fishery Industry

專題生：吳承威，王樸

指導教授：林政宏

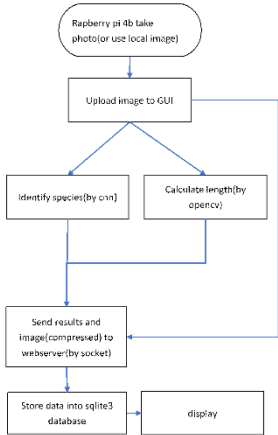
摘要—Artificial Intelligence(AI) and Internet of Things(IOT) are booming in recent years. Thus, there is more and more application of Artificial Intelligent of Things (AIOT) to make our life more convenient. Our project is the application of both of above with the image processing technology to identify fish species and determine the fish size by calculating the fish length

I. 導論

Fishery has always been an important industry in Taiwan. However, it is difficult to identify the fish species, calculate the length and record it in time when fish is caught. Therefore, we designed a device to solve the problem. We run Convolutional Neural Network(CNN) on the Raspberry Pi 4b to identify fish species, and use python Open CV to capture the frame so that we can calculate the fish length. Then we design a Graphic User Interface (GUI) to upload the fish photos and display the results on a web page.

II. 原理與架構簡介

A. Structure and Device



Our device is the Raspberry Pi 4b attached with a camera, a display screen, a stick and a monopod.

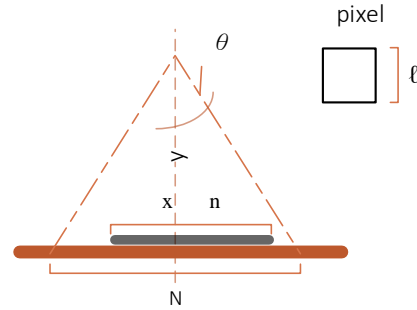
Through the GUI, the photos taken on site or other existing images can be fed to the device for further analysis.

Subsequently, we can identify the fish species and calculate the fish length. Finally, we can send data to the webserver.

B. Calculating the length

For this, we use a very simple methods of image processing and math to calculate the length of the fish. It consists of getting the contour of the fish using Otsu threshold

and then using a method from Open CV called minareact(), to calculate the length in term pixel of the fish.



$\theta = \text{half camera angle of vision in } x \text{ axis}$

$n = \text{object number of pixel}$

Also, to calculate the length per pixel ℓ , we use the equation:

$$\ell \approx 2 \times \frac{y}{N} \times \tan \theta$$

where N is the number of pixels in this axis and y is the distance between camera and the object.

Once we have the length in term of pixel, we can calculate the real length in centimeter by multiplying the length per every pixel we calculated before.

$$x \approx \ell \times n$$

where x is actual length

C. Model architecture

For the classification part of the program we use a neural network for this job. Originally we decided to use the model YOLOv3-tiny[1] and YOLO v4[2]for our project, but because the model was too big for a raspberry pi 4 to run properly and with high accuracy, we decided to discard the model.

Thus, the model we decided to use is inspired by the model that appears on the paper [3], which better suits our requirement of lightness and accuracy. The difference is that our model has an extra dense layer before the output. The reason of this, is to give the network an extra layer of liberty.

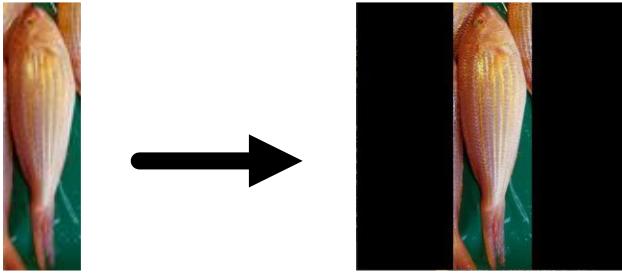
layer	number of kernels	size	stride	padding	activation	dropout
Inputs	400x400x3					
conv1	32	5x5	1x1	2x2	ReLU	
maxpool1		5x5	5x5	1x1		
conv2	64	5x5	1x1	2x2	ReLU	
maxpool2		5x5	5x5	1x1		
conv3	32	5x5	1x1	2x2	ReLU	
maxpool3		5x5	5x5	1x1		
Flatten	288					
Dense1		2048			ReLU	0.2%
Dense2		2048			ReLU	0.2%
Dense3		5			softmax	

III. 實驗步驟、過程、與結果

A. Preprocessing

Before feeding the dataset to the neural network, we make some extra processing to artificially increase the dataset, to make sure that the model doesn't over feat. These processes are: image random rotation, random horizontal or vertical flip, add random perspective and random image crop.

Also, because the model we use need a specific image size (400x400x3), we need to resize the image to meet this condition. We want to feed the network with the same aspect ratio, but we don't want the image of the fishes gets distorted, so if the image does not have the same image ratio we add black spaces, to equalize the aspect ratio and then resize.

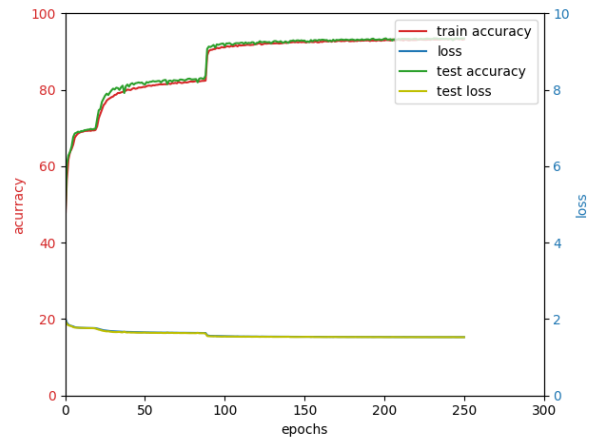
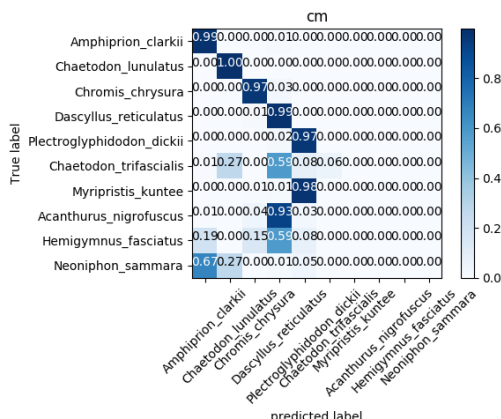


B. Datasets

(1) Test model dataset

We use a dataset provided from the website: <http://groups.inf.ed.ac.uk/f4k/GROUNDTRUTH/RECOG/> and select 10 classes which have more data from it to train our neural network. In this case the first five classes have more than 2000 images, specially the first one which has 11312. Summing this five classes number of images it makes more than 18000 images. However, the last 5 classes have less than 200 images per class.

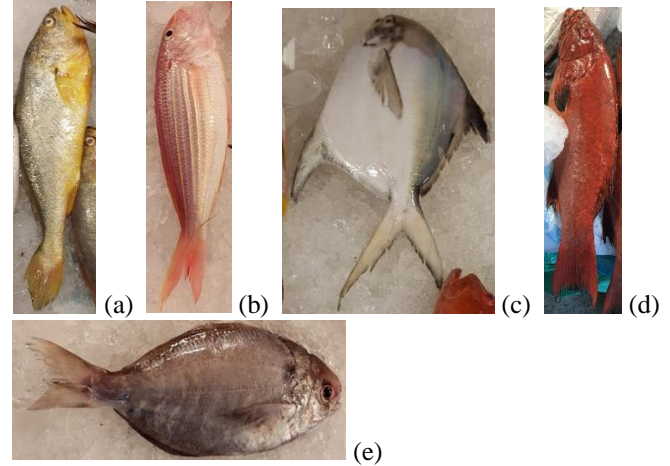
Due to the difference in the amount of data, we can see the difference of results from the confusion matrix below. The first 5 classes have been identified correctly with accuracy more than 97%. However, the last 5 classes either have been misidentified or have low accuracy.



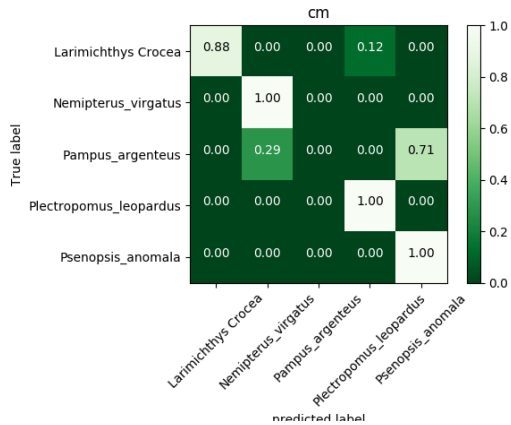
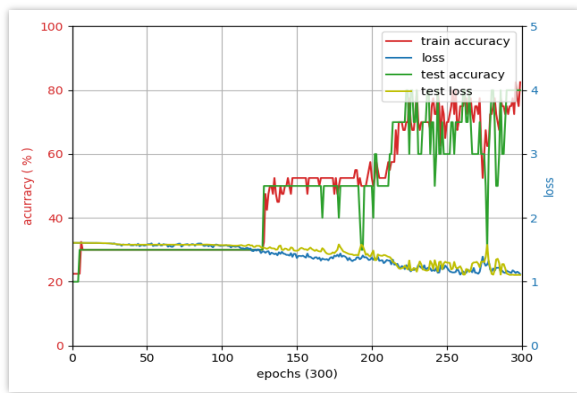
(2) Our dataset

Since we couldn't find a proper dataset to meet our project's needs, and it is difficult to collect data in fishing harbor, we decided to collect data in Taiwanese markets.

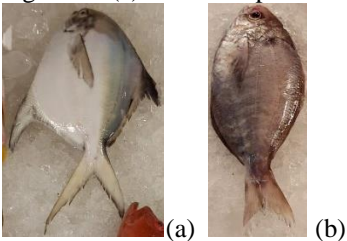
The dataset collected in Dongmen Market and Binjiang market consists of 5 classes of fishes that are common in Taiwanese fish markets. This are: *Larimichthys Crocea* (a), *Nemipterus virgatus* (b), *Pampus argenteus* (c), *Plectropomus leopardus* (d), *Psenopsis anomala* (e). This classes are selected based on the amount of images we collected.



Originally, the data set consist of photos that we took directly from the market, plus some other images downloaded from the internet. Due to lack of images, which does not exceed 30 photos per class, the best accuracy that we managed to get does not pass over 80%.



From the confusion matrix of the model, we can see that the classes that has more trouble classifying are Pampus argenteus (a) and Psenopsis anomala (b).

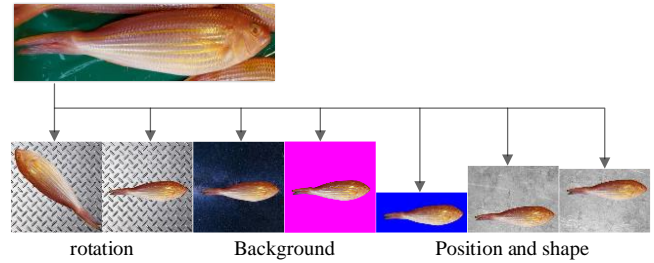


As it can be seen, both fishes have a very similar shape, color and background, which indicates that the data we have is not enough. So we decided that we have to do some data augmentation, manually.

C. Manual data augmentation

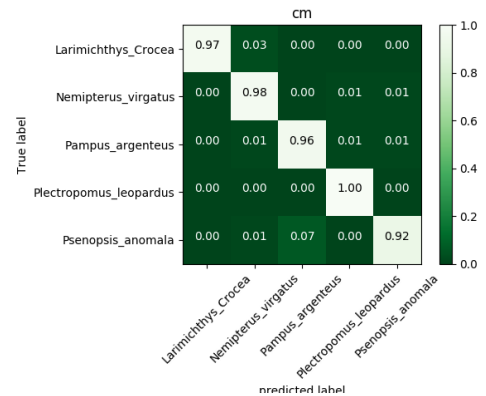
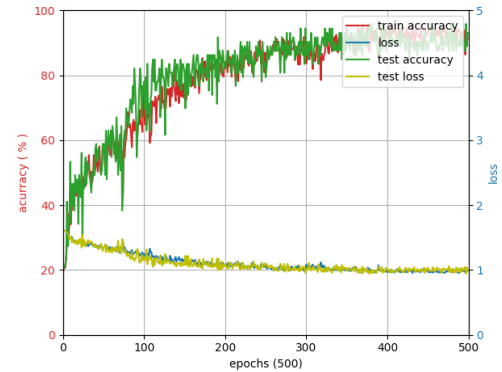
The methods we use to add extra data consist, using a photo editor and change the background of some of the fishes, the rotation with respect of the original photo, and changing the position of it.

The goal of this method is to make sure that the network can differentiate the fish from the background, and the fact that not all the fishes of the same class has the same shape, position, rotation and other small details.



This was applied on 2 of the fishes of every class. And with different backgrounds increasing from less than 30 photos per class, to more than 70 photos per class.

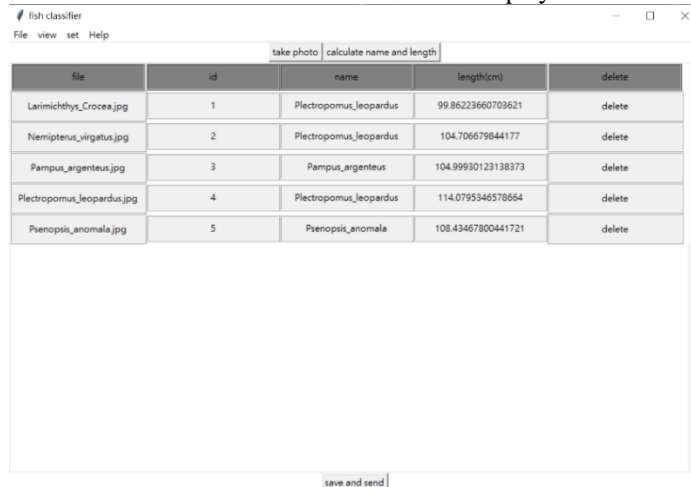
By doing this we manage to increase from the original 80 % of accuracy to 96 % of accuracy.



From the confusion matrix it can be seen that the manual data augmentation has an impact on the training results.

D. GUI and Webpage

(1) GUI: The device allows the users to take photos on site then upload them by clicking a button as shown below. Users can also directly feed other existing images to the device via the ‘file’ function.. By clicking another button ‘calculate’, we can identify the species and calculate the fish length. With that, we send the results to a web server database to display it.



(2) Webpage: A webpage is made ready with a search function, either by name or length, to display the results from the database.



IV. 結論

Although our identification accuracy is more than 90% in all five species, the dataset is still too small that our results sometimes go wrong. On the other hand, data augmentation for training neural network in absence of sufficient data was very useful. We will continue to make the network more robust by adding new data and using new data argumentation methods.

誌謝

Special thanks to the lab seniors and our professor for the help.

參考資料

- [1] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. CoRR, abs/1804.02767, 2018.
- [2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” arXiv preprint arXiv:2004.10934, 2020.
- [3] Rathi, D., Jain, S., and Indu, S., “Underwater Fish Species Classification using Convolutional Neural Network and Deep Learning”, <https://arxiv.org/abs/1805.10106v1>, 2018.