# Temporal 3D RetinaNet for fish detection

Zhou Shen[1,2]
[1]College of Computer Science and Engineering
Australian National University
Canberra, Australia
zhou.shen@anu.edu.au

Chuong Nguyen[2]
[2]Cyber Physical Systems - Imaging and Computer Vision
CSIRO Data61
Canberra, Australia
chuong.nguyen@csiro.au

***Abstract*—Automatic detection and tracking of fish provides valuable information for marine life science. Deep convolutional networks have been applied with some success but performance is affected by challenging imaging conditions including complex background, variation of light and the low visibility of the underwater environment. Existing works including Fast R-CNN and RetinaNet rely on single frame fish detection and suffer noisy and unreliable detections. In this paper, we propose and examine two 3D deep learning networks using temporal features to improve fish detection performance. The first one called 3D-backbone RetinaNet based 3D ResNet for temporal information is found worse than 2D RetinaNet. The second one called 3D-subnets RetinaNet based on 3D Regression subnet and Classification subnet to extract the temporal information is found better than 2D RetinaNet. To validating the performance of these networks, we also created a new fish data set which will be made publicly available with codes of the proposed networks.**

***Index Terms*—Fish detection, 3D convolution, Temporal feature, Deep learning, RetinaNet, 3D-subnets RetinaNet**

## I. Introduction

Fish videos by underwater observation system have been used to study the marine biology [1]. Due to large population of fish, the variation of light, and the low visibility and complex the underwater environment, it is often a labour-intensive and time-consuming job to manually analyze fish in those videos. Thus, there have been works to automate this process and tackle the challenging imaging conditions.

Several fish detection studies have been done including sonar imaging [2], edge detection [3], optical gated sampling [4], Gaussian Mixture Model (GMM) [5], [6], and Principal Component Analysis (PCA) [9]. As deep learning techniques have been developed quite rapidly in the field of object detection [7], they have been applied in some recent works on fish detection. Li et al. [8] used Fast R-CNN [10] as a fish detector. Levy et al. [11] used RetinaNet [12], which was proved to be faster and more accurate to detect fish than the Fast R-CNN, at least with limited data. In both works, their detectors were based on single-frame detection on individual frames extracted from videos.

Although [11] achieved good improvement with RetinaNet's single-frame detection, there are many cases that detection fail easily. We believe that additional temporal information from multiple frames capturing the moving pattern of fish and the light fluctuation could provide valuable visual information for better detection. [13] showed that temporal information could improve action recognition, and [14] also showed that the temporal information could be helpful to improve the performance of water-hazard detection. In an underwater environment, fish move by their own specific motions, while background objects like underwater plants and rocks are static, and light refraction by water surface changes periodically. Some fish could stop and float along water flow, but they still move mouths and gills to breath. As fish swim in a different speed than water flow, the light reflected on its body should be changed differently than the light on other background objects. These cues could not be captured and processed by single-frame detections, therefore leading to false detections.

In this work, we propose and validate two network models called 3D-backbone RetinaNet and 3D-subnets RetinaNet that use temporal information to improve the performance of fish detection in videos. The two models are based on RetinaNet [12] which is a single-stage detector using focal loss [12] to avoid the class imbalance problem. Our work is motivated by [13] and [14] which studied the effect of temporal information from 8 continuous frames and its use in action recognition as well as water-hazard detection.

By training and testing on our new dataset, we found that if implemented properly, temporal-based networks could achieve better fish detection accuracy. Comparing with the 2D RetinaNet as baseline, the 3D-backbone RetinaNet performs worse, while the 3D-subnets RetinaNet achieves better fish detection accuracy. The former is based on inflated 3D RetinaNet and the latter is based on 3D Regression subnet and 3D Classification subnet. We demonstrate that by increasing the number of continuous frames, the accuracy of 3D-subnets RetinaNet increases.

Due to limited availability of public datasets for benchmarking temporal fish detections, we have created a new fish detection dataset to benchmark our proposed networks. This dataset is released with the codes of our proposed networks at [24].

## II. Method

In this work, we include temporal information to existing RetinaNet by applying the 3D convolution and pooling method introduced by [13] to extract hidden temporal features from multiple continuous video frames. We examine two architectures called 3D-backbone RetinaNet and the 3D-subnets

RetinaNet to see which approach is better to extract and use temporal information for fish detection.

Based on RetinaNet [12], our two models consist of ResNet-50 [18] and Feature Pyramid Network (FPN) [17] backbone, a Classification subnet and a box Regression subnet. Generally, an input of 8 continuous RGB video frames would pass through the ResNet-50 backbone to generate feature maps $X_2$, $X_3$ and $X_4$ which then pass through the FPN backbone to compute pyramid features $P_3$ to $P_7$. After that, the features would pass through the two subnets: the Classification subnet and the Regression subnet. The classification subnet would classify anchor boxes, while the Regression subnet would regress object bounding boxes from anchor boxes.

### A. 3D-backbone RetinaNet

3D-backbone RetinaNet uses a 3D-ResNet backbone to extract temporal feature maps. The architecture of the network is shown in Figure 1.

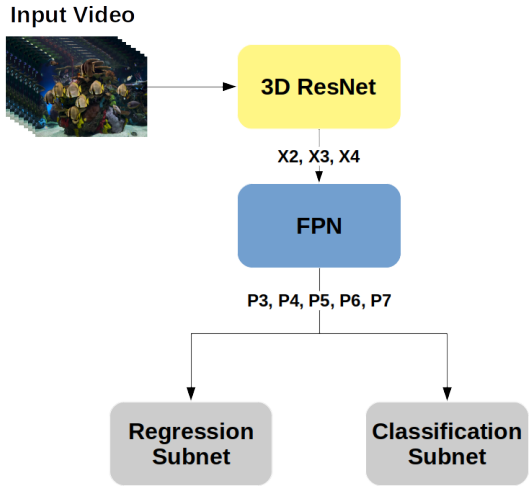


Fig. 1. Our 3D-backbone RetinaNet architecture. X2, X3 and X4 are outputs from 3D ResNet, and P3, P4, P5, P6, P7 are outputs from FPN.

The detail of the 3D-Resnet-50 backbone is shown in Figure 2. The Conv layer is a 3D convolutional layer with a $7 \times 7 \times 7$ kernel size and the stride of 2 in all 3 dimensions. A 3D max pooling layer is implemented after the convolutional layer. For the bottleneck blocks, from Block1 to Block4, we add temporal dimension by modifying all $1 \times 1$ kernels to $1 \times 1 \times 1$ kernels and $3 \times 3$ kernels to $3 \times 3 \times 3$ kernels. Following each 3D convolutional layer, there is a batch normalisation layer and a Relu layer. The outputs of Block2, Block3 and Block4 will pass through FPN to generate feature pyramid levels P3 to P7.

The inflation method [15] is also applied in our work. According to [15], we repeat the kernel parameters from the 2D pre-trained model 7 times or 3 times according to the kernel size in our 3D network.

### B. 3D-subnets RetinaNet

3D-subnets RetinaNet however uses the original 2D RetinaNet backbone to generate feature maps frame by frame, but it adopts 3D classification subnet and 3D regression subnet to extract temporal features. The group of 8 corresponding feature maps of the 8 input frames would be combined to a 3D one as the input of 3D-subnets. The architecture is shown in Figure 2.

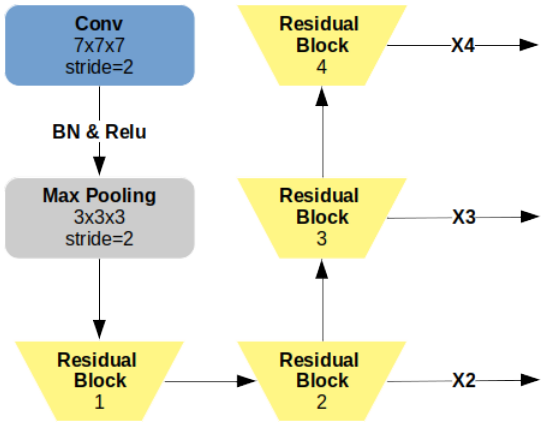


Fig. 2. The architecture of 3D-ResNet-50. BN and Relu mean Batch Normalization layer and Relu layer after the 3D convolutional layer. The residual blocks are bottleneck layer groups. X2, X3 and X4 are outputs from the corresponding blocks which will be passed to FPN.

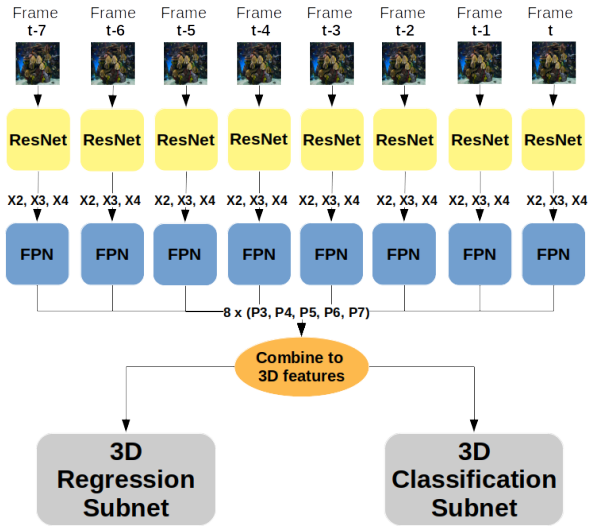


Fig. 3. Our 3D-subnets RetinaNet architecture. ResNet and FPN blocks are the same as in 2D RetinaNet to process individual frame separately. A group of 8 sets of P3, P4, P5, P6, P7 corresponding to 8 frames becomes an input into 3D Regression Subnet and 3D Classification Subnet.

Figure 4 shows the detail of the two 3D subnets. These two subnets have similar structures including 5 3D-convolutional layers. A ReLU layer is used after the first 4 3D-convolutional layers, but the Classification subnet has an extra Sigmoid layer after the final 3D convolutional layer. All 3D-convolutional layers have $3 \times 3 \times 3$ kernels, while their stride are different. First 2 convolutional layers have $1 \times 1 \times 1$ stride, but the rest 3 layers have $2 \times 1 \times 1$ stride.

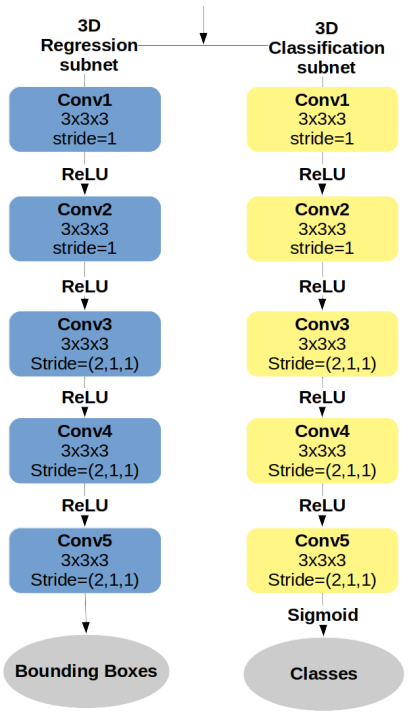


Fig. 4. The detailed structure of 3D regression subnet and 3D classification subnet of our 3D-subnets RetinaNet architecture.

## III. Experiments and Results

### A. Experiments

We created a new fish detection dataset [24] from a YouTube video [19] to train and test our networks. The dataset contains 1640 frames with our ground truth annotations. For training temporal networks, we create overlapping video clips starting at every 4 frames with temporal length equal to the input number of frames of the network. Our network is trained on 1280 videos and tested on 296 videos.

The focal loss [12] and the ADAM optimizer [20] are used. The total loss is defined as:

$$\textbf{Total loss} = \text{binary focal loss} + \text{smooth L1 loss} \quad (1)$$

The binary focal loss is defined as:

$$\textbf{FL} = \begin{cases} -\alpha(1-p)^{\gamma} log(p), & \text{if } y = 1 \\ -(1-\alpha)p^{\gamma} log(1-p), & \text{if } y = 0 \end{cases} \quad (2)$$

$\alpha$ here is a weighting factor that is helpful for class imbalance problem, $y$ represents two class labels in binary classification, $\gamma$ is the focusing parameter which can adjust the down-weighted rate of easy examples, and $p$ is the estimation possibility of the fish class while $1-p$ is the estimation possibility of the background. The smooth L1 loss introduced in Fast R-CNN [25] is defined as the sum of loss over the top left coordinates (x,y) and width and height (w, h) of bounding boxes:

$$\textbf{SL} = \sum_{i \in x,y,w,h} \text{smooth}_{\text{L1}}(t_i^u - v_i) \quad (3)$$

where,

$$\textbf{smooth}_{\textbf{L1}}(\text{x}) = \begin{cases} -0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (4)$$

$t^u$ represents the scale-invariant translation and height/width shift in the log space of the ground truth class $u$, and $v$ is the ground truth bounding box.

In our implement, we set $\alpha = 0.25$ and $\gamma = 2.0$ which are the same as in [12]. ADAM optimizer is chosen with the learning rate of $1e{-}5$. The training and testing of the networks are carried out on a mid-range GPU desktop with an NVIDIA RTX 2070S GPU which has 8GB RAM. To accommodate our proposed networks on this limited GPU memory, input images are resized to $320 \times 544$ pixels.

To compare our 3D networks with others, video clips of 8 continuous frames are used to train our 3D networks. An example of 8 frames with annotations provided on the last frame is shown in Figure 5. We do not use future frames, only current and past frames since the camera does not provide future frames in real-life detection. The D-subnets RetinaNet is also trained with input number of frames of 4, 8 and 16 to study the impact of the temporal depth.

As 2D RetinaNet [12] was used for fish detection in [11] with single frames as input, it is considered as the baseline of our work to compare with our 3D-backbone RetinaNet and 3D-subnets RetinaNet. SSD [22] and Faster R-CNN [23] are also fine-tuned and included in the comparison as the latter was also used in fish detection in [10]. As the training set is not large enough, both the baseline 2D RetinaNet and the 3D-subnets RetinaNet use a pre-trained 2D-ResNet-50 on ImageNet [21] before fine-tuning on our fish dataset, while the 3D-backbone RetinaNet uses inflation method [15] to expand the 2D pre-trained model to 3D before fine-tuning.

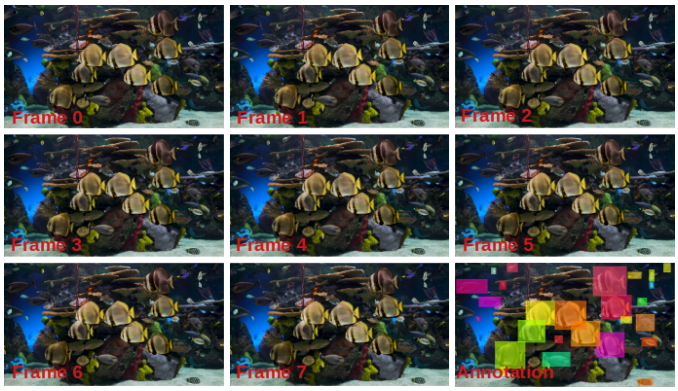


Fig. 5. The example of 8 continuous frames input for 3D RetinaNet and the annotation for the last (current) frame.

### B. Results

An example of the detection result is shown in Figure 6. The baseline 2D RetinaNet cannot detect the blue fish (Yellow-tail Blue Damsel) on the right. The 3D-backbone RetinaNet also cannot detect that fish, and it even outputs a

low precision bounding box for the bottom-left fish (Pennant coral fish). The 3D-subnets RetinaNet can detect all three fish, and all the bounding boxes are at good size and position. It seems to retain the good accuracy of original RetinaNet while having additional sensitivity and temporal consistency. The right blue fish is small and a little far away, and moves slowly in front of complex background, while the bottom-left fish moves quickly between frames. This suggests that 3D-subnets RetinaNet performs well for complex background and fast motion.

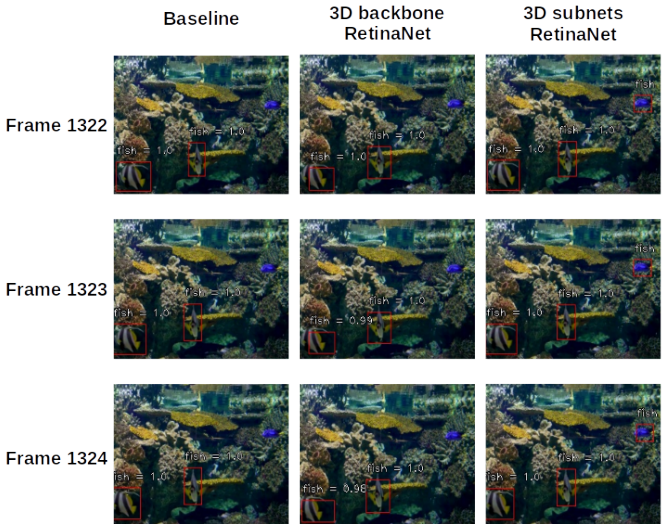


Fig. 6. The example of detection results of three models in frame 1322, frame 1323 and frame 1324. 3D-subnet RetinaNet picks up the blue fish (Yellow-tail Blue Damsel) which is missing from the baseline and 3D-backbone RetinaNet.

To evaluate the detectability, we adopt the average precision (AP) of bounding boxes with Intersection over Union (IoU) $\geq 0.5$. The AP is defined as:

$$\mathbf{AP} = \int_0^1 p(r)dr \tag{5}$$

which represents area under the precision-recall curve. $p(r)$ is the precision-recall curve and $r$ is the recall value.

Table I shows the qualitative comparison of detect ability and computational requirement between SSD [22], Faster R-CNN [23], the baseline 2D RetinaNet, 3D backbone RetinaNet and 3D subnets RetinaNet. The 3D-subnets RetinaNet has the best AP value of 0.733 with the 8-frame inputs as compared to 0.539 for 3D-backbone RetinaNet with same inputs and 0.693 for 2D RetinaNet with single frame inputs.

TABLE I
RESULT COMPARISON

| Models | AP | Time (sec/frame) |
|---|---|---|
| SSD [22] | 0.649 | **0.037** |
| Faster R-CNN [23] | 0.595 | 0.201 |
| 2D RetinaNet [12] | 0.693 | 0.099 |
| 3D-backbone RetinaNet (ours) | 0.539 | 0.125 |
| 3D-subnets RetinaNet (ours) | **0.733** | 0.236 |

Although also extracting temporal information, the 3D-backbone RetinaNet does not work as well as 3D-subnets RetinaNet. The worse result of the former could only be explained by the differences in the structure of the backbone network. One possible reason is that in the 3D-backbone RetinaNet, the output 3D feature maps is reduced to 2D to fit the input of 2D FPN causing a bottle neck. Although the feature maps does contain some temporal information, the 2D subnets could simply treat them as common 2D feature maps and temporal features effectively blurs the 2D features leading to worse performance. Another possible reason is that both baseline RetinaNet and 3D-subnets RetinaNet apply more effective transfer learning with pre-trained ImageNet [21] ResNet-50 backbone, while the 3D-backbone RetinaNet uses inflation method to convert the same 2D pre-trained model into 3D model. Such inflation may not be enough to capture sufficient temporal information as compared to 3D Regression and Classification subnets. As a result the additional temporal information allow 3D-subnets RetinaNet to be more robust with complex background.

The computational requirement is measured by the computation time of one frame detection. Among three RetinaNet models, the baseline 2D RetinaNet has the fastest speed of 0.099 sec/frame, while the two 3D models have slower speeds of 0.125 sec/frame and 0.236 sec/frame, which are due to more parameters to compute in 3D kernels. The 3D-subnets RetinaNet needs around twice the time of the others since each of the two subnets need to compute on the 5 pyramid features with 3D kernels.

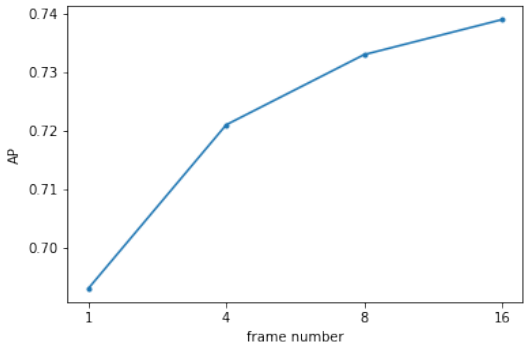


Fig. 7. The averaged precision (AP) as function of input number of frames. Here 2D RectinaNet [12] uses 1 frame input, and our proposed 3D-subnets RetinaNet uses between 4 to 16 frame input for better AP.

We also test whether the different input number of frames can affect the testing accuracy. Figure 7 shows the averaged precision (AP) as function of input number of frames. With the input frame number increasing from 1 to 16, the AP rate also increases. The growth rate from 1 input frame to 4 input frames is the fastest one due to our additional temporal information. However, the growth rate slows down from 4 input frames to 8 input frames and slightly slows down again from 8 input frames to 16 input frames. The reason to explain might be that the most important temporal information usually hides

in frames near the current frame, and too large time horizon might have less extra useful temporal information that helps improve the precision.

## IV. Conclusion

In this paper, we propose and examine two 3D models based on the RetinaNet: 3D-backbone RetinaNet and 3D-subnets RetinaNet for fish detection with additional temporal information. The 3D networks utilize temporal features extracted from 8 consecutive frames by 3D convolution. Although using 3D ResNet to include temporal information, 3D-backbone RetinaNet does not perform better than 2D RetinaNet. On the other hand, 3D-subnets RetinaNet uses 3D Regression subnet and 3D classification subnet for temporal information, and can improve detection result. So we recommend using 3D-subnets RetinaNet for better performance. By increasing the number of input frames, the accuracy of 3D-subnets RetinaNet is shown to increase with the most significant jump between 1 frame to 4 frames. This shows that temporal information play a major role in the accuracy improvement. Due to the increase of kernel dimension, this 3D network requires more computation than the 2D RetinaNet. We also find that the increase of input frame size can improve the detection precision, but the promotion effect by larger frame number becomes unapparent due to less extra useful temporal information. Future works include fish swimming behaviour tracking and optimisation to reduce computation time.

## References


[1] D. Mallet and D. Pelletier, "Underwater video techniques for observing coastal marine biodiversity: A review of sixty years of publications (1952–2012)," Fisheries Research, vol. 154, pp. 44-62, 2014.

[2] J. A. Holmes et al, "Accuracy and precision of fish-count data from a "dual-frequency identification sonar" (DIDSON) imaging system," ICES Journal of Marine Science, vol. 63, (3), pp. 543-555, 2006.

[3] G. T. Shrivakshan and C. Chandrasekar, "A Comparison of various Edge Detection Techniques used in Image Processing," International Journal of Computer Science Issues, vol. 9, (5), pp. 269, 2012.

[4] L. J. Campbell et al, "FISH Detection of PML-RARA Fusion in ins(15;17) Acute Promyelocytic Leukaemia Depends on Probe Size," BioMed Research International, vol. 2013, pp. 164501, 2013.

[5] Z. Y. Yan, J. Y. Song and L. Li, "Moving Object Detection Based on the Fish," Applied Mechanics and Materials, vol. 644-650, pp. 1253-1256, 2014.

[6] E. Hossain, S. M. S. Alam, A. A. Ali and M. A. Amin, "Fish activity tracking and species identification in underwater video," 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, 2016, pp. 62-66, doi: 10.1109/ICIEV.2016.7760189.

[7] Z. Zhao et al, "Object Detection With Deep Learning: A Review," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, (11), pp. 3212-3232, 2019.

[8] Xiu Li, Min Shang, H. Qin and Liansheng Chen, "Fast accurate fish detection and recognition of underwater images with Fast R-CNN," OCEANS 2015 - MTS/IEEE Washington, Washington, DC, 2015, pp. 1-5, doi: 10.23919/OCEANS.2015.7404464.

[9] M. Ravanbakhsh et al, "Automated Fish Detection in Underwater Images Using Shape-Based Level Sets," The Photogrammetric Record, vol. 30, (149), pp. 46-62, 2015.

[10] R. Girshick, "fast r-cnn," in 2015, . DOI: 10.1109/ICCV.2015.169.

[11] D. Levy et al., "Automated Analysis of Marine Video with Limited Data," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, 2018, pp. 1466-14668, doi: 10.1109/CVPRW.2018.00187.

[12] T. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2999-3007, doi: 10.1109/ICCV.2017.324.

[13] D. Tran, L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 4489-4497, doi: 10.1109/ICCV.2015.510.

[14] J. Li, C. Nguyen and S. You, "Temporal 3D Fully Connected Network for Water-Hazard Detection," 2019 Digital Image Computing: Techniques and Applications (DICTA), Perth, Australia, 2019, pp. 1-5, doi: 10.1109/DICTA47822.2019.8945849.

[15] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 4724-4733, doi: 10.1109/CVPR.2017.502.

[16] K. Hara, H. Kataoka and Y. Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 6546-6555, doi: 10.1109/CVPR.2018.00685.

[17] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection, " 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 936-944, doi: 10.1109/CVPR.2017.106.

[18] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[19] Balu - Relaxing Nature in 4K, "The Best 4K Aquarium for Relaxation II Relaxing Oceanscapes - Sleep Meditation 4K UHD Screensaver," https://youtu.be/YRF GuvYeug, July 2018.

[20] K. Diederik and B. Jimmy, "Adam: A Method for Stochastic Optimization," 2014 International Conference on Learning Representations.

[21] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu and A. C. Berg, "Ssd: Single shot multibox detector", October 2016, In European conference on computer vision, pp. 21-37. Springer, Cham.

[23] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

[24] Z. Shen and C. Nguyen, Github-Temporal-3D-RetinaNet-for-fish-detection, https://github.com/shenzhouchn/Temporal-3D-RetinaNet-for-fish-detection, September 2020.

[25] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.