Automatic Video Analysis for Electronic Monitoring of Fishery Activities

Tsung-Wei Huang

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Jenq-Neng Hwang, Chair

Blake Hannaford

Linda G. Shapiro

Program Authorized to Offer Degree:

Electrical and Computer Engineering

©Copyright 2019

Tsung-Wei Huang

University of Washington

Abstract

Automatic Video Analysis for Electronic Monitoring of Fishery Activities

Tsung-Wei Huang

Chair of the Supervisory Committee:

Jenq-Neng Hwang

Department of Electrical and Computer Engineering

Recently, automated imagery analysis techniques have drawn increasing attention in fishery science and industry. Compared to traditional human observing and monitoring, automated imagery analysis techniques are more scalable and deployable, and thus have been widely used in recent years for numerous fishery survey tasks, such as abundance estimation, species identification or length measurement. One of the emerging fishery survey tasks which can effectively take advantage of the automated imagery analysis is the electronic monitoring (EM) for fishery activities. The goal of EM is to monitor the fish catching on fishing vessels, either for scientific survey or legal purpose. For example, a fishing vessel may not retain fish catching if the length is below some threshold or exceeding the quota of the vessel for certain species. Therefore, accurate tracking, counting, measurement and species classification is required in the EM systems. There are, however, challenges from the inspected subjects and operation environments.

Deformable objects, noise from the wild sea surface, and dynamic background, make conventional tracking, segmentation and classification methods unreliable.

To overcome the challenges encountered in the electronic monitoring, this dissertation presents an online 3D tracking and segmentation system for stereo video based monitoring of rail fish catching on wild sea surface. Based on the result of a pre-trained image object (fish) detector, a Kalman filtering-based tracking system overcomes the issues of low detection scores of deformed objects and unreliable bounding boxes by rescoring multiple object proposals using spatial information in 3D. A clustering-and-scoring strategy is then applied on the depth map, so that a plane classification method can effectively segment the objects from the dynamic background without any prior modeling. The object segmentation is further refined using fully connected conditional random fields based on color and geometric features. With the segmentation results, we can measure the 3D lengths of objects and update the positions of bounding boxes to help tracking. Experimental results show that a reliable tracking and measurement performance under noisy and dynamic sea surface environment is achieved.

Once fish are tracked and measured, one of the primary tasks in the electronic monitoring is to identify the species of fish. In the work of object classification, one challenge is that the feature generation needs to be robust with fish in any orientations and poses, which yield diverse visual features and large within-class variation. Other challenges include the high visual similarity among fish species. Therefore, in this dissertation, we utilize the deep metric learning to learn a feature representation which can separate visually similar species in the feature space. By adding more constraints based on the temporal order of image sequences, we can make the model to learn a more structured and compact feature space. Besides, by exploiting the clustering properties and temporal relationship in the feature space, the learning of the model can be further improved.

Combining all the stages above, the proposed electronic monitoring system can automatically process input video data, and analyze the fishery activities. The monitoring results can be further used to perform fish abundance estimation for either regulatory or scientific research purposes.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Throughout the journey of my PhD study I have receive a great amount of support and assistance. I would like to first express my deepest appreciation to my advisor, Prof. Jenq-Neng Hwang, for his guidance and advice in research direction and methodology. From him, I learned how to approach research problems and give good presentations.

I would like to thank my committee members, Prof. Mark Ganter, Prof. Blake Hannaford and Prof. Linda Shapiro, for the valuable suggestions since my General Exam.

I very much appreciate the National Oceanic and Atmospheric Administration (NOAA) for providing the enormous imagery data.

I would like to acknowledge the current colleagues, alumni and visiting scholars I have met at the Information Processing Lab (IPL), Meng-Che Chuang, Kuan-Hui Lee, Xiang Chen, Qiuyu Chen, Adwin Jahn, Wenhuan Wei, Younggun Lee, Jounsup Park, Hao Xiao, Chen Bai, Yuxuan Jiang, Zexin Li, Zheng Tang, Renshu Gu, Gaoang Wang, Li Chen, Haotian Zhang, Jiarui Cai, Yizhou Wang, Chengqian Ma, Xinyu Yuan, Shih-Hao Yeh, Yao-Chung Liang, Shen-Chi Chen, Shih-Gu Huang, Yen-Shuo Lin, Sheng-Ting Shen, Tao Liu, Junchao Yang, Wei Huang, Xu Liu, Hung-Min Hsu, Yanting Zhang, Fangyi Zhu and Ping Zhang, for their help and discussions. Their extensive knowledge and innovative thinking make the research group an excellent environment.

I would also like to thank all the friends I have met at the University of Washington, for enriching my life and exploration of Seattle and the Pacific Northwest.

And finally, last but not least, I am greatful to my family, for their endless love and support along the way. Thank you.

1 Introduction

1.1 Motivations

Recently, automated imagery analysis techniques have drawn increasing attention in fishery science and industry [1-11], because they are more scalable and deployable than traditional manual survey and monitoring approaches. One of the emerging fishery survey tasks is the electronic monitoring (EM), which can effectively take advantage of the automated imagery analysis for fishery activities [3]. The goal of EM is to monitor the fish catching on fishing vessels, either for scientific survey or regulatory purpose. For example, a fishing vessel may not retain caught fish if the length is below some threshold or exceeding the quota of the vessel for certain species. Therefore, accurate tracking, counting, length measurement and species classification are critically required in the EM systems. There are, however, challenges from the inspected subjects and operation environments [5]. Deformable objects, noise from the wild sea surface, and dynamic background, make conventional tracking, segmentation and classification methods unreliable. The fish can change its appearance quickly during the catching process and be lost during tracking. The deformation and pose variation of the fish can greatly change the visual feature and make species classification challenging. The white spray noise on the sea can result in false alarms. In addition, the dynamic sea water background can cause problems in object segmentation and the inferred depth information. Therefore, in this proposal, we want to build an electronic monitoring system that is robust to the challenges above and can help fish abundance estimation for either regulatory or scientific research purposes.

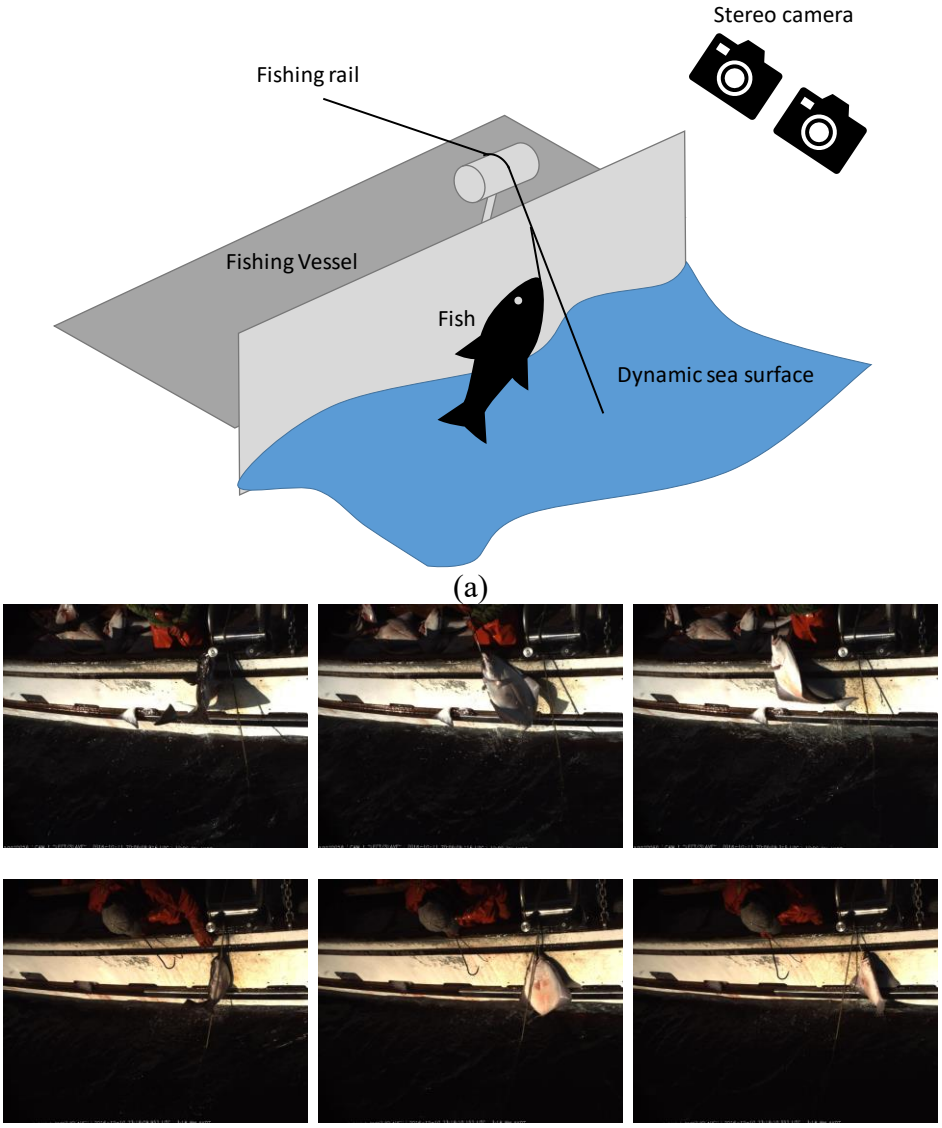


Figure 1.1 (a) Illustration of the electronic monitoring (EM) system for rail fishing and (b) two example video sequences of highly deformable fish being catched on the wild sea surface surrounded by white water spray. From left to right, t, t+2 and t+4 frames.

1.2 Rail-Based Stereo Camera System

To address these needs, we develop the stereo-video based EM system for rail fishing (Figure 1.1), where the fish caught on the fishing rail are pulled up from the sea to the fishing vessel. Occasionally, a fish is discarded and thrown back to the sea, either due to legal reason

or simply because it is not the fish of interest. The goal of our EM system is to track and count the fish caught on the rail, measure their length, identify their species and know whether they are retained or discarded.

With the help of the stereo cameras, we are able to acquire the depth information to facilitate tracking and measurement. The high-resolution stereo cameras capture 6-megapixel images at 10 frames per second (fps), and are connected to a PC for file output and configuration of camera settings, such as exposure and framerate. The cameras are automatically triggered by a pressure sensor, which is connected to the rail. Whenever the fisherman starts catching the fish and the rail is being pulled up, the sensor will trigger the cameras to start recording. Besides, for reliable stereo correspondence matching and depth estimation, the cameras have been calibrated using checkerboard patterns before the fishing vessel goes to the sea for fishing activities.

1.3 Overview of Electrical Monitoring

There are, however, challenges commonly faced from the inspected subjects and operation environments. When the fish caught on the fishing rail being pulled up from the sea, the fish bodies are highly deformable with fast changing color/texture appearance, making tracking extremely difficult (Figure 1.1). Custom tracking methods based on image object detectors can easily fail due to low detection score of the greatly deformed fish, which cannot be fully expected in collecting the training set. Other trackers which learn a target specific detector online, can also lose the track when the fish appearance changes too much in contiguous frames. Besides, the deformed object parts can make the bounding box given by the detector unreliable. In the case of a multibox detector [22, 23], the highest-score bounding box may not always be the best bounding box of an object.

Segmentation and length measurement also face challenges from the noisy and dynamic sea surface environments. Traditional segmentation methods using background subtraction in color image cannot work because the abrupt white-water noise and strong shadows can merge to the foreground. Segmentation methods in RGB-D image by background modeling can also fail due to the dynamic sea surface. Besides, the sea surface consists of curved and piecewise-connected planes, which cannot be fitted well using custom plane detection or segmentation methods, making segmentation of the fish difficult.

In addition to tracking and measurement, one of the primary tasks in the electronic monitoring is identification of the species of fish. However, due to the deformation body of fish, the feature generation and classification needs to be robust with fish in any orientations and poses, which yield diverse visual properties. In addition to the within class variation due to deformation, other challenges include the high visual similarity among fish species. Besides, when dealing with species identification in a finer level, similar species may not be separated well in the learned feature space.

Therefore, in this dissertation, we propose techniques to effectively improve the automatic tracking, length measurement and species classification performance in stereo videos for monitoring of rail fish catching on wild sea surface. For tracking of highly deformable objects in a noisy environment, we propose a tracking system which combines a deep convolutional neural network (D-CNN) image object detector with a Kalman filter in 3D. The Kalman filter is used to rescore the multiple object proposals given by the object detector so as to more reliably track the deformed objects which have low detection scores. To segment the detected and tracked objects, we combine the results from background subtraction and disparity map from stereo matching. The background planes are classified by a clustering-and-scoring

strategy, and the object segmentation is refined using fully connected conditional random fields (CRFs) [36] with color and geometric features. With the segmentation results, we can effectively measure the 3D lengths of objects and update the object positions for tracking. Finally, for fish species classification, we propose a metric learning strategy with temporal constraint to learn the embedding which capture the temporal closeness between frames. Two classification methods are proposed. The first method learns the representative features discriminatively to address the issue of large within class variance and uses the learned features to perform clustering along time domain to reduce noise from samples. The second method uses the semantically-decoupled temporal attention model to perform attention weighted average along time. Besides, the attention groups are learned with a temporal diversity constraint to locate the important frames of certain feature dimensions.

1.4 Contributions

The contributions of this dissertation are summarized as follows.

- Combine single image object detection in 2D and Kalman filter rescoring in 3D for reliable online tracking of highly deformable fish.
- Systematically determined the model weights for multiple features of the tracking system during the training stage.
- Efficiently classify 3D planes using clustering-and-scoring method for dynamic and noisy background.
- Segment the fish in noisy and dynamic environment based on plane detection and fully connected CRFs with color and geometric features.
- Classify fish species through deep metric learning and object classification. Temporal

constraints will help the model to learn a more structured and compact feature space.

- Aggregate feature and remove noise along time domain by learning discriminative representative features.
- Learn the importance of different frames through semantically-decoupled temporal attention model to help fish species classification in video.
- Learn the attention groups in high dimension feature space by applying diversity constraint along time domain.

1.5 Dissertation Organization

The rest of the dissertation is organized as follows.

Chapter 2: overviews of the related work for object tracking in video and segmentation in stereo or RGB-D type of imagery are provided.

Chapter 3: the proposed tracking, segmentation and measurement system are described in detail. Experiments and analysis of each proposed component in the framework are given.

Chapter 4: two proposed species classification methods is presented. The first method is classification by learning representative features. The second method is classification with the help of a semantically-decoupled temporal attention model. Experiments and analysis of the proposed methods are given at the end.

Chapter 5: conclusion of this dissertation is given by summarizing the main contributions.

2 Related Work

2.1 Object Tracking

The classical approaches to online tracking of moving objects in videos can be roughly separated into two categories based on how objects are initialized. One is based on background subtraction [34, 35], from which the objects are initialized by foreground blobs and tracked by features such as kernel histogram or position in images [10, 12-15]. This kind of methods are usually preferred in video surveillance systems based on static cameras due to their efficiency. However, when the environmental noise is too large for consistent background modeling, the results of background subtraction can be unreliable, resulting in tracking failure.

The other category is based on object detection, from which the objects are detected by an image object detector [20-23] and tracked by detection or other local features [11, 16-19]. This kind of methods usually do not assume a static background and thus is suitable for unconstraint videos. However, when the object is deformable and shows an appearance which is rare in the training set, the object detector can fail due to low detection score.

Instead of using an image object detector, some trackers learn a target specific detector online after initialization [17, 38-40]. These trackers learn the features of an object online and can deal with the problem of low detection scores of rare appearances. However, when the appearances of an object change too much in contiguous frames, the detector cannot learn the features of the object well.

In addition to detection, the bounding box is also necessary to localize the object and facilitate tracking. For image object detectors, usually the non-maximum suppression (NMS) is applied with intersection-over-union (IOU) between two candidate bounding boxes, and only the local maximum around an object will be selected when multiple candidate boxes are

available. For the detectors which are target specific, the bounding box with the highest score is selected, and can be improved by a bounding box regression model which is learned in the first frame. However, for an object consists of highly deformable parts, the highest-score bounding box may not always be the best bounding box, because a candidate bounding box may still get reasonably high score even if some parts are ignored, especially in the image object detectors where the offsets of different anchor boxes are predicted independently. Similarly, the bounding box regression model cannot work well because the spatial relationship between each part of object can change too if large deformation occurs during tracking.

Recently, video object detection methods have been applied to address the problem of low detection scores of image object detector. In [16], a Bayesian classifier is used to rescore the tracking results by considering the temporal information. However, their method cannot be applied for online tracking. In [17], a closed-loop method is used to rescore the objectiveness of proposals and improve the speed and quality of image object detector. However, their method only calculates IOU with previous detection results and does not predict the location of the detected object using temporal information.

Therefore, in this dissertation, we combine the results of object detector and Kalman filter to track the objects with low detection scores and find the best bounding boxes of objects.

2.2 Segmentation and Background Plane

Segmentation or scene labeling methods for stereo or RGB-D images have been proposed with stable performance. Most of them are designed for controlled environments, such as indoor scenes or street views [26-29].

One of the popular method to separate a foreground object from a background plane is plane detection. Typical strategies of plane detection include RANSAC, Hough transform-

based methods, and plane growing [30-33]. Once the background plane is found, the foreground object can be instantly acquired by subtracting the background from the input image. Besides, the background plane can further be used as a reference plane for defining features. For example, in [26], the height and angle with respect to the ground plane are used as features for semantic and instance segmentation of indoor scenes. However, when the ground is not an ideal plane or consists of curved surfaces, these strategies cannot be directly applied.

For videos of static scenes, background modeling in RGB-D is a convenient method to find the foreground objects. In [28], background model of an indoor scene is used for segmenting foreground objects. However, when the background is dynamic (e.g., sea surface in our case), the method is not applicable.

Recently, the fully-convolutional networks (FCNs) demonstrate great success in image segmentation tasks. In [29], FCNs is used for semantic segmentation in RGB-D images. However, it required multiple views (>30) to get the final segmentation, which is not suitable in unconstraint and dynamic environments. Besides, to train FCNs or other deep learning based segmentation methods, huge amount of training data with pixelwise labels are needed. Therefore, we do not consider this type of methods in our application.

Due to the limitation of existing methods, we propose a plane clustering strategy by combining superpixel and background subtraction in our proposed segmentation method for dynamic and noisy background.

2.3 Fish Species Classification

Feature selection has always been a critical part in fish species identification or other object classification tasks. Huang et al. [60] uses hand-crafted features such as color, shape and

texture properties from several parts of fish and performs dimension reduction with forward sequential feature selection (FSFS). A balance-enforced optimized tree with reject option (BEOTR) is also proposed to deal with the inter-class similarities in hierarchical classification, and a Gaussian mixture model (GMM) is used to reject low probability samples. Wang et al. [62] combines scale-invariant feature transform (SIFT) feature and shrinking encoding to extract useful feature and perform fine-grained fish recognition. A two-level codebook is proposed to learn the importance of local descriptor. As for unsupervised feature selection, Chuang et al. [61] proposes an unsupervised approach for learning object parts based on saliency and relaxation labeling. A non-rigid part model is learned based on fitness, separation and discrimination of each object part, and an unsupervised clustering approach is used to generate a binary class hierarchy.

Recently, the deep convolutional neural networks (CNNs) have also drawn increasing attention due to their great success in image object classification and other computer vision tasks. The AlexNet proposed by Krizhevsky et al. [42] is one of the pioneer work of using CNNs to perform image object classification tasks. Simonyan et al. [43] proposes the VGGNet which has deeper structure than AlexNet and achieve better performance. He et al. [44] introduces a residual structure which learns the residual function with reference to the inputs and proposed the ResNet, which is deeper than the VGGNet and performs better in the object classification tasks but has lower complexity. As for the application of CNNs in fish species classification, Siddiqui et al. [63] proposes a cross-layer pooling algorithm for feature extraction using pre-trained CNN, and uses support vector machine (SVM) to perform classification on the extracted features. Kratzert et al. [64] improved classification performance of CNN by combining additional meta-information (date of migration and fish length).

In addition to using a CNN which directly predict the object class, another approach is using a CNN to perform deep metric learning and acquire a feature embedding, which can further be used for classification, verification or clustering tasks. Schroff et al. [49] proposes a training method which directly optimize the feature embedding based on a triplet (anchor, positive example, negative example) loss function. In each training batch, the distance from the anchor to the positive example is minimized and the distance from the anchor to the negative example is maximized. The learned feature embedding shows great performance in the face verification tasks. Song et al. [50] proposes a lifted structured loss function based on all positive and negative pairs within a training batch, and shows improved performance on clustering and classification tasks of bird and car data. Rippel et al. [51] explicitly represent the distributions of the clusters of different classes in the feature space and proposes a magnet loss function which penalizes their overlaps. Ding et al [52] proposes the Mean Distance Regularized Triplet Loss which takes into account the clustering properties and alleviates the problem of non-uniform intra- and inter-class distance distributions. Other variation of triplet loss training [53-58] also show improved performance in the classification or verification tasks. In addition to feature embedding for static image, Misra et al. [59] proposes a temporal order verification framework which trains the CNN to verify the order of an image tuple. The feature embedding learned by the CNN shows the metric property that captures the temporal distance between samples.

Therefore, the proposed work handles the difficulties of visual similarity among fish species using deep metric learning. Also, temporal information is used to help the model to capture the temporal closeness and improve the classification performance over temporal sequences of fish images.

2.4 Temporal Attention for Video Data

Recently, the temporal attention strategy has drawn increasing attention in the video data classification or reidentification tasks [65-73]. By taking account of frame-level features from multiple contiguous frames at once, the temporal attention model learns to predict the temporal attention weight, i.e., the importance of each frame, to help generating more representative clip-level feature. For video-based person reidentification tasks, Gao et al. [65] proposes a temporal convolution layer which takes mid-level CNN features from multiple frames and predict their temporal attention weights. The weights are used to calculate the temporal weighted average of the learned frame-level feature as the clip-level feature. Li et al. [66] proposes a diversity regularized spatiotemporal attention model to learn the importance of different human body parts from sequence of images. Spatial constraint is applied to force different attention modules to focus on different image regions. Wu et al. [67] proposes a spatial gated recurrent unit which takes the attention weights from each frame and generate clip-level feature in an iterative manner. For video classification tasks, Song et al. [68] uses a temporal-spatial mapping to locate informative features from multiple frames and image regions. The input video is then represented as the weighted average of the mapped features. Long et al. [70] proposes the multimodal attention in which different data modalities have their own attention module, and the output of all attention modules are concatenated together as the video-level feature.

However, the existing temporal attention models usually predict only one attention weight for all feature dimensions, which may fail if different feature dimensions show importance in different frames. Therefore, in this dissertation, we propose a semantically-decoupled temporal attention model, which learns multiple temporal groups for different feature dimensions, to

solve this problem.

3 Tracking, Segmentation and Measurement

3.1 System Overview

As shown in Figure 3.1, the proposed rail-fishing tracking system consists of an image object detector, which gives multiple 2D object proposals in terms of bounding boxes with different detection scores, and a Kalman filter-rescoring module in 3D. Besides, with the refined object segmentation result of RGB-D image derived from stereo images, the location of the object is further updated for better tracking performance.

3.2 Tracking

3.2.1 Object Proposal

The object proposals can come from any image object detector. For a single image object detection task, the objects can be localized by performing non-maximum suppression (NMS) on all object proposals so that only the proposals which are local maxima are preserved. However, in our case, because an actual object may not always give high detection score during tracking due to object deformation, making the maximum-

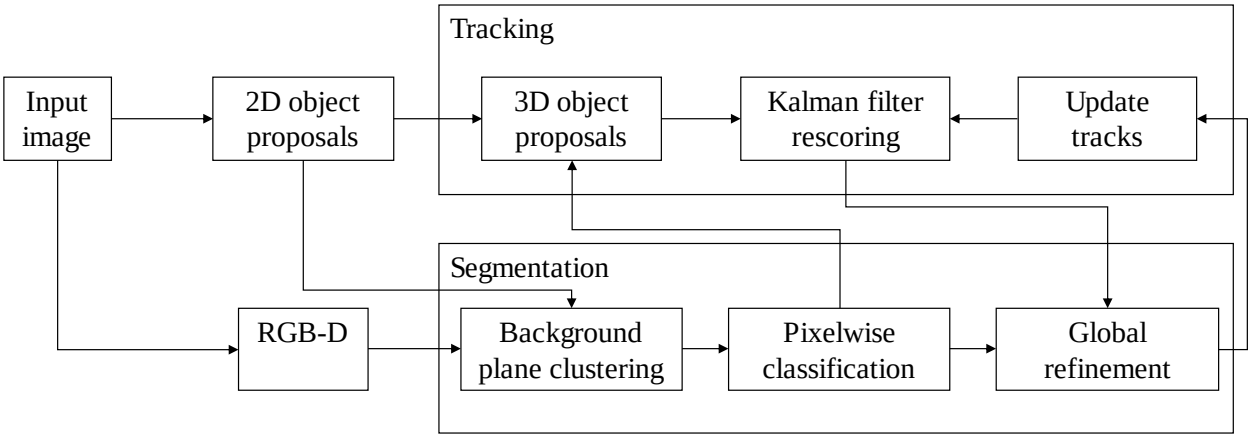


Figure 3.1 Tracking and segmentation system overview.

score detection deviate from the true location of object, we thus preserve all the detections with scores higher than a threshold τ_1 as object proposals. With the foreground segmentation in RGB-D (Section 3.3) without object-wise refinement, we project the foreground pixels of each 2D object proposal to 3D and calculate the position (X, Y, Z) and size (W, H, D) of the 3D object proposal (see Figure 3.2). The 3D object proposals are then passed to a Kalman filter-rescoring process and associated with objects in previous frames.

3.2.2 Proposal Rescoring

We use Kalman filter to track the objects and predict the locations in 3D. Intuitively, if an object proposal is close to the prediction, it is more likely to be correct even if it has a lower detection score, which can be due to object deformation. Therefore, we propose a rescoring strategy to consider the temporal information in addition to the detection scores.

To rescore the object proposals and associate them with current tracks from the previous frame, we first use a Kalman filtering tracking procedure similar to [13] but in 3D space to predict the objects' position and size. The state vector used is $[X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}, W, H, D, \dot{W}, \dot{H}, \dot{D}]$ and the measurement vector is $[X, Y, Z, W, H, D]$, where

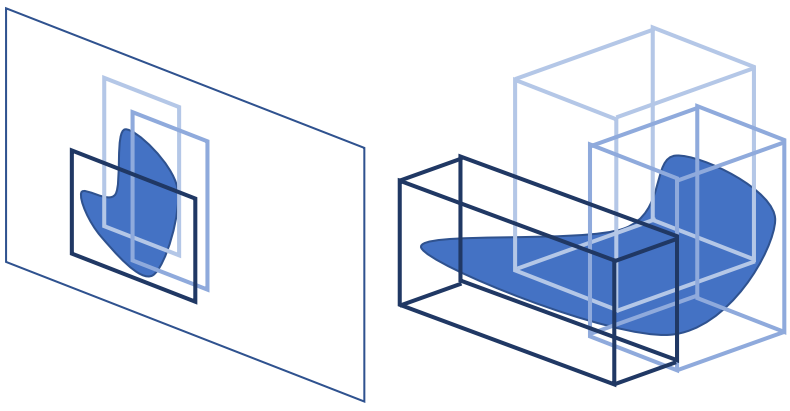


Figure 3.2 Project 2D object proposals to 3D object proposals using foreground segmentation in RGB-D (Section 3.3).

(X, Y, Z), $(\dot{X}, \dot{Y}, \dot{Z})$, (W, H, D) and $(\dot{W}, \dot{H}, \dot{D})$ denote the object position, velocity, size, and size changing rate, respectively. We introduce the size changing rate to approximate the deformation process of the object.

Given an object proposal $\boldsymbol{X} = [X, Y, Z]$ and $\boldsymbol{S} = [W, H, D]$, we would like to find its distance to the Kalman filter prediction $\boldsymbol{X}^* = [X^*, Y^*, Z^*]$ and $\boldsymbol{S}^* = [W^*, H^*, D^*]$. The features we use are: position, size, diagonal length because they are more consistent for the freely moving/deforming objects. We use Gaussian kernels for the distances and define the tracking score as:

$$
\begin{aligned}
s_{track} &= 1 - \lambda_{pos} d_{pos} - \lambda_{size} d_{size} - \lambda_{diag} d_{diag}, \\
d_{pos} &= 1 - \exp\left(- \frac{\|\boldsymbol{X} - \boldsymbol{X}^*\|_2^2 / \|\Delta \boldsymbol{X}^*\|_2^2}{2\sigma_{pos}^2} \right), \\
d_{size} &= 1 - \exp\left(- \frac{\|\boldsymbol{S} - \boldsymbol{S}^*\|_2^2 / \|\boldsymbol{S}^*\|_2^2}{2\sigma_{size}^2} \right), \\
d_{diag} &= 1 - \exp\left(- \frac{(\|\boldsymbol{S}\|_2^2 - \|\boldsymbol{S}^*\|_2^2)^2 / \|\boldsymbol{S}^*\|_2^2}{2\sigma_{diag}^2} \right),
\end{aligned}
\tag{3.1}
$$

where $\Delta \boldsymbol{X}^*$is the object movement from the previous frame to $\boldsymbol{X}^*$, and $(\lambda_{pos}, \lambda_{size}, \lambda_{diag})$ are the weighting constants for position/size/diagonal length distance $(d_{pos}, d_{size}, d_{diag})$.

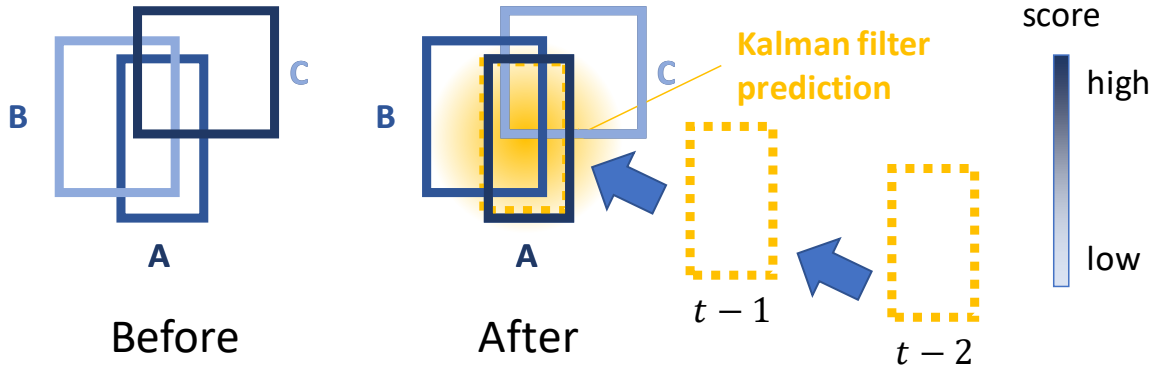


Figure 3.3 Rescoring object proposals using Kalman filter and tracking scores. The scores of object proposals closer to prediction are increased. (Illustrate in 2D for clearness.)

The parameters σ_{pos}, σ_{size} and σ_{diag} control the degrees of distances. The rescoring process is illustrated in Figure 3.3. Note that the position difference is normalized by the object movement $\Delta \boldsymbol{X}^*$ because the prediction error of position is usually proportional to the object velocity. Similarly, the size and diagonal length differences are normalized by $\boldsymbol{S}^*$.

By combining the detection score with tracking score, we can now rescore the object proposals as follow:

$$s_{total} = s_{detection} + s_{track}, \tag{3.2}$$

where $s_{detection}$ is the detection confidence.

3.2.3 Tracking Criteria

After the object proposals are rescored, we associate current tracks with the proposals by greedily matching the highest score above a threshold [37]. On the other hand, if a track cannot be associated with any object proposal, we assume it is temporarily missing and treat the prediction from the Kalman filter as the observed position and size. If a track keeps missing for several frames (3 in the experiments), we then stop tracking. After all tracks are tracked or handled, we may apply NMS with IOU threshold 0.5 to all unmatched proposals, and only initialize those with a high detection score τ_2 as new tracks. Finally, the bounding boxes of each track are further updated through the segmentation process with object-wise refinement (Section 3.3). Besides, for the new tracks, because it is possible that objects enter the scene with low detection scores, we also track them backward in time to recover their past trajectories.

3.2.4 Learning Parameters and Weight Constants

The parameters used in the proposal rescoring can be systematically learned from the

training data. To decide σ_{pos}, σ_{size} and σ_{diag}, we take the root mean square of the errors of the Kalman filters in position, size and diagonal length accordingly from the training data. More specifically, in each frame we treat the ground truth bounding boxes as measurements and calculate the error of predictions. The $\{\sigma$'s$\}$ are then set as the root mean square of errors in all frames.

For λ_{pos}, λ_{size} and λ_{diag}, we hope that in each frame the best object proposal gets the highest score in (3.2) (best object proposal means the largest IOU with the labeled bounding box in 2D). Therefore, we formulate the tracking problem as a classification problem, where the best object proposals are positive examples and the rest are negative examples. In practice, from all the frames in the training data, we select the object proposals whose IOU with labeled bounding boxes larger than 0.8 as positive examples. By using the parameters σ_{pos}, σ_{size} and σ_{diag} learned above, for an object proposal i, (3.2) becomes a linear equation of λ_{pos}, λ_{size} and λ_{diag}:

$$s(i) = s_{detection}(i) + c_0 + \lambda_{pos} c_{1,i} + \lambda_{size} c_{2,i} + \lambda_{diag} c_{3,i}, \tag{3.3}$$

where c_0, $c_{1,i}$, $c_{2,i}$ and $c_{3,i}$ are constants. Therefore, we can find the best of λ_{pos}, λ_{size} and λ_{diag} by solving a linear support vector machine (SVM).

3.2.5 Experimental Result

Our proposed algorithm adopts two different state-of-the-art object detectors: SSD [23] and YOLOv2 [22]. The training of the detectors is based on 10000 labeled frames from the video sequences, in which 80% are used as the training set and the rest 20% as the validation set. To make the model more robust to different object size and environment changes, we augment the data by randomly applying: horizontal flip, random crop, expansion and color distortion, and train the model for 15 epochs. Then, the parameters in

Table 3-1 Tracking Performance

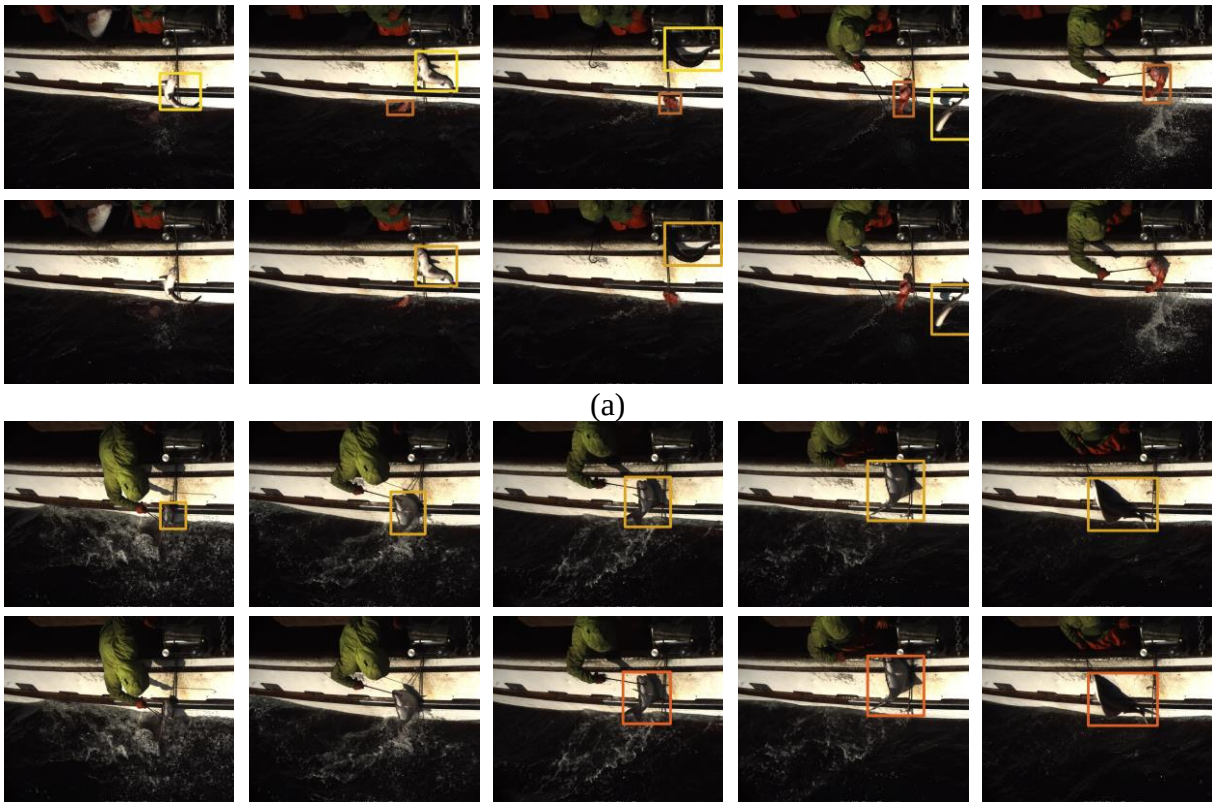
	Clip A1 (GT 112)			**Clip A2 (GT 131)**			**Clip A3 (GT 105)**		
Method	***TP***	***FN***	***FP***	***TP***	***FN***	***FP***	***TP***	***FN***	***FP***
baseline SSD	86	26	12	90	41	16	72	33	18
FCNT[17] SSD	92	20	7	101	30	12	78	27	14
CFNet[38] SSD	90	22	7	94	37	13	80	25	13
rescoring SSD	100	12	5	120	11	6	89	16	8
res.+seg. SSD	**106**	**6**	**3**	**124**	**7**	**2**	**99**	**6**	**4**
baseline YOLOv2	80	32	11	84	47	14	70	35	16
rescoring YOLOv2	96	16	3	115	16	5	85	20	7
Method	**Clip B1 (GT 30)**			**Clip B2 (GT 65)**			**Clip B3 (GT 43)**		
baseline SSD	19	11	6	43	22	7	25	18	13
FCNT[17] SSD	23	7	5	55	10	3	31	12	5
CFNet[38] SSD	20	10	6	49	16	5	29	14	5
rescoring SSD	28	2	4	60	5	2	38	5	4
res.+seg. SSD	**30**	**0**	**1**	**63**	**2**	**1**	**41**	**2**	**2**
baseline YOLOv2	19	11	5	40	25	8	26	17	11
rescoring YOLOv2	26	4	3	58	7	2	39	4	5

object proposal rescoring are learned from the object proposals in another 3000 frames given by the trained object detector. For SSD, we set the thresholds (τ_1, τ_2) of detection score as (0.3, 0.7), and for YOLOv2 as (0.2, 0.6).

The proposed method is tested on 6 video clips from two vessels, A and B, at different time of day: A1/B1 (early morning), A2/B2 (morning) and A3/B3 (afternoon/strong sunlight). All video clips from A are 20 minutes at 10 fps and B are 10 minutes at 16fps.

The baseline method treats the detection of each frame as single image object detection, i.e., simply perform NMS for detection scores larger than a threshold (0.6 for SSD and 0.5 for YOLOv2), and tracking using a Kalman filter. The second method is FCNT [17], which is a CNN based tracker that predicts target heat map with one CNN captureing the category information and the other discriminating targets from background. The third is CFNet [38], which is a correlation filter based tracker that uses CNN feature for

correlation operation and learns the feature end-to-end. In our experiments, both FCNT and CFNet use detection results from SSD to initialize tracks. The fourth and fifth methods are the proposed tracking method without and with segmentation, respectively. To evaluate the tracking performance, we consider an object is successfully tracked if 80% of its trajectory is recovered with IOU > 0.5 with ground truth bounding box.



(a)

(b)

Figure 3.4 Tracking results. Video sequence (a) and (b), Top row) proposed method, and bottom row) baseline method. The proposed method can successfully track the highly deformed fish with low detection scores.

The result in Table 3-1 shows the proposed tracking method greatly improves the number of true positive (TP) and false negative (FN) tracks for both SSD and YOLOv2 detectors, while FCNT and CFNet still get a lower TP counts, because the features of the fish changes quickly due to deformation and appearance changes. Note that, when the refined segmentation is used to update the object location, the number of false positive (FP)

tracks is decreased due to better removal of noise such as white spray or sharp shadows using the disparity information. Sample results in Figure 3.4 show that the proposed tracking method can successfully track deformed fish with low detection scores.

3.2.6 Discussion

Tracking Score vs. Detection Score

The major challenge in tracking the highly deformable fish is the low detection score. Therefore, we analyze the distribution of the minimum detection score of tracks in Figure 3.5. Most of the tracks have a minimum detection score lower than 0.55, which means that if we use a hard threshold 0.55 for detection, we will lose them during tracking. However, if we consider the tracking score at those minimum-detection-score frames, the histogram shows that even when the detection scores are low, the tracking scores are still high enough for successful tracking.

Proposal Rescoring

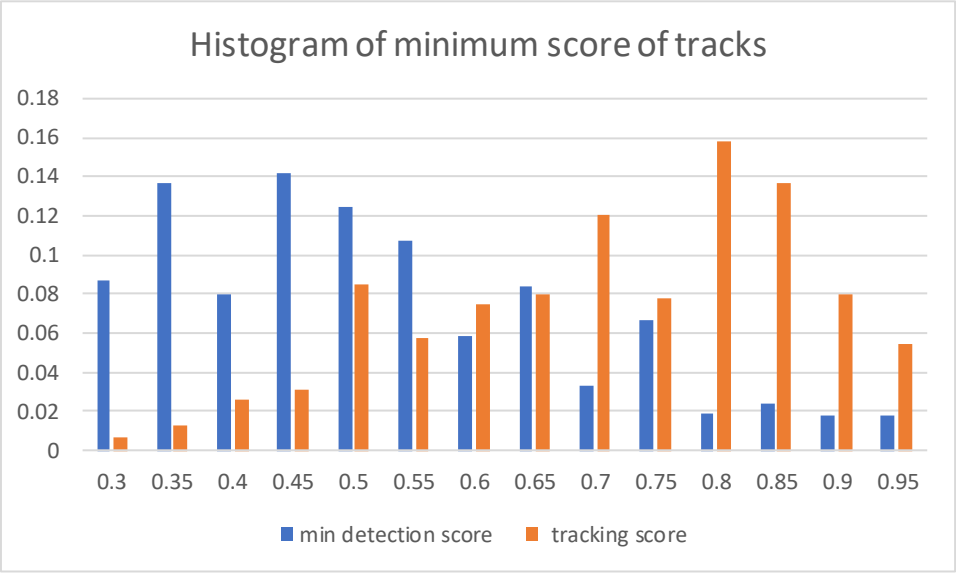


Figure 3.5 Histogram of detection/tracking score at the minimum-detection-score frames of tracks. Most of the tracks have a minimum detection score lower than 0.55, while the tracking scores at those frames are still reliable.

Another problem of detecting highly deformable fish is that the best object proposal

may not always get the highest detection score, but a bad object proposal could still get reasonable score even if some deformed object parts are ignored. Therefore, we analyze how well the rescoring strategy help us to assign a higher score to the best object proposal. For each image, we sort all object proposals based on their detection scores to see what the rank of the best object proposal is. In the ideal case, the rank should be 1. The histogram of the rank is shown in Figure 3.6. Before rescoring, if we simply select the object proposals with highest detection score, only 18% of them are the best object proposals. However, after rescoring is applied, the best object proposals are more likely to get a higher ranking, enabling more accurate tracking and length measurement.

3.3 Segmentation

In addition to tracking fish objects, we also want to measure the 3D lengths of the objects, which require us to do segmentation on the RGB-D images derived from stereo videos. Therefore, in this section, we propose a reliable procedure for the segmentation with dynamic

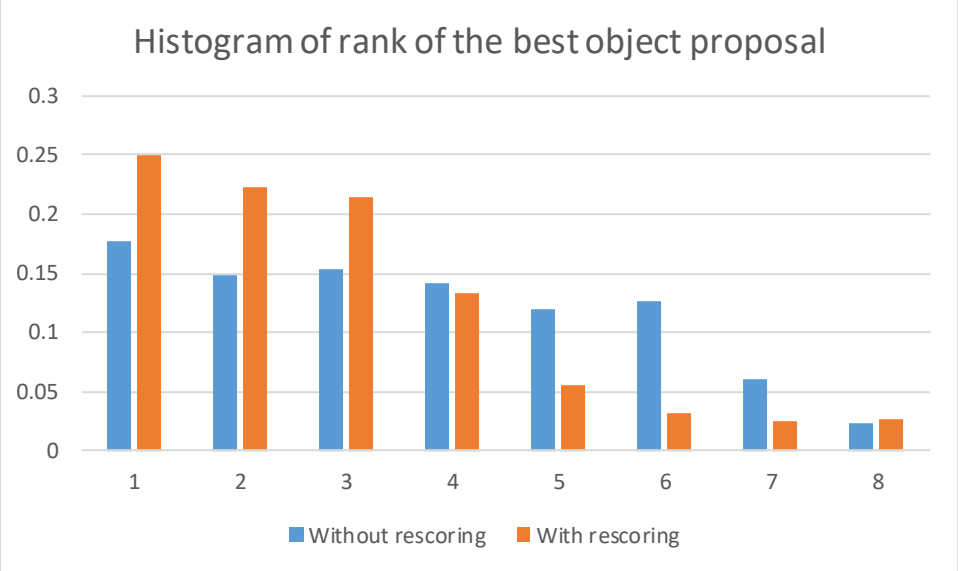


Figure 3.6 Histogram of the rank in score of the best object proposals. When rescoring strategy is applied, the ranks of the best object proposals are higher.

and noisy background in the RGB-D images.

We consider the segmentation problem as a labeling problem. Given the bounding boxes $Box = \{box_i | i = 1, \dots, K\}$ of K objects from the tracking results, we would like to label all the N pixels $\{p_1, \dots, p_N\}$ with the label set $L = \{l_0, l_1, \dots, l_K\}$, where the extra label l_0 stands for background. Due to the highly deformable character of the objects, rather than modeling the object pixels, we model and remove the background pixels from the objects' bounding boxes.

The segmentation process consists of three steps: background plane clustering, pixelwise classification and global refinement.

3.3.1 Background Plane Clustering

We would like to cluster the noisy, dynamic and non-smooth background pixels into several background planes. Because the edges in depth map usually align with the surface/object boundaries while the edges in color image occur on both boundaries and planar areas, we start with the superpixels acquired using color features only and cluster them using geometric features. The superpixels are generated from an initial background area, which consists of the pixels outside the object bounding boxes and classified as background in background subtraction. The purpose of using background subtraction is to remove other irrelevant moving objects which are not detected by the object detector from the background plane.

Given N_s superpixels $\{s_i | i = 1, \dots, N_S\}$, for each superpixels s_i we use RANSAC to find the 3D plane $\boldsymbol{P}_i$ and use the 3D location $\boldsymbol{X}_i$ and surface normal $\boldsymbol{N}_i$ as the geometric feature for clustering. We define the distance between two 3D plane as:

$$
\begin{aligned}
d(s_i, s_j) &= \|\boldsymbol{d}_{loc}\|_2 + \alpha_{normal}\|\boldsymbol{d}_{normal}\|_2 + \alpha_{fit}|d_{fit}|, \\
\boldsymbol{d}_{loc} &= \boldsymbol{X}_i - \boldsymbol{X}_j, \\
\boldsymbol{d}_{normal} &= \boldsymbol{N}_j - (\boldsymbol{N}_i \cdot \boldsymbol{N}_j)\boldsymbol{N}_i, \\
d_{fit} &= d(\boldsymbol{P}_i, \boldsymbol{X}_j),
\end{aligned} \tag{3.4}
$$

where α_{normal} and α_{fit} are weighting constants. The first term and second term represent the difference in location and surface normal respectively. The Third term is the distance of fitting superpixels s_j to the plane of s_i and therefore making the distance asymmetric, i.e., $d(s_i, s_j) \neq d(s_j, s_i)$.

Then, we adopt the simple linear iterative clustering (SLIC) [41] strategy in superpixel segmentation for its efficiency. During initialization, N_p of cluster $\{C_k = [\boldsymbol{X}_k, \boldsymbol{N}_k, x_k, y_k]^T | k = 1, \dots, N_p\}$ is sampled from the superpixels which are closest to the regular grid, where the grid interval $S = \sqrt{N/N_p}$. The 2D pixel location (x_k, y_k) is used to facilitate the searching process at each iteration. Then, at each iteration, each superpixel s_i is assigned to the nearest cluster within a search range $2S \times 2S$, and the cluster center is updated as the average of its members. The process is repeated until the deviation changes of cluster centers is below a threshold or the maximum number of iteration is reached.

After the clustering is finished, we use RANSAC to remove outliers and re-estimate the 3D plane of each cluster. The probability distributions of the geometric features of each cluster are then modelled using Gaussian distributions: $\boldsymbol{d}_{normal} \sim \mathcal{G}(0, \boldsymbol{\Sigma}_{normal})$, and $d_{fit} \sim \mathcal{G}(0, \sigma_{fit})$. Note that we do not model $\boldsymbol{d}_{loc}$ because we assume the plane can be extended to other regions, such as inside of object bounding boxes, which do not belong to the initial background area. The full process is summarized in Algorithm 3.1.

Algorithm 3.1 Background Plane Clustering

	Input: N_s superpixels $\{s_i \vert i = 1, \dots, N_s\}$
	Output: N_p clusters $\{C_k \vert k = 1, \dots, N_p\}$
1:	Initialize cluster center $C_k = [\boldsymbol{X}_k, \boldsymbol{N}_k, x_k, y_k]^T$ by sampling superpixels which are closest to the regular grid of step S
2:	**repeat**
3:	**for** each cluster center C_k **do**
4:	**for** each superpixel s_i in search region $2S \times 2S$ around C_k **do**
5:	Compute the distance $d(C_k, s_i)$
6:	**if** $d(C_k, s_i) <$ s_i's previous assignment distance
7:	assign s_i to C_k
8:	**end if**
9:	Update cluster centers
10:	**until** converge
11:	Calculate distribution within each cluster

3.3.2 Pixelwise Classification

Given a pixel p_m, we can now calculate the likelihood of belonging to the cluster C_k as $\mathcal{L}(C_k|p_m) = P(p_m|C_k)$. To aggregate the likelihoods from all the clusters to get a better estimation, we define the background plane score of a pixel p_m inside the bounding boxes as the weighted sum of likelihoods:

$$BP_m = \sum_{k=1}^{N_p} \mathcal{L}(C_k|p_m)\, A_k \exp(-\|\boldsymbol{X}_m - \boldsymbol{X}_k\|_2^2/\sigma_S^2)\,, \tag{3.5}$$

where A_k is the area of the cluster and σ_S is a scaling constant for the distance between the pixel and cluster. Intuitively, the higher the background plane score, the more likely the pixel belongs to background. On the contrary, when the background plane score is close to 0, the pixel should be foreground. Therefore, we define the probability of foreground as:

$$P_{fg} = \exp(-\alpha BP_m)\,. \tag{3.6}$$

Note that α is an adaptive scale satisfying that:

$$\exp(-\alpha BP_{otsu}) = 0.5, \tag{3.7}$$

where BP_{otsu} is the Otsu's threshold that best separates the background plane score inside the bounding boxes. Then, we have the probability of background $P_{bg} = 1 - P_{fg}$. Besides, the probability of p_m being the i 'th object can be derived as $P(l_j) = P_{fg}/|\{box_j | p_m \in box_j\}|$, assuming that if the pixel belongs to more than one bounding box, it has equal probabilities of belonging to each object. Finally, for pixels outside the bounding boxes, we simply assign $P_{bg} = 1$.

Using this strategy, we can find a better guess of the background area given the noisy segmentation result from background subtraction of color image. For example, the shadow or the white spray on the sea surfaces will get a higher background plane score even if they are regarded as foreground in color images; while the fish, which is partially connected to or above the background planes, will get a lower score even if it is missed in initial background subtraction.

3.3.3 Global Refinement

To globally refine the pixelwise classification and make the segmentation adhere to color and geometrical features, we incorporate the fully connected CRFs [36] to our system. The fully connected CRF is characterized by the Gibbs energy function:

$$E(\mathbf{y}) = \sum_m \psi_u(y_m) + \sum_{m<n} \psi_p(y_m, y_n), \tag{3.8}$$

where $\mathbf{y}$ is the labeling for all pixels $\{p_1, \dots, p_N\}$.

The unary potential ψ_u is defined as $\psi_u(y_m) = -\log P(y_m)$, where $P(y_m)$ is the

probability of labeling acquired from the background plane score in the previous section. The pairwise potential is defined as the weighted Gaussian kernels to enable efficient inference:

$$\psi_p(y_m, y_n) = \mu(y_m, y_n)\begin{bmatrix} w_1 \exp\left(-\frac{\|p_m - p_n\|_2^2}{2\theta_{\alpha_1}^2} - \frac{\|I_m - I_n\|_2^2}{2\theta_{\beta_1}^2} - \frac{\|\boldsymbol{N}_m - \boldsymbol{N}_n\|_2^2}{2\theta_{normal}^2}\right) \\ +w_2 \exp\left(-\frac{\|p_m - p_n\|_2^2}{2\theta_{\alpha_2}^2} - \frac{\|I_m - I_n\|_2^2}{2\theta_{\beta_2}^2}\right) \\ +w_3 \exp\left(-\frac{\|p_m - p_n\|_2^2}{2\theta_{\gamma}^2}\right) \end{bmatrix}. \quad (3.9)$$

The first term empirically weighted by w_1 is the appearance kernel based on pixel positions, colors and geometric features (surface normal), and the second term weighted by w_2 is the appearance kernel without geometric features to avoid over-fitting to noisy geometric boundary. The third term weighted by w_3 is the smoothness kernel based on pixel positions only. The degree of similarity of features are controlled by parameters θ_α, θ_β and θ_{normal}. The compatibility function μ is defined as $\mu(y_m, y_n) = 1$ if $y_m \neq y_n$, and 0 otherwise.

After the refinement step, we do the speckle check and connected component analysis to ensure that the object is in a reasonable shape in the 3D space. If two object masks touch, we assume the farther object is partially occluded and extend its 3D bounding box in the occluded direction using the predicted size. Besides, if an object mask becomes empty after refinement, we consider it is as an invalid track due to false alarm (shadow or water noise) of the detector and stop tracking it. Finally, we use the segmentation results to update the bounding boxes in 3D of each track to further improve the tracking performance.

3.3.4 Experimental Result

For acquiring 3D information from stereo image pairs, we use the slanted plane smoothing stereo matching [24] to find the disparity map and remove outliers and

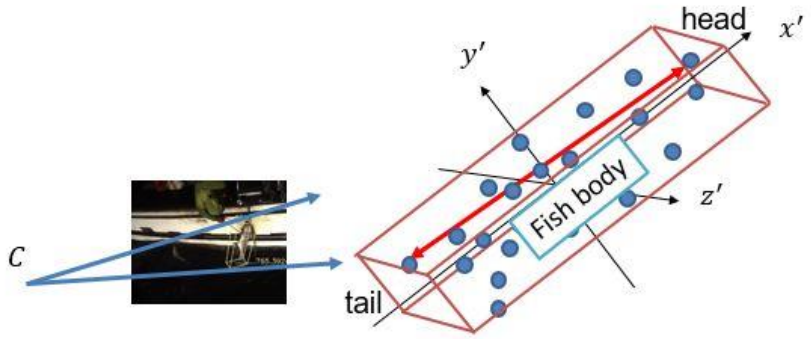


Figure 3.7 Rotated 3D bounding box derived from PCA is used to estimate the length of fish.

fish.

foreground fattening effect using weighted mode filtering [25]. The disparity map is converted to depth using pre-calibrated camera parameters. For background plane clustering, we start with N_s=1600 superpixels and cluster them into N_p=32 clusters.

The measurement in 3D is done by backprojecting the object pixels into 3D space. The orientation of the object (x', y', z') is found by performing principle component analysis (PCA) on the 3D coordinates of all the backprojected pixels, and the length is measured as the distance between the two endpoints along the major axis given by PCA (Figure 3.7).

To estimate the actual length of a fish based on segmentation and measurement during tracking, we only select the frames when the fish is above the water surface for sake of reliability. Among the lengths in these frames, we select those within 90% of the maximum length and take their average as the length estimation because the shorter ones are usually due to curved fish body and the actual lengths are underestimated.

The performances of length measurements of several methods when compared with the ground truth is given in Table 3-2. All results are based on the proposed tracking method with SSD, which achieves the best tracking performance. The background planes clustering

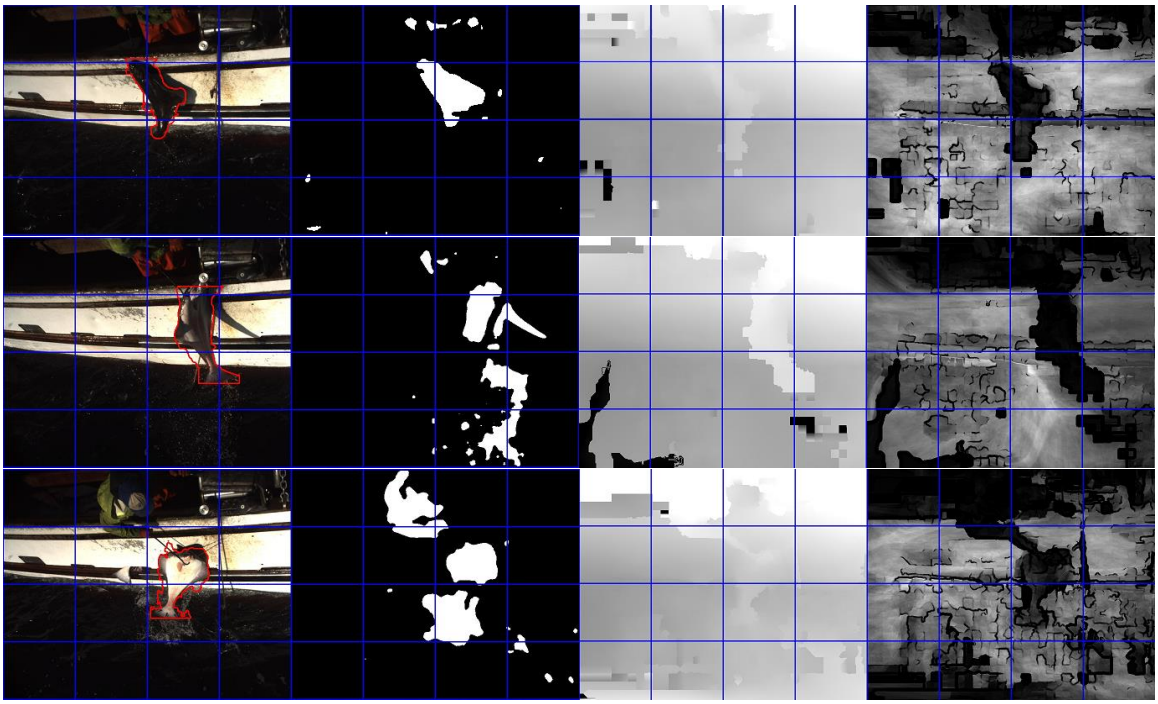


Figure 3.8 From left to right: segmentation of objects, original results of background subtraction, disparity map, and background plane score. The strong shadow and noise on sea surface are distinguished from fish.

Table 3-2 Mean Absolute Error of Length

Method	**Clip A1 (21)**	**Clip A2 (36)**	**Clip A3 (21)**
w/o bg. planes	13.8%	16.5%	18.2%
RANSAC*	12.1%	13.7%	13.8%
with bg. planes	**5.4%**	**4.3%**	**6.0%**

*For RANSAC we assume there are two background planes: sea surface and ship

**Only the fish captured and retained (not released) in vessel A have ground truth length

greatly reduces the mean absolute error due to better segmentation, while using background plane from RANSAC does not give much improvement. Besides, in video clip A3 where strong shadow exists, the proposed method greatly reduces the error by removing the shadow from object segmentation. Figure 3.8 shows examples of background plane score maps, in which the shadow or noise on the sea surface are distinguished from fish.

3.3.5 Discussion

Threshold of Background Plane Score

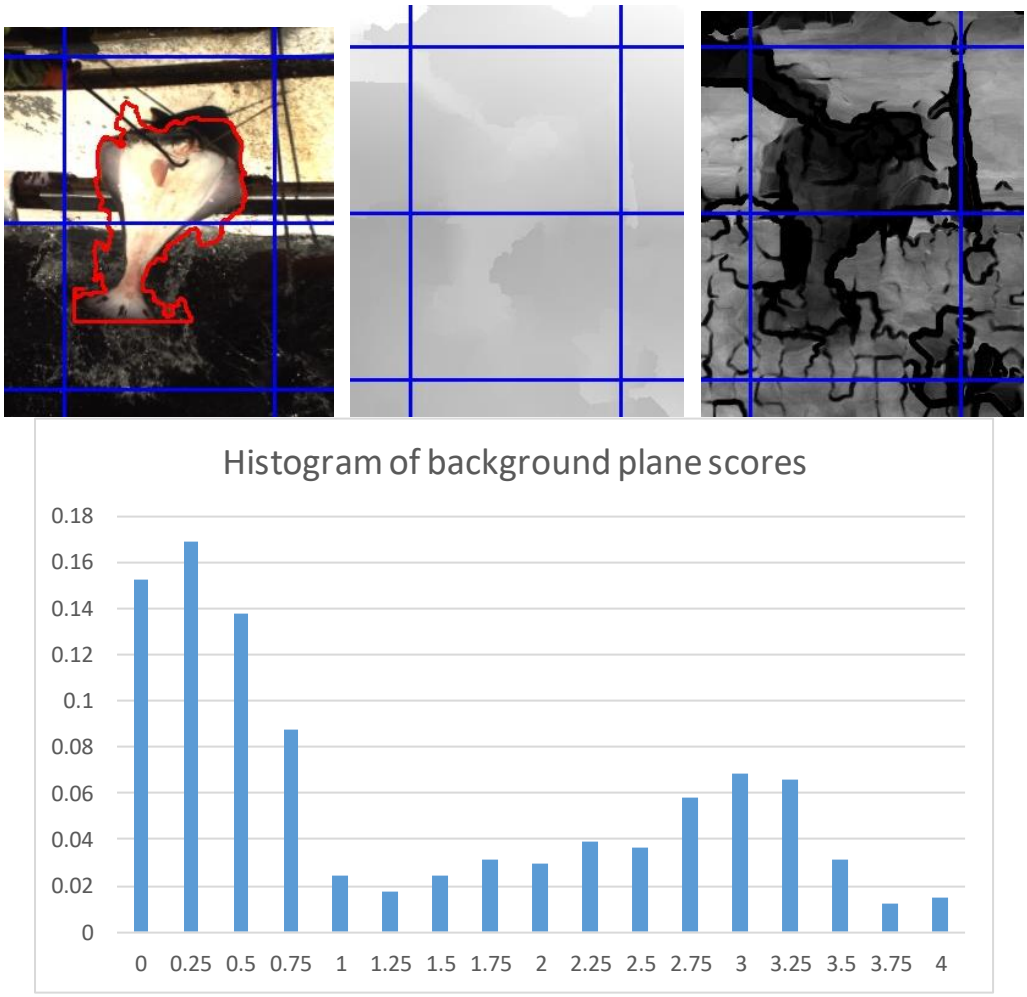


Figure 3.9 Histogram of background plane score inside object bounding box. The foreground pixels get a much lower score, while the background pixels get larger scores and with wider distribution.

Given background plane score map, we use Otsu's threshold to separate foreground and background. Figure 3.9 shows an example image and the histogram of background plane score inside the object bounding box. It shows that the foreground pixels get a much lower (<1) background plane score, while the background pixels get larger background plane scores, which means that the background pixels are well fitted by the background plane clusters. Besides, the background pixels have a wider distribution in background plane score because the smoothness of the sea surface are different at different locations. The smoother the sea surface, the more and better the background pixel can fit to the

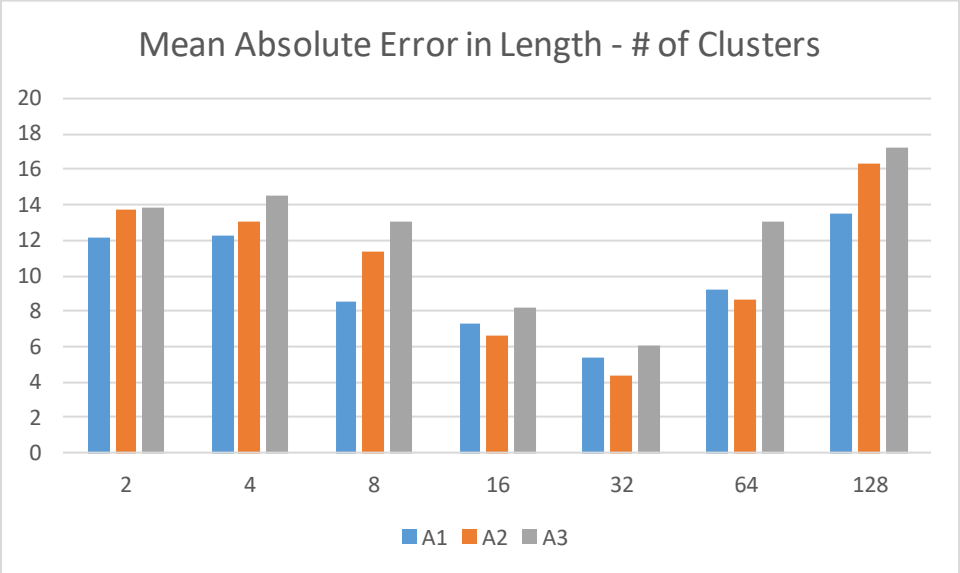


Figure 3.10 Effect of number of clusters on length measurement. When the number of cluster > 32, the clusters become overfitted and the error in length increase.

clusters, and the higher the background plane score.

Effect of Number of Clusters

We use background plane clusters to model the curved sea surfaces. Intuitively, the rougher the surfaces are, the more cluster we need to model the surface. However, the more the clusters, the smaller each cluster is and the less superpixel segmented regions it contains. When the cluster area becomes too small, it can overfit to local region and lose the long-range information. Therefore, we analyze the effect of number of clusters on length measurement (see Figure 3.10). When there are only two clusters, the performance is similar to using RANSAC to fit the two major surfaces, i.e., ship and sea surface. When we use more clusters to model the background planes, the error in length decreases, implying better segmentation. However, when the number of cluster is 64 or more, the performance decreases due to overfitting, since each cluster only contains 25 or less superpixels.

3.4 Length Measurement

3.4.1 3D Midline

To measure the fish length more accurately when the fish body is curved, we find the 3D midline of the fish body based on the back-projected point cloud of fish body in 3D. We first equally separate the of fish body into H bins along the major axis given by PCA. For each bin, we find the geometric center of the point cloud. Then, we connect the center points of each bin to form the 3D midline and measure the midline length of fish (Figure 3.11).

3.4.2 Length from Multiple Frames

To estimate the actual fish length from multiple frames, we assume the observed fish length is a random variable whose probability density function consists of mixture of Gaussian functions, and find the true length using expectation-maximization (EM) algorithm. Because of the curved fish body and self-occlusions, in most cases our observed

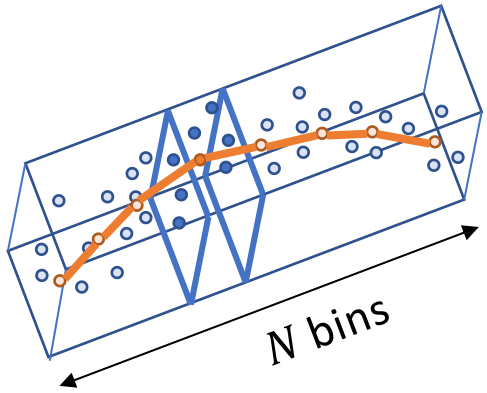


Figure 3.11 3D midline is acquired by connecting geometric center of each bin along the major axis.

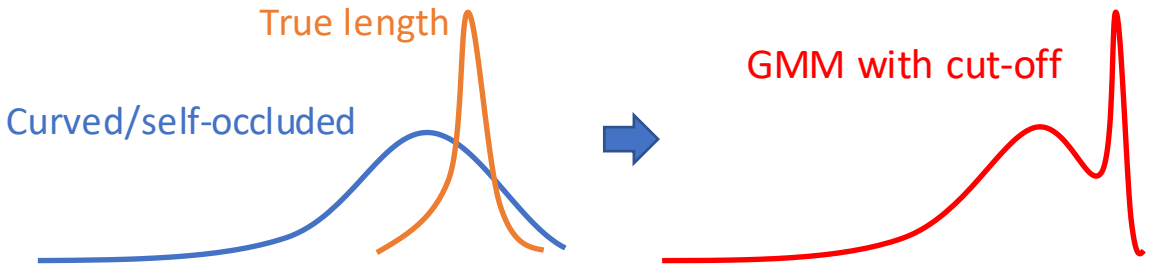


Figure 3.12 Gaussian distribution with cut-off.

fish lengths are less than the true fish length. Besides, it is less likely that the length of the midline we get is much larger than the true fish length, unless it is an outlier (possibly other connected objects, e.g. hands or tools). Therefore, we assume the observed fish length as the minimum of a set of Gaussian random variables:

$$X = \min_i X_i \,, X_i \sim \mathcal{N}(\mu_i, \sigma_i) \forall i \tag{3.10}$$

and the true fish length would be $\max_i \mu_i$. The probability density function of X can be derived as follow:

$$P(X = x) = \sum_{i=1}^{N} P(X_i = x) \prod_{j \neq i} P(X_j \leq x) \tag{3.11}$$

Example probability density function are shown in Figure 3.12.

Finally, by considering observed lengths from multiple frames, we can use the EM algorithm to find each Gaussian component and the true fish length.

3.4.3 Experimental Result

We evaluate our 3D midline method on a more challenging dataset, which has more self-occlusion of fish body due to the top-down viewing angle of camera. For this dataset, we only have the distribution of fish length. The results are shown in Figure 3.13. The

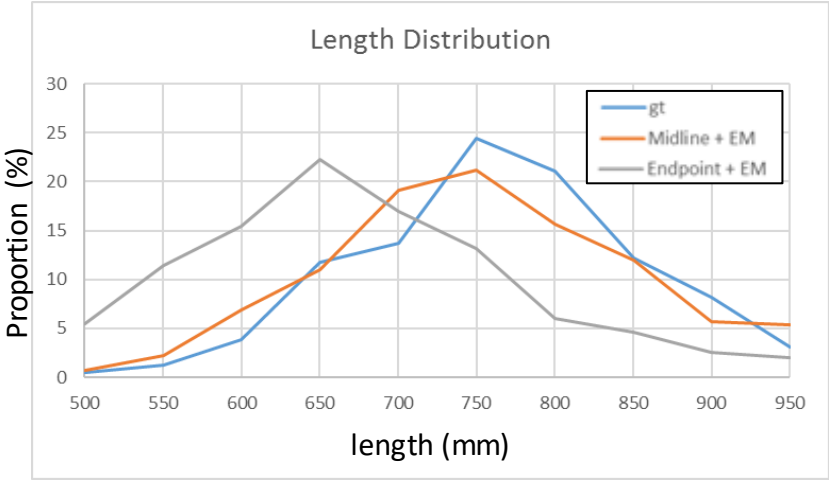


Figure 3.13 Histogram of length distribution measured from different methods.

Table 3-3 Earth moving distance (EMD) comparison

Method	**EMD (mm)**
3D Midline + EM	13.7
Endpoint + EM	88.5

histogram of fish length shows the midline method gives much smaller bias from the ground truth, while the endpoint method tends to underestimate the fish length. Besides, we also compare the earth moving distance (EMD) between the length histograms. The results are shown in Table 3-3. It shows the midline method gives much smaller EMD compared to endpoint method.

Table 3-4 Effect of Number of Gaussian Components

N	**EMD (mm)**
2	13.7
3	30.8
4	85.3
5	142.0

3.4.4 Discussion

Because the fish body can bend with a large degree of freedom, we try different number of Gaussian components in the EM algorithm to find the best performance. However, the result in Table 3-4 shows that using 2 components gives the best result. When the number of components increases, the performance drops dramatically. This is because the number of frames of observed lengths is too few (<100 frame for most fish) to solve so many components.

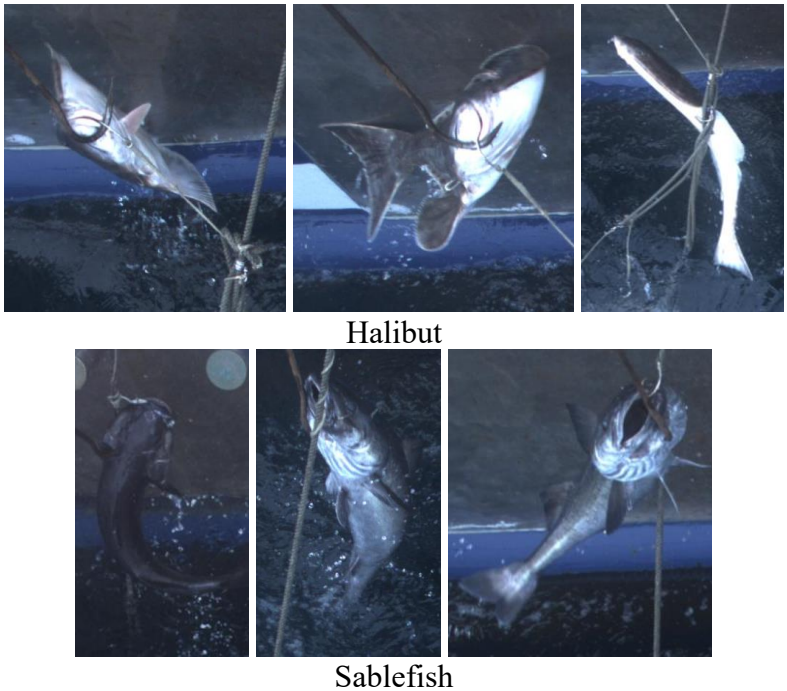
4 Fish Species Classification

Due to the deformation body of fish, the feature generation needs to be robust with fish in any orientations and poses, which yield diverse visual features (Figure 4.1). In addition, the classification method should deal with other challenges include the high visual similarity among fish species. Therefore, in the proposed work, we plan to develop a reliable fish species classification framework based on deep metric learning.

4.1 Metric Learning with Temporal Constraint

4.1.1 Triplet Loss

The deep metric learning has been used for verification, classification and clustering tasks [49-58]. Schroff et al. [49] proposes the triplet loss function to train the CNN to directly optimize the feature embedding for face verification task. In [49], the embedding is represented by $(x) \in \mathbb{R}^d$, with constraint $\|f(x)\|_2 = 1$. Motivated in the



Halibut

Sablefish

Figure 4.1 The changing shapes and orientation of fish result in large within-class variation.

context of nearest-neighbor classification, the model aims to make an image x_i^a (anchor) of a specific person closer to all other images x_i^p (positive) of the same person than to any images x_i^n (negative) of any other person (Figure 4.2):

$$\left\|x_i^a - x_i^p\right\|_2^2 + \alpha < \|x_i^a - x_i^n\|_2^2, \forall\left(x_i^a, x_i^p, x_i^n\right) \in \mathcal{T}, \tag{4.1}$$

where α is a margin between positive and negative pairs, and $\mathcal{T}$ is the set of all positive triplets in the training set. Thus, for a training set of size N. The triplet loss is thus defined as:

$$L_{triplet} = \sum_i^N \left[\left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha\right]_+ . \tag{4.2}$$

To make the model converge faster, given x_i^a the x_i^p and x_i^n of the triplet is selected as $\arg\max_{x_i^p}\left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2$ (hard positive) and $\arg\min_{x_i^n}\|f(x_i^a) - f(x_i^n)\|_2^2$ (hard negative). An online generation of triplets and large mini-batches strategy is used for practical computation of argmax and argmin. In practice, the hardest negatives can result in a collapsed model. Therefore, the semi-hard negative examples such that $\left\|f(x_i^a) - f\left(x_i^p\right)\right\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2$ are chosen in each training batch to mitigate this problem.

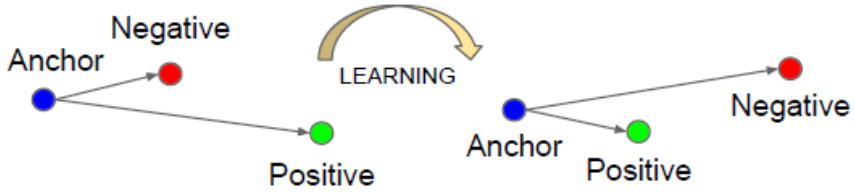


Figure 4.2 The triplet loss minimize the distance between an anchor and a positive example, and maximize the distance between the anchor and a negative example (figure from [49]).

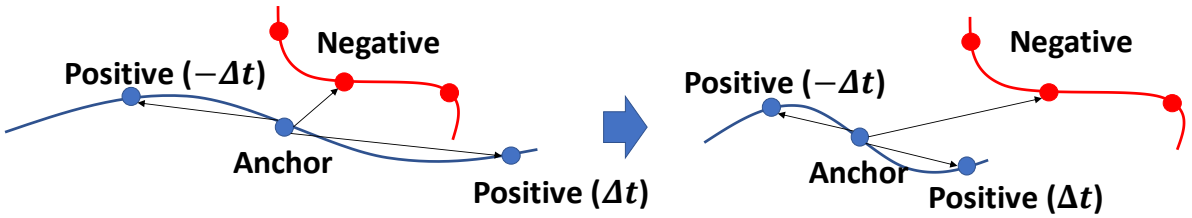


Figure 4.3 Temporal constraint for metric learning.

4.1.2 Temporal Loss Function

In our scenario, we want to add a stronger constraint on the temporal neighbors of the same class so that they can be closer in the feature space than different classes. Given an input image x^a at frame t, its temporal (positive) neighbor $x^{p_{\Delta t}}$ in the same track at frame $t + \Delta t$ and an image of different class x^n, we define the temporal loss function as:

$$L_{temp} = \max\left(0, \|f(x^a) - f(x^{p_{\Delta t}})\|_2^2 - \|f(x^a) - f(x^n)\|_2^2 + \alpha_{temp}\right), \quad (4.3)$$

where α_{temp} is the desired margin between temporal neighbors and negative examples. In this case, the temporal constraint only focuses on temporal neighbors and still allows larger distance between images which are less temporally-correlated. The idea is shown in Figure 4.3.

4.1.3 Adaptive Temporal Triplet

For training efficiency, we select the temporal triplets in a per-batch basis. We first randomly select N_a possible anchors within a batch. For any positive selection, which should have temporal relation with the anchor, but still different in visual features so that it is not redundant. Therefore, we adaptively select the Δt which satisfies:

$$\|f(x^a) - f(x^{p_{\Delta t}})\|_2^2 > \beta \text{ and } |\Delta t| \leq H, \quad (4.4)$$

where H is maximum time window.

For the negative selection, in order to train the model efficiently, we adopt the idea of semi-hard negative example [49], and select the negative example which satisfies:

$$\max_{|\Delta t'|\leq 2|\Delta t|} \|f(x^a) - f(x^{p_{\Delta t'}})\|_2^2 < \|f(x^a) - f(x^n)\|_2^2. \quad (4.5)$$

We use a larger time window for taking the maximum so that the negative example will implicitly be semi-hard to $x^{p_{\Delta t}}$ as well.

4.2 Representative Feature Classifier

4.2.1 System Overview

The first classification method we proposed is the representative feature classifier. An overview of the system is shown in Figure 4.4. First, the fish images are sent into a base CNN (Inception-ResNet-v2 [45]) to generate d-dimensional feature embeddings (d=512 in our experiment). Then, for each of the C classes, we assume there are K representative features, which are in the same dimension as the feature embeddings. To learn the KC representative features, we treat each of them as a column in the weight matrix of a fully-connected layer, where each of the KC outputs corresponds to a representative feature. Finally, a max pooling layer is applied to select the maximum responses within each class to generate the C-dimensional output, which is the classification result. We refer the fully-connected layer and the max pooling layer as the representative feature classifier.

The model is trained with the following loss functions. The temporal loss L_{temp}, which is defined in (4.3), is calculated from the feature embeddings of contiguous frames, given the temporal constraint in metric learning. The softmax loss $L_{softmax}$ is calculated from the representative feature classifier output. The representative feature loss L_{rep} is a regularization term to make the representative features discriminative.

In testing stage, first the feature vectors of a series of frames of the tracked fish are calculated. Then, the features are aggregated into several clips along the time and sent into the fully-connected layers for classification. The final classification result is acquired by weighted majority vote of all the clips.

4.2.2 Representative Feature

To deal with the large intra-class variations, we introduce K representative features for each of the C classes (Figure 4.5). The KC representative features are treated as the column vectors of a $d \times KC$ weight matrix of the fully-connected layer. We denote the representative features of class c as $W_k^c, k = 1, \dots, K$. We normalize the W_k^c's and set the bias term of fully-connected layer to 0 [74, 75]. Then, given a set of input image and class label $(x_i, y_i), i = 1, \dots, N$, the softmax loss $L_{softmax}$ after the max pooling layer can be written as:

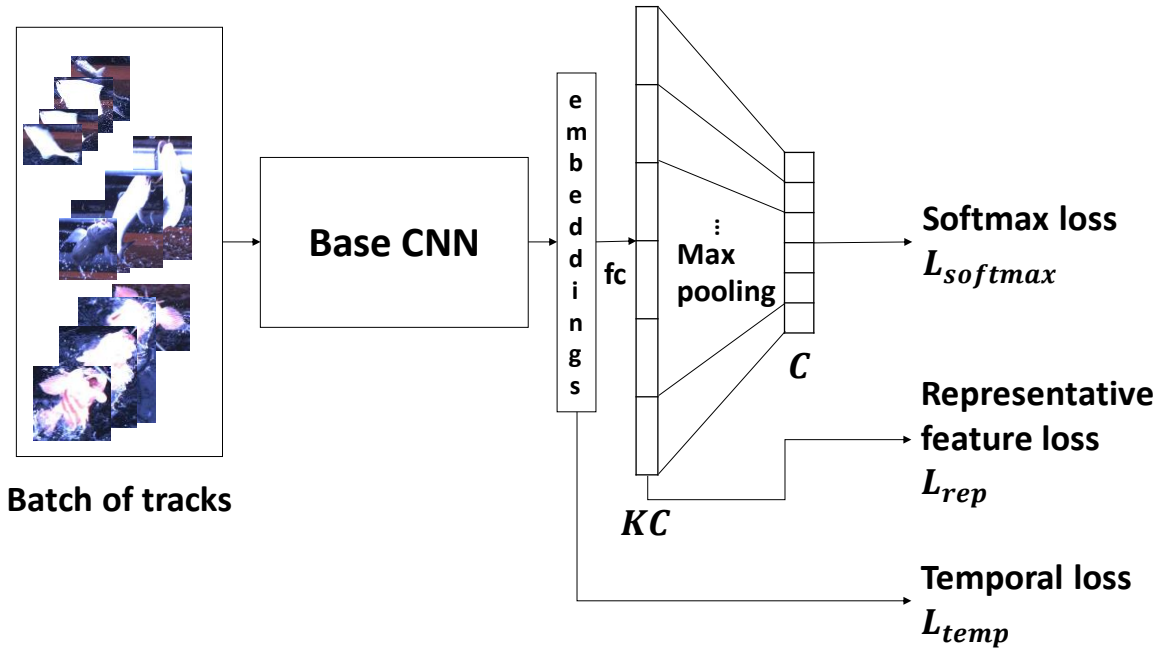


Figure 4.4 System overview of representative feature classifier.

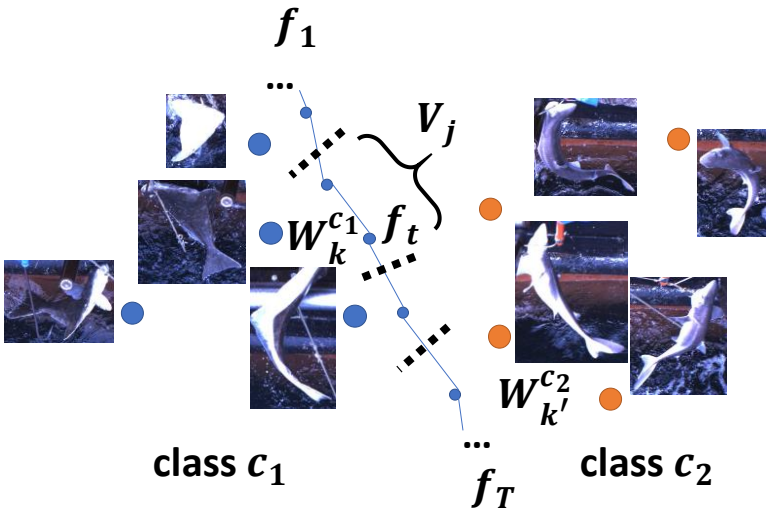


Figure 4.5 Representative features for large intra-class variations.

$$L_{softmax} = \frac{1}{N}\sum_{i=1}^{N} -\log \frac{\max_k e^{s\left(W_k^{y_i}\right)^T f(x_i)}}{\sum_{c=1}^{C} \max_k e^{s\left(W_k^{c}\right)^T f(x_i)}}, \tag{4.6}$$

where s is a scaling constant (we use s=64 as suggested in [76]). The softmax loss function implies that as long as the input has a strong response with a representative feature of the class it belongs to, it can be correctly classified.

To learn the diverse features within a class, we want to encourage the representative features of the same class to be different from each other. Therefore, we regularize each W_k^c using the representative feature loss function L_{rep}.

$$L_{rep} = \max_{k' \neq k}\left(0, \left\|W_k^c - W_{k'}^c\right\|_2^2 - \beta_1\right) + \max_{k' \neq k}\left(0, -\left\|W_k^c - W_{k'}^c\right\|_2^2 + \beta_2\right) \tag{4.7}$$

The parameter β_1 is the margin of the largest distance of the representative features of the same class, and β_2 is the margin of the smallest distance between the representative features.

Finally, we we define the total loss as the combination of the three losses,

$$L_{total} = L_{softmax} + \lambda_{temp} L_{temp} + \lambda_{rep} L_{rep}. \quad (4.8)$$

4.2.3 Feature Aggregation and Classification

For classification of a tracked fish, our goal is to use the temporal information and aggregate the contiguous frames to get a final classification result. One of the simplest methods is to perform classification of each frame independently and then perform a majority vote or a weighted-majority vote based on the class probability. However, prediction based on single frame may suffer from noise or occlusion in images, and thus affect the final voting results. Therefore, rather than performing a per-frame prediction, we aggregate the feature vectors of contiguous frames to perform classification and final voting.

First, we pass all the frames of a tracked fish into the convolutional network to get the feature vectors $f_t, t = 1, \dots, T$. Then we aggregate the contiguous features into one clip V_j if they have the maximum response to the same representative feature:

$$V_j = \left\{ f_t \middle| \arg\max_{c,k} (W_k^c)^T f_t = c_j, k_j \forall\ t = t_j, \dots, t_{j+1} - 1 \right\} \quad (4.9)$$

where t_j is the beginning frame of clip V_j (Figure 4.5). Then we take the averaged feature $v_j = \sum_{f_t \in V_j} f_t / |V_j|$ of the clip and pass it to the fully-connected classifier layer to get the classification result $p_c(v_j)$. Finally, we can perform a weighted majority vote based on the classification results of all clips to get the final per-track classification result:

$$c^* = \arg\max_c \sum_j p_c(v_j) \times |V_j|. \quad (4.10)$$

In practice, we try to aggregate as many contiguous features as we can in each (non-overlapping) clip from the same track, but limited to a maximum size of clip as 16 frames

to avoid over-smoothing.

4.3 Semantically-Decoupled Temporal Attention

4.3.1 System Overview

The second classification method we proposed is the semantically-decoupled temporal attention model. An overview of the system is shown in Figure 4.6. First, T frames of fish images are sent into a base CNN (Inception-ResNet-v2 [45]) to generate T d-dimensional frame-level feature embeddings (d=512 in our experiment). Then, K temporal attention groups are used to apply the attention weighted average on the frame-level features to generate K clip-level features. The K clip-level features are then passed through the group assignment process to generate the final clip-level feature. Finally, we can use the final clip-level feature for classification purpose.

The model is trained with the following loss functions. The temporal loss L_{temp} is defined in (4.3). The softmax loss $L_{softmax}$ is calculated from the clip-level classifier output. The diversity loss L_{div} is a regularization term to enforce the temporal diversity of

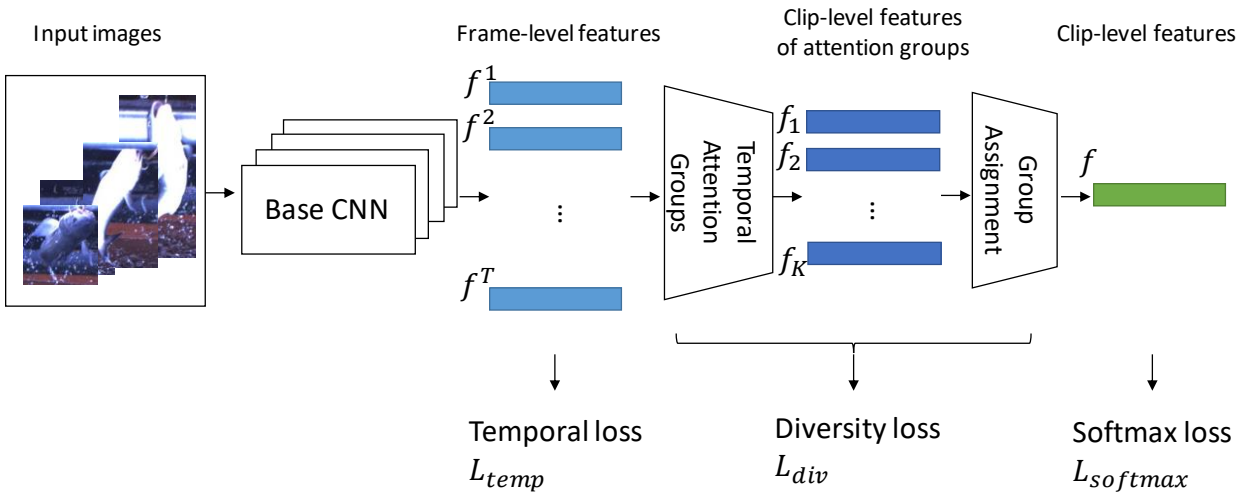


Figure 4.6 System overview of the semantically-decoupled attention model.

the attention model.

At testing, because the input sequence may have different length, we first cut the frames into several clips, perform prediction for each clip independently, and finally perform a weighted majority vote to get the final classification result.

4.3.2 Temporal Attention

In the temporal attention model, our goal is to apply an attention weighted average on the frame-level features of a series of contiguous frames (clip) to represent the clip-level feature. Given a video clip of length T, frame-level feature $f^t \in \mathbb{R}^d, t \in [1, T]$ from backbone neural network and attention weights $w^t \in \mathbb{R}, t \in [1, T]$, we have the clip-level feature:

$$f' = \sum_{t=1}^{T} w^t f^t . \tag{4.11}$$

To generate the attention weight, we use a temporal convolution attention module [65] as shown in Figure 4.7. Here f_m^t is the mid-level feature from the middle layer of backbone neural network, and the convolution is performed along the time domain. From the output of the temporal convolution s^t, we apply the softmax function,

$$w^t = \frac{e^{s^t}}{\sum_{i=1}^{T} e^{s^t}} \tag{4.12}$$

so that $\sum_{t=1}^{T} w^t = 1$.

Finally, the clip-level feature f' is passed to a fully-connected layer for classification. We can then use the softmax cross-entropy loss to train the classifier for C classes,

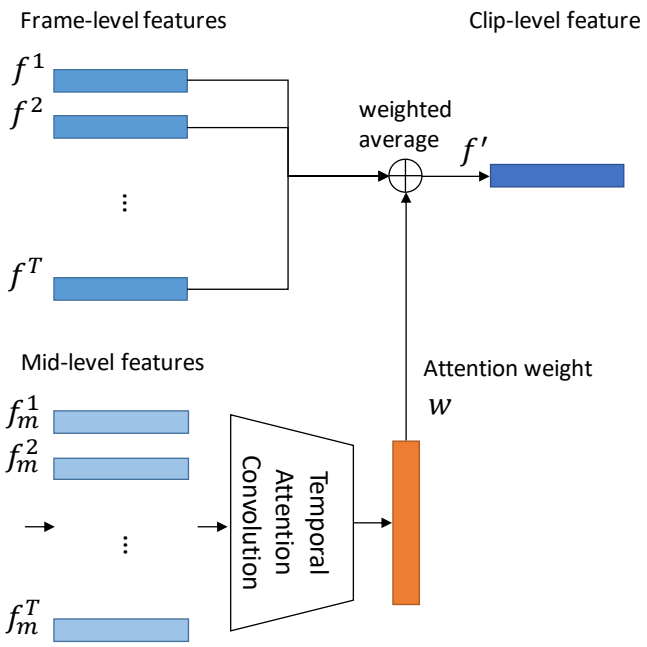


Figure 4.7 Temporal attention convolution.

$$L_{softmax} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} -y_{i,c}\log p_c(f_i'), \tag{4.13}$$

where y_i is the one-hot encoding of ground truth class and $p(f_i)$ is the predicted probability of sample i.

However, the limitation of the above temporal attention model is that all the 512 feature dimensions in f^t share the same attention weight w^t. If different feature dimensions need attention at different frames, one attention weight may not be able to catch this diversity along time and feature dimension. Therefore, in the following section, we propose a semantically-decoupled temporal attention model, which generate different attention weights for different feature dimensions, to address this issue.

4.3.3 Attention Group

To generate different attention weights for different feature dimensions, we want to decouple the feature dimension into K semantic attention groups, so that each group can

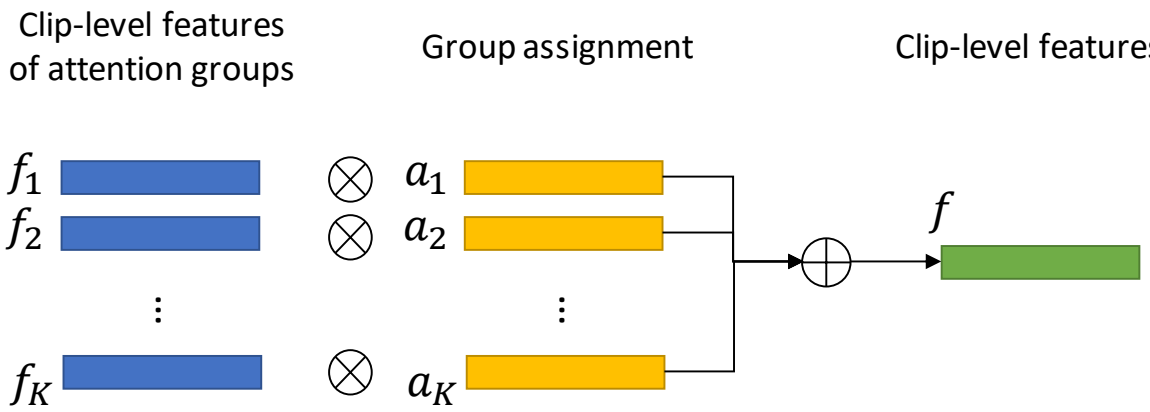


Figure 4.8 Generate final clip-level feature from K attention groups.

has its own temporal attention module. The resulting clip-level feature becomes:

$$f = \sum_{k=1}^{K} f_k \otimes a_k \,, \tag{4.14}$$

where $\otimes$ means the element-wise multiplication, f_k is the clip-level feature given by attention group k, $a_k \in [0,1]^d$ is the group assignment of each feature dimension to group k, and $\sum_{k=1}^{K} a_k = \mathbf{1}$ (Figure 4.8). With the assignment, only feature dimensions of the same group will share the same attention weight, and each temporal attention module will focus on the assigned dimensions only. Furthermore, to efficiently learn the group assignment a_k, we relax each dimension d_i of a_k to a real number between 0 and 1, i.e. $0 \leq a_k(d_i) \leq 1 \forall d_i \in [1, d]$. This is done by applying a softmax function among the weight α_k for all K groups,

$$a_k(d_i) = \frac{e^{\alpha_k(d_i)}}{\sum_{i=1}^{K} e^{\alpha_i(d_i)}} \forall d_i \in [1, d]. \tag{4.15}$$

4.3.4 Diversity Constraint

To efficiently learn the group assignment, we apply a temporal diversity constraint,

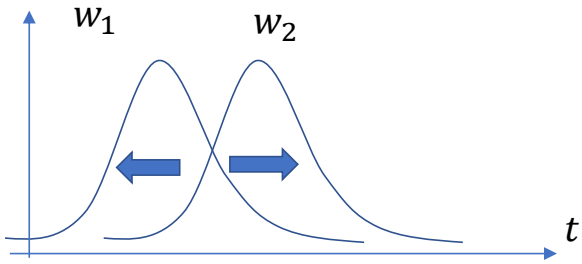


Figure 4.9 Illustration of diversity constraint along temporal domain. Different groups are forced to focus on different frames. The diversity loss is the overlapping area under both curves.

which will implicitly make the assignment sparse and reduce the redundancy. Our strategy is to enforce the attention weights from different groups to focus on different frame during training. Therefore, we define the diversity loss,

$$L_{div} = \sum_{t=1}^{T} \sum_{k=1}^{K} 1\left(w_k^t \neq \max_{k'} w_{k'}^t\right) \times w_k^t, \tag{4.16}$$

which penalizes the attention weight at frame t if it is not the maximum among all groups. Intuitively, if two attention groups give high attention weights on the same frames, one of the groups is redundant and should be either merged to the other group or assigned with other feature dimensions to focus on. The idea is illustrated in Figure 4.9.

Finally, we define the total loss as the combination of the three losses,

$$L_{total} = L_{softmax} + \lambda_{div} L_{div} + \lambda_{temp} L_{temp}. \tag{4.17}$$

4.3.5 Testing Stage

At testing, the input sequence may have variable length which is different from the time window T used for training. Therefore, we first cut the input sequence into S clips of length T and perform inference on each clip-level feature individually. Since the temporal information is already well captured by the attention model, we directly take the weighted majority vote of each clip as the final classification result,

$$c^* = \arg\max_c \sum_{s=1}^{S} p_c(f_s)\,, \tag{4.18}$$

where f_s is the clip-level feature of clip s, and p_c is the predicted probability of class c.

4.4 Experimental Result

4.4.1 Dataset and Simulation Setup

The proposed method is evaluated on a dataset, which consists of 141,685 cropped image frames of 22 classes of fish. The training and testing set consists of 72,840 and 68,845 images respectively. The number of fish instances in each class are highly unbalanced, ranged from 865 tracks to 10 tracks for each class. Because the images are from video frames, we split the training and testing set in a per-track basis to avoid presenting the same fish instance in both training and testing sets. The dataset is captured on 6 different cameras from different fishing vessels. The bounding boxes of the fish and the fish species are labelled by experienced fishery scientists.

Representative Feature Classifier

For training the representative feature classifier model, we select the Inception-ResNet-v2 as the base CNN for its effectiveness in classification and verification tasks. We use the model weights pretrained on ImageNet and perform fine-tuning on our dataset. For the model parameters, we set the feature dimension d as 512 and the number of representative features K for each class as 8. The margins α_{temp} and β are set to 0.5 and 0.1, respectively, and the parameters (β_1, β_2) are set to (0.3, 0.15). The total loss is a weighted sum of softmax loss, temporal loss and representative feature loss with weights 1.0, 0.1, 0.1.

Semantically-Decoupled Temporal Attention

Table 4-1 Classification results

Method	**Total accuracy (%)**		**Mean per-class accuracy (%)**	
	Per-image	Per-track	Per-image	Per-track
Inception-ResNet-v2	95.0	96.1	81.3	85.3
VGG-16	94.6	95.8	77.3	81.8
DFL-CNN [78]	96.4	97.1	85.0	88.0
MA-CNN [77]	96.1	97.0	83.1	87.4
Representative feature	96.7	98.2	88.0	90.1
Semantically-Decoupled TA	**98.3**	**98.6**	**89.6**	**90.3**

For training the semantically-decoupled temporal attention model, we use the same setting of Inception-ResNet-v2 for fair comparison with the previous method. In addition, we set the mid-level feature dimension for f_m^t as 256. For the model parameter, we set the clip length T as 16 and number of attention groups K as 8. The weight for temporal diversity loss α_{div} is set to 0.1, and the weight for temporal triplet loss α_{temp} is set to 0.1.

4.4.2 Classification Performance

The classification total accuracy and mean per-class accuracy are shown in Table 4-1. We compare our method with the conventional softmax classifiers and other state-of-the-art methods. For the representative feature classifier, the per-frame accuracy is acquired by passing each frame into the model and make prediction independently without temporal aggregation. For the temporal attention classifier, we take the clip-level prediction result as the per-frame prediction for all frames within the clip. For other single image-based methods, the per-track accuracy is acquired by weighted majority vote of classification probabilities from all the involved frames. It shows that although our method gives similar performance as others on per-frame accuracy, its per-track accuracy is much higher, showing the effectiveness of the feature aggregation along temporal domain. Besides, ours

Table 4-2 Importance of each component in representative feature classifier

Loss functions	**Total accuracy (%)**	**Mean per-class accuracy (%)**
Softmax	96.1	85.3
Softmax + Temp.	96.4	88.0
Softmax + Rep.	96.9	87.4
Softmax + Temp. + Rep.	98.2	90.1

achieves highest mean per-class accuracy than others by a large margin, showing that the minority classes get more benefit from the proposed method. In addition, between the two proposed methods, the temporal attention model performs better, especially in the per-image accuracy.

4.5 Discussion

4.5.1 Representative Feature Classifier

Ablation Study

We investigate the importance of each part in the proposed representative feature classifier model based on the ablation study. The comparison in Table 4-2 shows that the temporal loss or representative feature loss does not improve the accuracy individually, but when trained together, the model gives higher total accuracy and mean per-class accuracy.

Number of Representative Features

Table 4-3 Effect of number of representative features

K	**Total accuracy (%)**	**Mean per-class accuracy (%)**
2	96.3	85.4
4	97.5	87.9
8	98.2	90.1
12	97.2	88.5
16	96.0	86.9

Table 4-4 Importance of each component in semantically-decoupled TA model

Components	**Total accuracy (%)**	**Mean per-class accuracy (%)**
w/o TA	96.1	85.3
w/ TA	97.1	87.8
w/ TA + Div.	98.2	89.0
w/ TA + Temp.	97.8	88.8
w/ TA + Temp. + Div.	98.6	90.3

We also try different number of representative features K to see the effectiveness of the proposed method. The result in Table 4-3 shows that the performance reaches the top when $K = 8$. For larger K, the performance drops, because the representative feature loss L_{rep} makes the training instable when enforcing too many representative features to represent the intra-class difference which is not so diverse. Besides, another reason could be due to the feature aggregation step at testing stage. When we use too many representative features to aggregate the input frames into clips, contiguous frames are more likely to be assigned to different representative features, and the resulting clips will be too short and less robust for averaging out the noise from single frame.

4.5.2 Semantically-Decoupled Temporal Attention

Ablation Study

We investigate the importance of each component in the proposed semantically-decouple temporal attention classifier model. The results in Table 4-4 show that the temporal-attention alone does not increase the performance much, unless the temporal loss and diversity loss are presented during training. Besides, the diversity loss improves the performance by a large margin, showing that it can effectively help the model to learn better assignment for each attention group.

Table 4-5 Effect of number of attention groups

K	**Total accuracy (%)**	**Mean per-class accuracy (%)**
1	96.4	86.0
2	97.1	87.1
4	98.0	89.5
8	98.6	90.3
16	98.7	90.6
24	98.4	90.3

Number of Attention Groups

We try different number of attention groups to see its effect on the proposed model. Table 4-5 shows the performance of $K = 1$ to 24. Note that when $K = 1$, the model is equivalent to the basic temporal attention without grouping, and the diversity loss L_{div} is not applied. The result shows the performance increases for larger K, and reaches plateau for $K \geq 16$. The reason is that the learned feature embedding from neural network is already highly concentrated and around 16 groups are enough to capture the diversity of temporal relationship between frames for classification task.

5 Conclusion

In this dissertation, an automatic video analysis system for electronic monitoring of fishery activities on dynamic and noisy wild sea surface is proposed. The system consists of two major parts, including fish tracking/segmentation/measurement and fish species classification.

The fish tracking, segmentation and measurement system takes stereo video as input, and use the calculated depth information to perform 3D reasoning. By combining the result from deep learning object detector and the 3D information, the 2D object proposal are converted into 3D object proposals for reliable tracking in 3D. A Kalman filter rescoring strategy is used to help the tracker to select the best 3D object proposals, in terms of object's position, size and diagonal length. Besides, the model parameters can be efficiently learned from the training data. For fish segmentation, an efficient background plane clustering strategy is used to classify the foreground pixel from background plane in 3D. The pixel location and surface normal in 3D are used to calculate the distance in clustering step. A refinement step based on CRF is used to recover the fine detail of segmentation. Finally, the length of the tracked fish is estimated from 3D midline and EM algorithm. Experimental result shows the proposed system can achieve reliable tracking, segmentation and length measurement for highly deformable fish in noisy wild sea surface.

The fish species classification model takes the tracked fish in video as input and use the temporal information between frames to extract useful features. By applying the deep metric learning with a temporal constraint, we force the model to learn the closeness between temporal neighbor frames. Based on the temporal constraint, two classification models, representative feature classifier and semantically-decoupled temporal attention, are proposed. The representative feature classifier discriminatively learns the representative feature of each class and uses them for feature aggregation during prediction. The semantically-decoupled temporal attention model

learns multiple attention groups to focus on different feature dimensions. A diversity constraint is applied to make different attention groups focus on different frames and feature dimensions. Experimental result shows the proposed classification methods can outperform other state-of-the-art methods.

In conclusion, the proposed automatic video analysis system shows that with appropriate modeling and learning of the 3D and temporal information, the challenges from fish subject and sea environment can be overcome. Besides, the promising experimental results validate the proposed system can benefit the fishery regulatory and scientific survey.

References

[1] T.-W. Huang, J.-N. Hwang, and C.S. Rose "Chute-based Automated fish Length Measurement and Water Drop Detection," IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2016.

[2] T.-W. Huang, J.-N. Hwang, S. Romain, and F. Wallace, "Live Tracking of Rail-Based Fish Catching on Wild Sea Surface," Workshop on Computer Vision for Analysis of Underwater Imagery (CVAUI), IEEE 23rd International Conference on Pattern Recognition (ICPR), 2016.

[3] T.-W. Huang, J.-N. Hwang, S.-T. Shen, S. Romain, and F. Wallace, "Live Tracking of Rail-Based Fish Catching on Wild Sea Surface," American Fisheries Society 141th Annual Meeting, Aug. 2017.

[4] T.-W. Huang, J.-N. Hwang, S. Romain, and F. Wallace, "Live Tracking of Rail-Based Fish Catching on Wild Sea Surface," submitted to IEEE Trans. on Circuits and Systems for Video Technology (TCSVT), 2018.

[5] B. Zion, "The Use of Computer Vision Technologies in Aquaculture – A Review," Computers and Electronics in Agriculture, vol. 88, pp. 125-132, 2012.

[6] J.R. Mathiassen, E. Misimi, M. Bondø, E. Veliyulin, and S.O. Østvik, "Trends in Application of Imaging Technologies to Inspection of Fish and Fish Products," Trends in Food Science and Technology, vol. 22, no. 6, pp. 257-275, 2011.

[7] D.J. White, C. Svellingen, and N.J.C. Strachan, "Automated Measurement of Species and Length of Fish by Computer Vision," Fisheries Research, vol. 80, pp. 203-210, 2006.

[8] M.-C. Chuang, J.-N. Hwang, and C.S. Rose, "Aggregated Segmentation of Fish from Conveyor Belt Videos," IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2013.

[9] K. Williams, N. Lauffenburger, M.-C. Chuang, J.-N. Hwang, and R. Towler, "Automated measurements of fish within a trawl using stereo images from a Camera-Trawl device (CamTrawl)," Methods in Oceanography, vol. 17, pp. 138-152, 2016.

[10] M.-C. Chuang, J.-N. Hwang, K. Williams, and R. Towler, "Tracking Live Fish from Low-Contrast and Low-Frame-Rate Stereo Videos," IEEE Trans. on Circuits and Systems for Video Technology (TCSVT), vol. 25, no. 1, pp. 167-179, 2015.

[11] G. Wang, J.-N. Hwang, K. Williams, and G. Cutter, "Closed-Loop Tracking-by-Detection for ROV-Based Multiple Fish Tracking," Workshop on Computer Vision for Analysis of Underwater Imagery (CVAUI), IEEE 23rd International Conference on Pattern Recognition (ICPR), 2016.

[12] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and Tracking of Multiple Humans in Crowded Environments," IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI), vol. 30, no. 7, pp. 1198-1211, 2008.

[13] C.-T. Chu, J.-N. Hwang, S.-Z. Wang, and Y.-Y. Chen, "Human tracking by adaptive Kalman filtering and multiple kernels tracking with projected gradients," ACM/IEEE International Conf. on Distributed Smart Cameras (ICDSC), August 22-25, 2011.

[14] C.-T. Chu, J.-N. Hwang, H.-I. Pai, and K.-M. Lan, "Tracking Human Under Occlusion Based on Adaptive Multiple Kernels With Projected Gradients," IEEE Trans. on Multimedia (TMM), vol. 15, no. 7, pp. 1602-1615, 2013.

[15] Z. Tang, J.-N. Hwang, Y.-S. Lin, and J.-H. Chuang, “Multiple-kernel adaptive segmentation and tracking (MAST) for robust object tracking,” IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2016.

[16] K. Kang, W. Ouyang, H. Li, and X. Wang, “T-CNN: Tubelets with Convolutional Neural Networks for Object Detection from Videos,” IEEE Trans. on Circuits and Systems for Video Technology (TCSVT), 2017.

[17] L. Wang, W. Ouyang, X. Wang, and H. Lu, “Visual Tracking with Fully Convolutional Networks,” IEEE International Conf. on Computer Vision (ICCV), 2015.

[18] S.-H. Bae, and K.-J. Yoon, “Robust Online Multi-Object Tracking based on Tracklet Confidence and Online Discriminative Appearance Learning,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2014.

[19] M. Andriluka, S. Roth, and B. Schiele, “People-Tracking-by-Detection and People-Detection-by-Tracking,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2008.

[20] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2008.

[21] R. Girshick, “Fast R-CNN,” IEEE International Conf. on Computer Vision (ICCV), 2015.

[22] J. Redmon, and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2017.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single Shot MultiBox Detector,” European Conf. on Computer Vision (ECCV), 2016.

[24] K. Yamaguchi, D. McAllester, and R. Urtasun, “Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation,” European Conf. on Computer Vision (ECCV), 2014.

[25] D. Min, J. Lu, and M.N. Do, “Depth Video Enhancement Based on Weighted Mode Filtering,” IEEE Trans. on Image Processing (TIP), vol. 21, no. 3, pp. 1176-1190, 2011.

[26] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning Rich Features from RGB-D Images for Object Detection and Segmentation,” European Conf. on Computer Vision (ECCV), 2014.

[27] X. Ren, L. Bo, and D. Fox, “RGB-(D) Scene Labeling: Features and Algorithms,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2012.

[28] E. Herbst, P. Henry, and D. Fox, “Toward Online 3-D Object Segmentation and Mapping,” IEEE International Conf. on Robotics and Automation (ICRA), 2014.

[29] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker Jr., A. Rodriguez, and J. Xiao, “Multi-view Self-supervised Deep Learning for 6D Pose Estimation in the Amazon Picking Challenge,” IEEE International Conf. on Robotics and Automation (ICRA), 2017.

[30] X. Chen, K. Kundu, Y. Zhu, A.G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, “3D Object Proposals for Accurate Object Class Detection,” Advances in Neural Information Processing Systems (NIPS), 2015.

[31] E. Trucco, F. Isgro, and F. Bracchi, “Plane detection in disparity space,” International Conf. on Visual Information Engineering (VIE 2003). Ideas, Applications, Experience, pp. 73-76, 2003.

[32] F.A. Limberger, and M.M. Oliveira, “Real-Time Detection of Planar Regions in Unorganized Point Clouds,” Pattern Recognition, vol. 48, no. 6, pp. 2043-2053, 2015.

[33] J. Poppinga, N. Vaskevicius, A. Birk, and K. Pathak, “Fast Plane Detection and Polygonalization in noisy 3D Range Images,” IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS), 2008.

[34] C. Stauffer, W. Eric, and L. Grimson, “Adaptive Background Mixture Models for Real-time Tracking,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 1999.

[35] P. St-Charles, G. Bilodeau, and R. Bergevin, “SuBSENSE: A Universal Change Detection Method with Local Adaptive Sensitivity,” IEEE Trans. on Image Processing (TIP), vol. 24, no. 1, pp. 359-373, 2015.

[36] P. Krähenbühl, and V. Koltun, “Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials,” Advances in Neural Information Processing Systems (NIPS), 2011.

[37] B. Wu, and R. Nevatia, “Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors,” International Journal of Computer Vision, vol. 75, no. 2, pp. 247-266, 2007.

[38] J. Valmadre, L. Bertinetto, J.F. Henriques, A. Vedaldi, and P.H.S. Torr, “End-to-end representation learning for Correlation Filter based tracking,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2017.

[39] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. Torr, “Staple: Complementary Learners for Real-Time Tracking,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[40] R. Tao, E. Gavves, and A. W. M. Smeulders, “Siamese instance search for tracking,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[41] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC Superpixels Compared to State-of-the-Art Superpixel Methods,” IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI), vol. 34, no. 11, pp. 2274-2282, 2012.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in Neural Information Processing Systems (NIPS), 2012.

[43] K. Simonyan, and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” International Conference on Learning Representations (ICLR), 2015.

[44] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[45] C. Szegedy, S. Ioffe, V. Vanhoucke, and A.A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” AAAI Conference on Artificial Intelligence, 2017.

[46] S. H. Khan, M. Hayat, M. Bennamoun, F.A. Sohel, and R. Togneri, “Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data,” IEEE Trans. on Neural Networks and Learning Systems (TNNLS), 2017.

[47] Q. Dong, S. Gong, and X. Zhu, “Class Rectification Hard Mining for Imbalanced Deep Learning,” IEEE International Conf. on Computer Vision (ICCV), 2017.

[48] F. Wu, X.-Y. Jing, S. Shan, W. Zuo, and J.-Y. Yang, “Multiset Feature Learning for Highly Imbalanced Data Classification,” AAAI Conference on Artificial Intelligence, 2017.

[49] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A Unified Embedding for Face Recognition and Clustering,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015.

[50] H.O. Song, Y. Xiang, S. Jegelka, and S. Savarese, “Deep Metric Learning via Lifted Structured Feature Embedding,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[51] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev, “Metric Learning with Adaptive Density Discrimination,” International Conference on Learning Representations (ICLR), 2016.

[52] C. Ding, and D. Tao, “Trunk-Branch Ensemble Convolutional Neural Networks for Video-Based Face Recognition,” IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI), vol. 40, no. 4, 2018.

[53] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, “Deep Metric Learning with Angular Loss,” IEEE International Conf. on Computer Vision (ICCV), 2017.

[54] C. Huang, Y. Li, C.C. Loy, and X. Tang, “Learning Deep Representation for Imbalanced Classification,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[55] Y. Wang, J. Choi, V. I. Morariu, and L.S. Davis, “Mining Discriminative Triplets of Patches for Fine-Grained Classification,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[56] X. Zhang, F. Zhou, Y. Lin, and S. Zhang, “Embedding Label Structures for Fine-Grained Feature Representation,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[57] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, “Deep Relative Distance Learning: Tell the Difference Between Similar Vehicles,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[58] Y. Cui, F. Zhou, Y. Lin, and S. Belongie, “Fine-grained Categorization and Dataset Bootstrapping using Deep Metric Learning with Humans in the Loop,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016.

[59] I. Misra, C.L. Zitnick, and M. Hebert, “Shuffle and Learn: Unsupervised Learning using Temporal Order Verification,” European Conf. on Computer Vision (ECCV), 2016.

[60] P. X. Huang, B. J. Boom, and R. B. Fisher, "Hierarchical classification with reject option for live fish recognition," Machine Vision and Applications, vol. 26, no. 1, pp. 89-102, 2015.

[61] M.-C. Chuang, J.-N. Hwang, and K. Williams, “ A Feature Learning and Object Recognition Framework for Underwater Fish Images, ” IEEE Trans. on Image Processing (TIP), vol. 25, no. 4, pp. 1862-1872, 2016.

[62] G. Wang, J.-N. Hwang, K. Williams, F. Wallace, and C. Rose, “Shrinking Encoding with Two-Level Codebook Learning for Fine-Grained Fish Recognition,” Workshop on Computer Vision for Analysis of Underwater Imagery (CVAUI), IEEE 23rd International Conference on Pattern Recognition (ICPR), 2016.

[63] S.A. Siddiqui, A. Salman, M.I. Malik, F. Shafait, A. Mian, M.R. Shortis, and E.S. Harvey, “Automatic fish species classification in underwater videos: exploiting pre-trained deep neural network models to compensate for limited labelled data,” ICES Journal of Marine Science, vol.75, no. 1, 2018.

[64] F. Kratzert, and H. Mader, “Fish species classification in underwater video monitoring using Convolutional Neural Networks,” EarthArXiv, doi:10.17605/OSF.IO/DXWTZ, 2018.

[65] J. Gao, and R. Nevatia, “Revisiting Temporal Modeling for Video-based Person ReID,” arXiv:1805.02104, 2018.

[66] S. Li, S. Bak, P. Carr, and X. Wang, “Diversity Regularized Spatiotemporal Attention for Video-based Person Re-identification,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2018.

[67] L. Wu, Y. Wang, J. Gao, and X. Li, “Where-and-When to Look: Deep Siamese Attention Networks for Video-Based Person Re-Identification,” IEEE Trans. on Multimedia (TMM), vol. 21, no. 6, 2019.

[68] S. Xu, Y. Cheng, K. Gu, Y. Yang, S. Chang, and P. Zhou, “Jointly Attentive Spatial-Temporal Pooling Networks for Video-based Person Re-Identification,” IEEE International Conf. on Computer Vision (ICCV), 2017.

[69] X. Song, C. Lan, W. Zeng, J. Xing, X. Sun, and J. Yang, “Temporal-Spatial Mapping for Action Recognition,” IEEE Trans. on Circuits and Systems for Video Technology (TCSVT), DOI 10.1109/TCSVT.2019.2896029, 2019.

[70] X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen, “Attention Clusters: Purely Attention Based Local Feature Integration for Video Classification,” IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[71] X. Long, C. Gan, G. de Melo, X. Liu, Y. Li, F. Li, and S. Wen, “Multimodal Keyless Attention Fusion for Video Classification,” AAAI Conference on Artificial Intelligence, 2018.

[72] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D.A. Ross, and J. Deng, and Rahul Sukthankar, “Rethinking the Faster R-CNN Architecture for Temporal Action Localization,” IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[73] Y. Peng, Y. Zhao, and J. Zhang, “Two-Stream Collaborative Learning With Spatial-Temporal Attention for Video Classification,” IEEE Trans. on Circuits and Systems for Video Technology (TCSVT), vol. 29, no.3, 2019.

[74] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “SphereFace: Deep Hypersphere Embedding for Face Recognition,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2017.

[75] F. Wang, X. Xiang, J. Cheng, and A.L. Yuille, “NormFace: L2 Hypersphere Embedding for Face Verification,” 25th ACM international conference on Multimedia, 2017.

[76] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “CosFace: Large Margin Cosine Loss for Deep Face Recognition,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2018.

[77] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo, “Learning multi-attention convolutional neural network for fine-grained image recognition,” IEEE International Conf. on Computer Vision (ICCV), 2017.

[78] Yaming Wang, Vlad I. Morariu, and Larry S. Davis, “Learning a Discriminative Filter Bank Within a CNN for Fine-Grained Recognition,” IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2018.