

Article

Behavior Recognition of Squid Jigger Based on Deep Learning

Yifan Song ^{1,2}, Shengmao Zhang ^{1,*}, Fenghua Tang ¹, Yongchuang Shi ¹, Yumei Wu ¹, Jianwen He ³, Yunyun Chen ⁴ and Lin Li ⁵

- ¹ Key Laboratory of Fisheries Remote Sensing, Ministry of Agriculture and Rural Affairs, East China Sea Fisheries Research Institute, Chinese Academy of Fishery Sciences, Shanghai 200090, China; syf_fyss@163.com (Y.S.); f-h-tang@163.com (F.T.); syc13052326091@163.com (Y.S.); wuym@ecsf.ac.cn (Y.W.)
- ² College of Information, Shanghai Ocean University, Shanghai 201306, China
- ³ China Agricultural Development Group Zhoushan Ocean Fishing Co., Ltd., Zhoushan 316100, China; hjw809893@163.com
- ⁴ China Aquatic Products Zhoushan Marine Fisheries Corporation Co., Ltd., Zhoushan 316100, China; cnfc_cyy@163.com
- ⁵ Inspur Group Co., Ltd., Jinan 250000, China; lilintx@inspur.com
- * Correspondence: zhangsm@ecsf.ac.cn

Abstract: In recent years, with the development of pelagic fishing, the working environment and monitoring of crew (squid jigger) members have become increasingly important. However, traditional methods of pelagic human observers suffer from high costs, low coverage, poor timeliness, and susceptibility to subjective factors. In contrast, the Electronic Monitoring System (EMS) has advantages such as continuous operation under various weather conditions; more objective, transparent, and efficient data; and less interference with fishing operations. This paper shows how the 3DCNN model, LSTM+ResNet model, and TimeSformer model are applied to video-classification tasks, and for the first time, they are applied to an EMS. In addition, this paper tests and compares the application effects of the three models on video classification, and discusses the advantages and challenges of using them for video recognition. Through experiments, we obtained the accuracy and relevant indicators of video recognition using different models. The research results show that when NUM_FRAMES is set to 8, the LSTM+ResNet-50 model has the best performance, with an accuracy of 88.47%, an $F1$ score of 0.8881, and an m_{ap} score of 0.8133. Analyzing the EMS for pelagic fishing can improve China's performance level and management efficiency in pelagic fishing, and promote the development of the fishery knowledge service system and smart fishery engineering.

Keywords: deep learning; jigger behavior identification; squid fishing vessel

Key Contribution: This paper demonstrates the application of three different models (3DCNN, LSTM+ResNet, and TimeSformer) in video-classification tasks for the Electronic Monitoring System (EMS) used in China's pelagic fishing industry. The paper compares the performance of these models and evaluates their effectiveness in video recognition. The results showed that the LSTM+ResNet-50 model achieved the highest accuracy of 88.47%, indicating its potential to improve the management efficiency of pelagic fishing and contribute to the development of smart fishery engineering.



Citation: Song, Y.; Zhang, S.; Tang, F.; Shi, Y.; Wu, Y.; He, J.; Chen, Y.; Li, L. Behavior Recognition of Squid Jigger Based on Deep Learning. *Fishes* **2023**, *8*, 502. <https://doi.org/10.3390/fishes8100502>

Academic Editor: Yang Liu

Received: 15 August 2023

Revised: 30 September 2023

Accepted: 7 October 2023

Published: 8 October 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, with the development of pelagic fishing, the working environment and monitoring of crew members have become increasingly important. Through the process of dataization and informatization, organizations can develop more comprehensive methods and regulations for effective management [1]. According to research conducted by Michelin, traditional methods of sea observation are expensive, have low coverage rates, poor timeliness, and are influenced by subjective factors [2]. In contrast, the Electric Monitoring System (EMS) can operate continuously under various weather conditions, providing objective, transparent, and efficient data for extended periods [3].

In some studies, researchers have attempted to combine traditional human observers and Electronic Monitoring System(EMS) to monitor and obtain information from fishing vessels in a more objective and intuitive manner (Ruiz et al., [4]). In a study conducted by Ruiz et al. [4] that compared the effectiveness of human observers and EMS in tuna purse seine operations, it was found that EMS can reliably determine the number of fishing sets, similar to human observers. The study also revealed that the analysis results from EMS were more reliable when dealing with larger fish catches. Nonetheless, the study also highlighted that the effectiveness of EMS in identifying objects was highly dependent on the location of the cameras, with identification rates ranging from 56.3% to 98.3% in their experiments. Therefore, there is still considerable room for improvement in the data processing of captured information on board using EMS [5].

In recent years, the advancement of deep learning technology has led to a continuous improvement in the accuracy of image recognition. Consequently, it has become possible to fully replace human observers with EMS. For instance, Wang et al. [6] achieved a detection recall rate of 98.3% for target floats and tuna in the electronic monitoring system of tuna fishing vessels using the YOLOv5 network model. Zhang et al. [7] successfully identified different fishing tools including floats, fish tanks, empty fish tanks, and fishing nets on *Scomber japonicus* fishing vessels, with a recognition accuracy ranging from 75% to 96.5%. Wang et al. [8], on the other hand, utilized 3DCNN to recognize the traveling status of fishing vessels, achieving a model accuracy of up to 97% on the validation set. However, solely identifying the presence or absence of targets for crew monitoring is not sufficient. Determining the behavior of the crew, which involves video recognition, is also necessary. Wang et al. addressed this issue by cropping videos into 100 frames and inputting them into a neural network for calculation. Nonetheless, the convolutional kernels used in this calculation method are specifically designed to capture local spatiotemporal information and are incapable of modeling dependencies beyond their receptive fields, resulting in a strong induction bias. Furthermore, training deep networks with a large amount of input data necessitates substantial computational resources.

With the advancement of computer performance, the application of deep learning models in video data has become increasingly widespread. Compared to image data, video data contain more information. In 2012, Ji et al. [9] proposed a method that utilizes 3D convolution in video clips to learn features that exist in both space and time, extending the capabilities of convolutional neural networks (CNNs) into the temporal domain. However, the development of deep learning in the field of video recognition has been slower compared to other areas [10]. It is only in recent years, with the increase in model parameters and the improvement in computing power, that researchers have started applying CNN models in sports videos and human-action recognition, achieving significant results. In 2020, Rafiq et al. [11] compared the performance of various CNN models in video recognition and found that their proposed AlexNet+CNN encoder model performed the best. Furthermore, in 2017, Varol et al. [12] demonstrated the application of Long Short-Term Memory (LSTM) in human-action recognition, highlighting the importance of time sequences in such tasks. Subsequently, in 2021, Zengkai Wang et al. [13] compared LSTM-ResNet and other models on motion datasets and discovered that LSTM+ResNet achieved superior performance.

However, most of the current research focuses on the classification of short videos. For the classification of long videos, the effectiveness of CNN architecture is extremely limited, and researchers have started to explore the use of Transformer architecture networks for long video classification [14]. Compared to single-image-classification tasks, most methods for video classification involve extracting multiple keyframes from the video and using the motion features and temporal features between these keyframes to recognize human behavior in the video [15]. However, current research mostly focuses on the differences brought about by different frame numbers and model sizes, and lacks specific experiments on factors such as frame numbers and model sizes.

Currently, the EMS of ocean-going vessels has not applied deep learning technology to the research of crew behavior recognition in videos. Although scholars have already applied deep learning technology in the field of video recognition, the focus has mainly been on the recognition of fishing equipment and vessel status, with limited research on crew behavior.

This paper examines the application of deep learning techniques in video-recognition tasks, drawing from existing research. Specifically, three deep learning models, namely, 3DCNN [8], LSTM+ResNet [13], and TimeSformer [16], are selected and applied to the EMS for monitoring the behavior data of squid fishing vessel crew members. In order to conduct this study, a squid fishing vessel crew behavior dataset is constructed using data produced by the EMS. The focus lies on analyzing the actions of crew members during fishing processes such as casting and reeling. However, it is important to note that in practical scenarios, the annotation of crew members' video actions may be subjective, and the actions of crew members may extend beyond a single video segment where the operations are performed. Additionally, the presence of numerous similar actions in repetitive tasks presents challenges for the implementation of video-recognition technology in crew member application scenarios. To address these issues, this paper performs tests and comparisons of deep learning algorithms for classifying video segments. The objective is to explore the advantages and challenges associated with the utilization of these three methods for video recognition. Our main contributions are summarized as follows:

1. This study first constructed a dataset on the work behaviors of squid fishing boat crew and proposed a division basis for their actions.
2. This study then utilized deep learning techniques in the fisheries EMS to classify the work behaviors of crew members.
3. This study, for the first time, applied the LSTM-ResNet, TimeSformer models in an offshore fishery EMS and improved the 3DCNN model used by Wang et al. in identifying whether a fishing boat is moving by imitating the ResNet model, allowing it to be applicable and maintain high accuracy on the EMS squid fishing boat crew dataset.
4. Based on previous scholars' research, this study compared the effects of commonly used 3DCNN, LSTM+ResNet, and TimeSformer models under different parameters in the field of video recognition on the EMS squid fishing boat crew dataset. The results indicate that the LSTM+ResNet model performs the best. It also provides a detailed analysis of the performance and reasons for the performance of the three models under different parameters and presents prospects for the future application of deep learning models in offshore EMS crew behavior recognition.

This article is divided into four sections. Section 2 describes the construction of the dataset and introduces the model used in this experiment. Section 3 presents the experimental results. Section 4 comprises a discussion of the experiment and its results. Lastly, Section 5 consists of the conclusions and future prospects of this research.

2. Materials and Methods

2.1. Data Collection

The EMS data used in this study were gathered from the Squid Fishing Vessel. The vessel has a length of 44 m, a width of 7.8 m, a main engine power of 662 kilowatts, and a total tonnage of 350 tons. In total, 30 TB of EMS data was collected from the system. The camera used is the Hikvision DS-2CD7A47EWD-XZS, capturing video at a resolution of 1280 (horizontal) \times 720 (vertical) and a frame rate of 25 fps. The DVR model is the Hikvision DS-7708NX-I4, supporting 8 channels of H.264 and H.265 hybrid video inputs, with a maximum capacity of 8 TB.

Squid fishing vessels catch squid by operating from a fixed position. Each fishing vessel is equipped with a fixed pulley system and fishing lines. Each fishing line is connected to 10 to 20 hooks, spaced approximately 30 to 50 cm apart. Typically, a fishing vessel has a crew of around 28 members, whose main task is catching squid. Due to

the need for individual operation in squid fishing, a marine observer is responsible for observation. Nevertheless, the human observer method suffers from high costs, limited coverage, delayed reporting, and subjective biases. Consequently, an EMS system is required to overcome these challenges.

During the dataset-construction process, we divided the recordings from the EMS system into 5 s segments and marked the position of a fixed pulley separately. As shown in Figure 1, the constructed samples consist of eight images captured in chronological order. This designated area allows for a clearer demonstration of the actions of the crew and the status of the fishing line on the deck.



Figure 1. Sample example.

The crew's behaviors during the fishing process were defined. We categorized the crew's behaviors into 12 types based on the video evidence, and detailed descriptions of these are provided in Table 1. For a better understanding of the crew's work status and job content, 8 frames were evenly extracted from the corresponding videos. These 8 frames clearly depict the crew's current work status and job content. Analyzing the crew's behaviors becomes more accurate through careful observation of these images.

Table 1 presents a comprehensive breakdown of the names, guidelines, and impacts of various action categories. Based on the categorization provided in Table 1, we collected 300 samples for each categorized action from the 30 TB of raw data we gathered. As a result, a total of 3600 segments of 5-s videos were obtained as the primary data for constructing the dataset. To account for variations in human behavior, we ensured that samples were collected from multiple crew members. This approach allows for a more comprehensive analysis of fishing operations and provides a broader representation of the actions involved and the model's predictions of crew fishing behaviors in different positions and conditions.

Table 1. Sample classification criteria.




Serial Number	Category	Behavior Description	Category Image
1	No crew	There are no crew members at the pulley	
2	Idle	The crew is at the pulley but not working	
3	Hands collect the line	The crew manually pulls the line through the pulley	

Table 1. Cont.



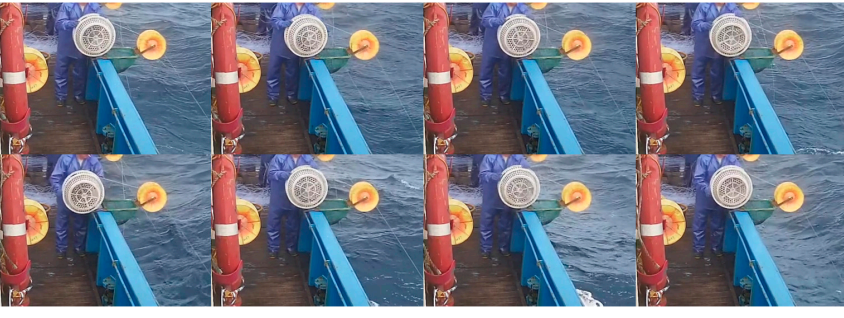
Serial Number	Category	Behavior Description	Category Image
4	Hands release the line	The crew releases the fishing line through their hands and the pulley	
5	Empty hands release the thread	The line is automatically released with the help of ocean current	
6	Use a wire winder to retrieve the cable	The crew uses a wire winder to retrieve the cable	

Table 1. Cont.

Serial Number	Category	Behavior Description	Category Image
7	Use a coil winder to unwind the line	The crew uses a coil winder to unwind the line	
8	Untangling the fishing line	The crew unties a knot	
9	Handling miscellaneous matters	The crew deals with other tasks at the pulley	

Table 1. Cont.

Serial Number	Category	Behavior Description	Category Image
10	Pull up the squid	The crew pulls the squid onto the vessel	
11	The squid is placed in the frame	The crew members put the squid into the net	
12	Receive and release	The crew alternates between pulling and releasing the rope	

By combining data from multiple crew members, we aim to capture the range of human behaviors that occur during fishing operations. These data were extracted from 14 randomly selected days of video recordings, during which five different crew members were observed performing operations at the location. Through a comprehensive analysis of their actions and the squid fishing process, we observed that the majority of actions performed by these five crew members during fishing were quite similar. Additionally, there was significant similarity in the sequence of their actions, with only a few actions displaying slight variations in order. This presents notable difficulties for the model's learning and recognition process.

2.2. Dataset Generation

To accurately recognize fishing behaviors of crew members in different directions and positions, we employed data-augmentation methods to handle data from various positions. Data augmentation increases the diversity of the original data and enhances the model's robustness [17]. We chose the following three data-augmentation methods to handle potential issues like camera blur on the fishing vessel, changes in camera position, and adjustments in crew members' working positions:

1. Add salt-and-pepper noise: Simulate the possible blur of the camera by adding salt-and-pepper noise, allowing the model to learn how to handle blurry images and improve its ability to deal with such situations.
2. Vertical flip: Use the ability to vertically flip images to simulate situations that may arise due to incorrect camera orientation.
3. Horizontal flip: Simulate different camera and crew positions by horizontally flipping the image, which can simulate fishing behaviors of crew members in different positions on the vessel.

Through the above data-augmentation methods, we can better simulate situations that may occur in real fishing environments, improving the accuracy and robustness of the model's predictions of crew fishing behaviors in different positions and conditions.

Table 2 presents examples obtained after applying data-augmentation techniques. Initially, we randomly divided the dataset into training and test sets, maintaining an 8:2 ratio. Subsequently, we implemented three different data-augmentation techniques alternately to augment the training set. As a result, the size of the training set expanded by eight times its original size, yielding a dataset with a total of 23,760 video clips. Within this dataset, the training set comprises 23,040 video clips, while the test set consists of 720 video clips. For dataset annotation, we employed an Excel spreadsheet to label the video categories. The format of the annotation is displayed in Table 3.

Table 2. Data expansion method.




Processed Image			
Processing method	Add salt-and-pepper noise	Vertical flipping	Horizontal flipping

Table 3. Label format.

Video Storage Path.	Category
./path	1

During the subsequent processing, the data are converted into the format “./path 1” and written in the standard data label format for the TimeSformer model in an Excel spreadsheet. Furthermore, this calamari fishing crew behavior dataset is further utilized for other model experiments in this paper.

2.3. Network Structure

2.3.1. 3DCNN

The 3D Convolutional Neural Network (3DCNN) is a deep learning model specifically designed for processing 3D data. It extends the 2D Convolutional Neural Network (CNN) to accommodate the characteristics of 3D data. In our constructed squid fishing dataset, each piece of sample data consists of a segment of a video. To process this data, we incorporate the time dimension as the third dimension, in addition to length and width, and input the sampled image data from a video segment into the 3DCNN simultaneously. The 3DCNN utilized in this paper is an altered version of the basic 3DCNN, with modifications to the dimensions of its output layer. The network structure is illustrated in Figure 2.

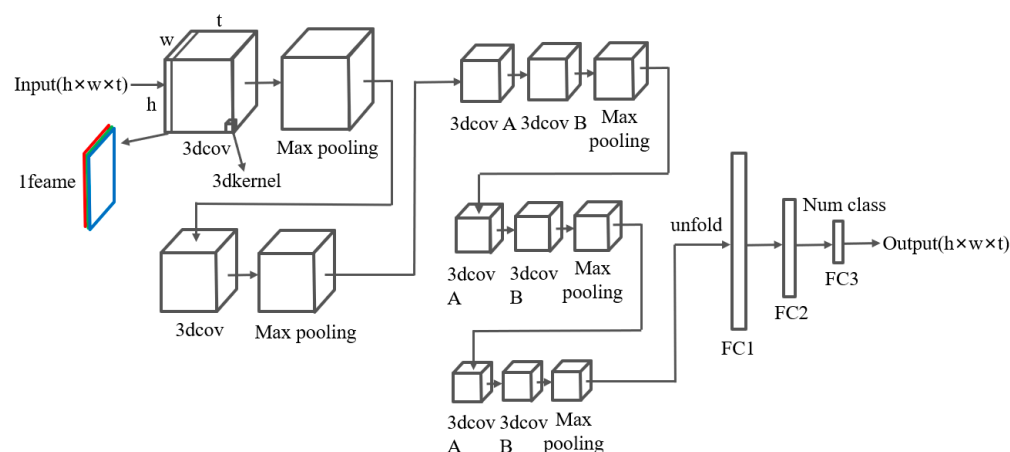


Figure 2. Diagram of 3DCNN architecture.

Figure 2 illustrates the detailed architecture of the 3D CNN, which consists of multiple layers designed to capture different levels of features from the input video data. The initial two layers consist of 3D convolutional operations and MaxPooling structures, which aim to capture intricate details. Subsequently, three additional layers with 3D convolutions and MaxPooling are employed to gradually extract larger features.

The 3D Convolutional Neural Network (3DCNN) model is composed of convolutional, pooling, and fully connected layers. The convolutional layers play a crucial role in extracting spatial features from the input data, while the pooling layers serve to reduce the size of the data. The fully connected layers are responsible for mapping the extracted features to the appropriate output class. Specifically, the model employs multiple 3D convolutional layers, denoted as 3DCov. The kernel size of these convolutional layers determines the receptive field size, and the input and output channel numbers indicate the depth of each convolutional layer.

In 3DCNNs, the temporal sequence is directly learned as a receptive field. By stacking multiple convolutional layers, higher-level features can be gradually extracted by the model. The model utilizes pooling layers (MaxPooling) after the convolutional layers. Downsampling the feature maps further reduces their size while preserving important features. Finally, the features obtained from the pooling layers are mapped to the final output classes through fully connected layers (FC). Non-linear transformations are introduced and the model’s expressive power is increased by adding activation functions (ReLU) after each linear layer.

Prior to training the model, it is advantageous to initialize the convolutional layers using the Xavier Gaussian method, as it significantly enhances the model's stability and convergence speed during training.

2.3.2. LSTM-ResNet

The LSTM-ResNet model, a commonly used recurrent neural network for modeling sequential data, was utilized in this study. We chose ResNet as the encoding model, based on the research conducted by Zengkai Wang et al. [13], which showed it to have the highest accuracy. Initially, a pre-trained ResNet network is utilized to extract features from each frame of the image [18]. To align with the input requirements of the LSTM model, an additional convolutional layer is added at the end of the ResNet network to reduce the dimension of the ResNet output to 128. The LSTM network is defined with an input dimension of 128 and a hidden layer size of 512. By analyzing the sequence of feature vectors extracted by ResNet, the LSTM outputs the hidden state of the last time step. Finally, two fully connected layers are employed to output the prediction result. The structure of this model is depicted in Figure 3.

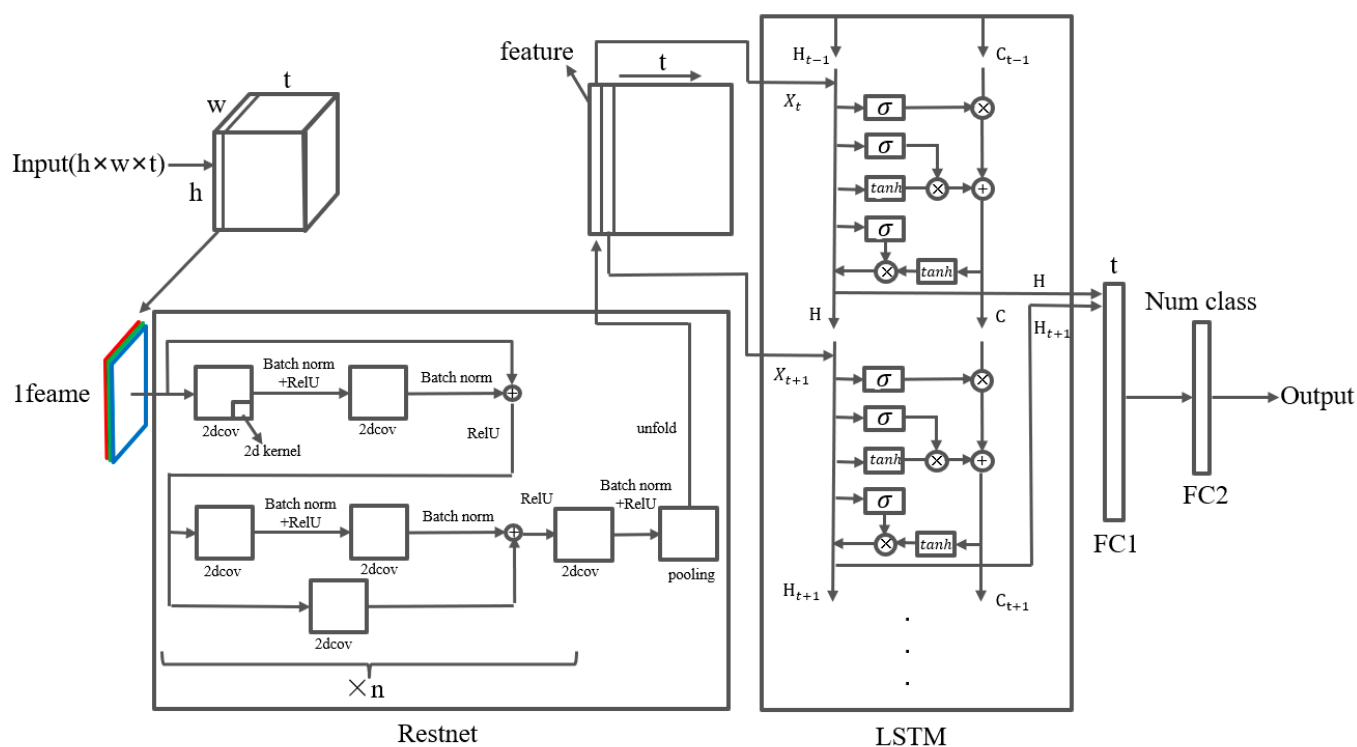


Figure 3. LSTM-ResNet Network Architecture Diagram.

The structure of the neural network model, which is a combination of ResNet and LSTM, is illustrated in Figure 3. This model is a combination of ResNet and LSTM. The ResNet component extracts features from input images and consists of multiple convolutional layers that extract high-level semantic features to enhance the network's understanding of images. ResNet is used as a feature extractor, with parameters retrained and fully connected layers removed, retaining only the convolutional part. After the convolutional part, a convolutional layer is added to reduce the number of channels in the feature map from 2048 to 128 to decrease the dimensionality of the input to LSTM. The LSTM component is primarily responsible for modeling temporal features, capturing long-term dependencies in the image sequence. Through the LSTM model, the network can effectively process temporal information in the image sequence. Finally, a fully connected layer is used to further map and process the output of LSTM, mapping it to the target class space to obtain the final prediction result.

This study proposes a combination structure of ResNet and LSTM to harness the power of both convolutional and recurrent operations. This approach enhances the network's modeling capability and improves its understanding of image sequences, consequently boosting network performance [13].

2.3.3. TimeSformer

TimeSformer, a video-classification model proposed by Facebook researchers Bertasius et al. in 2021 [16], is based on Transformers. Initially introduced for language translation tasks [19], Transformers consist of two separate modules: an encoder and a decoder. Each module is composed of multiple layers of Transformers that are stacked on top of each other. A Transformer layer is made up of various components, including Multi-Head Attention, Norm, and Feed-Forward layers, as depicted in Figure 4. The encoder generates representations of input sequences, which are then processed by the decoder to translate them into the target language [20].

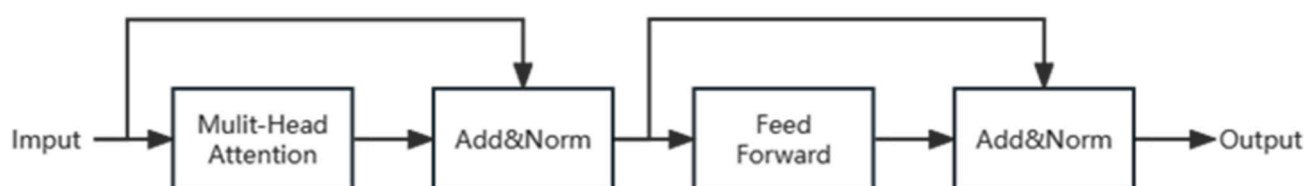


Figure 4. Transformer layer architecture diagram.

For image-recognition tasks, the most commonly used method for the encoder is to use a large pre-trained embedding network, which encodes the images using a pre-trained image transformer (usually ViT [21]), generating a set of feature vectors. This method further improves training efficiency because these networks have already undergone pre-training [22]. Then, the TimeSformer layer is trained for downstream tasks based on these feature vectors. Compared to training from scratch in an end-to-end manner, using State-of-the-Art (SOTA) models is usually easier and more efficient, as these carefully tuned models perform well on some supervised tasks [14].

TimeSformer's design concept involves treating each frame image in a video sequence as a time step and converting it into a set of feature vectors usable by the Transformer network. The feature vectors are concatenated and linearly mapped to the embedding vector $Z(l-1)$, which is subsequently fed into the TimeSformer block based on the Transformer layer for learning purposes. TimeSformer effectively captures long-term dependencies in video sequences and demonstrates outstanding performance in various video-classification tasks [16].

The TimeSformer block consists of multiple Transformer layers, while incorporating time and spatial attention, as illustrated in Figure 5. Each layer consists of several self-attention heads and a feed-forward neural network [23]. Within each Transformer layer, the input embedding vector sequence is independently processed in the multi-head self-attention mechanism and the feed-forward neural network. The multi-head self-attention mechanism captures long-term temporal dependencies in the input sequence, while the feed-forward neural network applies non-linear transformations to each embedding vector at each time step. The output of each Transformer layer is then passed to the subsequent layer, until the output of the final Transformer layer is fed into the global average pooling layer. This pooling layer generates the ultimate representation of the video sequence [24].

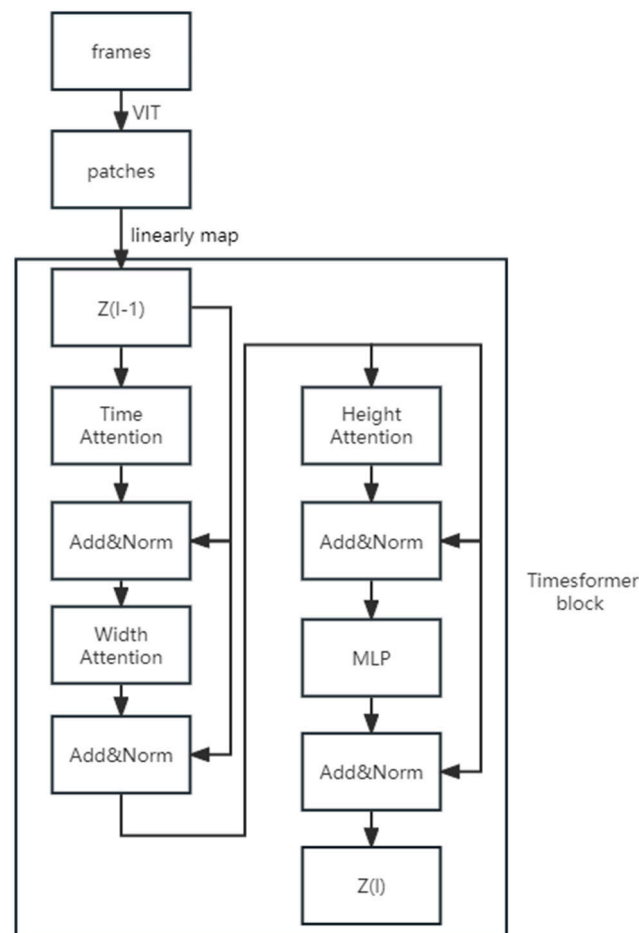


Figure 5. TimeSformer block structure with temporal and spatial attention added.

3. Results

3.1. Evaluation Indicators

For the evaluation of video-classification models, several commonly used metrics are as follows. These metrics play a crucial role in assessing the performance of video-classification models and determining the most effective models for specific tasks.

1. *Top – 1 acc* and *Top – 5 acc*: *Top – 1 acc* represents the ratio of correctly predicted videos (N_1) to the total number of videos (N). *Top – 5 acc* represents the ratio of correctly predicted videos (N_1) to the total number of videos (N), where any of the top five predicted results are considered correct.

$$Top - 1\ acc = \frac{N_1}{N}, \quad (1)$$

$$Top - 5\ acc = \frac{N_5}{N}, \quad (2)$$

2. *Mean Average Precision (m_{ap})*: m_{ap} is a widely used evaluation metric for assessing the performance of a model in multi-class classification tasks. It represents the average precision across all classes. The average precision for each class measures the model's precision on that class. It is calculated by computing the cross-entropy between the model's predicted confidence scores and the true labels [25].

$$m_{ap} = \left(\frac{1}{N} \right) * \sum_{i=1}^N AP(i), \quad (3)$$

where N represents the number of samples in the test set, indicating the targets that the object-detection algorithm is required to detect. $AP(i)$ represents the average precision of the i -th target and serves as a metric to evaluate the accuracy of the object-detection algorithm for that specific target.

3. $F1$: The $F1$ Score is a measure that represents the weighted average of precision and recall, and it is commonly used to assess the performance of binary classification tasks. Calculation of the $F1$ Score involves counting the occurrences of true positives, false positives, and false negatives [26].

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \quad (4)$$

where *Precision* represents the ratio of correctly predicted positive samples to the total number of predicted positive samples. The *Precision* parameter in the formula represents the model's accuracy, which is typically a value between 0 and 1. *Recall* represents the ratio of correctly identified positive samples to the total number of positive samples. The *Recall* parameter in the formula represents the model's recall rate, indicating the proportion of correctly recognized positive samples out of all positive samples.

The weighted-average method is employed in this article for multi-class tasks. This method estimates the weights by considering the true distribution proportions of the classes. Subsequently, the *Precision* and *Recall* values for each class are multiplied by their respective weights and aggregated. This approach addresses the issue of class imbalance. The formulas for calculating weighted *Precision* and weighted *Recall* are presented below:

$$weighted - Precision = \sum_{i=1}^N Precision(i) \times weighted(i), \quad (5)$$

$$weighted - Recall = \sum_{i=1}^N Recall(i) \times weighted(i), \quad (6)$$

The calculation formula for the $F1$ score, after the implementation of the weighted-average method, is as follows:

$$F1 = \frac{2 \times weighted - Precision \times weighted - Recall}{weighted - Precision + weighted - Recall}. \quad (7)$$

3.2. Experimental Results

During the experiment, we trained and validated the three models discussed earlier. For the software environment setup, we utilized Python 3.9 programming language, PyTorch deep learning framework, CUDA 11.4 operating platform, and Anaconda environment management software. We performed this experiment on a Ubuntu 18.04.5 LTS system equipped with a Tesla V100S 32G graphics card.

In the examination of the TimeSformer model, a study conducted by Bertasius et al. [16] revealed its superior performance in long-term sequence modeling. Based on this finding, we opted to vary the NUM_FRAMES parameter to observe the impact of sampling rate on the TimeSformer-L model for the sampled videos. For the 3DCNN model, we conducted a comparison between two models of different parameter sizes, S (53 MB) and L (119 MB), in order to target distinct receptive fields [18,27]. Additionally, in this paper, we compared the parameter sizes of the LSTM-ResNet model and the experimental results of NUM_FRAMES. We present the experimental results of these models on the squid fishing dataset in Table 4.

Table 4. Experimental results.

Model		Evaluation Criteria				
	Parameter size	NUM_FRAMES	<i>Top – 1 acc</i>	<i>Top – 5 acc</i>	<i>F1</i>	<i>m_{ap}</i>
3DCNN	S	8	88.06%	100.00%	0.8732	0.7835
	L		87.08%	100.00%	0.8622	0.7763
	Feature-extraction model	NUM_FRAMES	<i>Top – 1 acc</i>	<i>Top – 5 acc</i>	<i>F1</i>	<i>m_{ap}</i>
LSTM	ResNet-50	8	88.75%	100.00%	0.8865	0.8037
		16	82.36%	100.00%	0.8365	0.7290
		32	75.69%	98.05%	0.7323	0.5973
	ResNet-152	8	79.58%	98.33%	0.8037	0.6743
		16	69.17%	96.80%	0.6854	0.5027
		32	63.47%	96.25%	0.6184	0.4736
		NUM_FRAMES	<i>Top – 1 acc</i>	<i>Top – 5 acc</i>	<i>F1</i>	<i>m_{ap}</i>
TimeSformer-L		8	80.83%	100%	0.8632	0.7275
		16	79.72%	100%	0.8250	0.6630
		32	78.19%	100%	0.8367	0.6179

Figure 6 shows the loss-reduction curves of the three models during the training process. Figure 6a illustrates the loss-reduction curve of the 3DCNN model during training. Figure 6b depicts the loss-reduction curve of the LSTM-ResNet model during training. Figure 6c shows the loss-reduction curve of the TimeSformer-L model during training. The observed curves provide insight into the gradual decrease in loss for each model during training. These graphs are valuable references for evaluating the effectiveness and performance of the models.

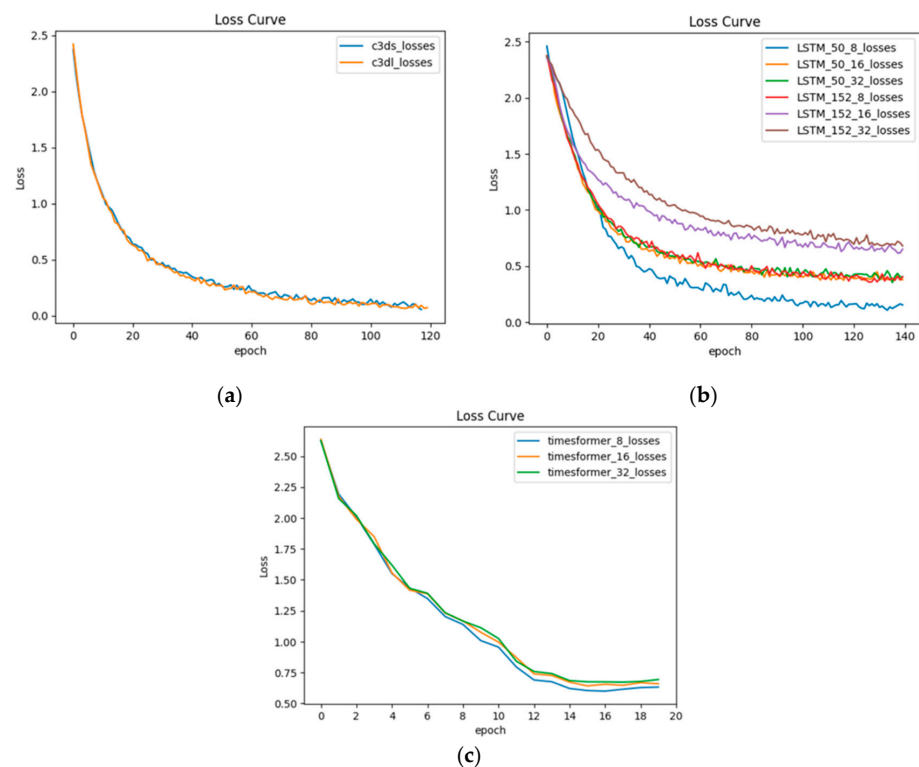


Figure 6. Training loss curve graph of (a) the 3DCNN model, (b) the LSTM-ResNet model, and (c) the TimeSformer-L model.

To assess the resilience of the model, we obtained 240 additional video clips from a different location, which were used as the validation set. From each category, we carefully selected 20 video clips as representative samples. The specific details of the sample data are provided in Figure 7.



Figure 7. Sample example of the validation set.

We conducted experiments on a validation set that is completely independent from the training and testing sets for the three models: 3DCNN-S-NUM_FRAMES-8, LSTM-ResNet-50-NUM_FRAMES-8, and TimeSformer-L-NUM_FRAMES-8. The results of these experiments are presented below.

Table 5 presents the experimental results on the validation set for the best-performing parameters of the three models tested on the test set. The evaluation criteria for the models remain the same: $Top-1\ acc$, $Top-5\ acc$, $F1$, and m_{ap} .

Table 5. Experimental results on the validation set.

Model	$Top-1\ acc$	$Top-5\ acc$	$F1$	m_{ap}
3DCNN-S-NUM_FRAMES-8	83.75%	98.33%	0.7809	0.6856
LSTM-ResNet-50-NUM_FRAMES-8	84.58%	98.75%	0.8137	0.7563
TimeSformer-L-NUM_FRAMES-8	74.16%	99.58%	0.7636	0.6453

4. Discussion

4.1. Data Collection

This study initially examined the conduct of the crew and the condition of the fishing lines on the deck, resulting in the creation of an EMS dataset. During the dataset-construction process, a fixed pulley position was chosen as a boundary region to more accurately exhibit the crew's actions and the fishing lines' condition. In comparison with the EMS ship behavior dataset developed by Shuxian Wang et al. [28,29], comprising nine distinct fishing vessel behaviors, the dataset constructed in this paper mostly demonstrates similar sequences of fishing actions, with only a few actions varying in their order. Consequently, this poses certain challenges for the model's learning and recognition.

In terms of sampling frequency, Wang et al. [8] conducted a study where their 3DCNN extracted 100 frames from each 30 min video as input. More specifically, they opted to sample one frame every 18 s. Given the repetitive nature and high repetition rate of crew actions, we established the duration of each sample to be 5 s. Different sampling frequencies (8, 16, and 32) were employed to construct the temporal sequence image dataset of input samples. By utilizing lower sampling frequencies, we could precisely capture the action details in the process of squid fishing, thereby extracting features with greater accuracy [30]. Furthermore, selecting a shorter sample duration enables us to concentrate more extensively on specific actions, reducing interference from irrelevant actions in the samples.

By observing and analyzing the actions of the crew and the state of the fishing line, researchers can gain a deeper understanding of the EMS's performance. The construction

process of the dataset offers a standardized foundation for assessing and comparing the learning and recognition performance of various models. However, the model might face challenges in distinguishing minority actions due to the high similarity in the order of most fishing actions. Therefore, future research should investigate potential solutions to enhance the model's accuracy and robustness in distinguishing these actions [31].

During the dataset-construction process, several data-augmentation techniques were employed to enhance the dataset's diversity and generalization ability. This involved applying operations such as salt-and-pepper noise, vertical flipping, and horizontal flipping to augment the image data. Salt-and-pepper noise, a common image noise, was utilized to simulate various environmental conditions and interferences encountered during maritime navigation, including sea fog and rainy weather. By introducing salt-and-pepper noise, the dataset could encompass samples under different specific conditions, thus improving the model's adaptability in rainy and foggy weather. As a result, the model achieves stability and robustness in practical applications. In addition to incorporating environmental noise, vertical and horizontal flipping operations were performed to simulate different camera placements in ship operations. These data-augmentation techniques made the dataset more comprehensive and capable of handling various environmental and operational conditions. With such a dataset, it becomes possible to better evaluate and compare the performance of different models in recognizing crew actions and fishing line states.

4.2. Analysis of Model Performance

The 3DCNN is a conventional video-classification model that employs a 3D convolutional neural network to extract spatial and temporal features from video sequences. Based on the findings presented in Table 4, it is evident that the 3DCNN-S model outperforms the 3DCNN-M model in terms of $Top - 1\ acc$, $Top - 5\ acc$, $F1\ score$, and m_{ap} value. This suggests that augmenting the parameters of the 3DCNN model results in reduced accuracy, likely attributable to overfitting. Nevertheless, an examination of the loss curves ($c3ds_losses$ and $c3dl_losses$) depicted in Figure 6 indicates no substantial divergence in the learning pace and ultimate loss magnitude between the 3DCNN-S and 3DCNN-L models. Hence, it can be deduced that augmenting the parameter count failed to enhance the performance of the 3DCNN model and further implies that the learning capability of the 3DCNN-S model adequately addresses the squid fishing dataset established in this investigation.

In contrast to Wang et al.'s [8] 3DCNN, the 3DCNN in this study incorporates MaxPooling as the pooling layer choice. This enhancement enables the network to decrease its emphasis on details and, instead, concentrate more on individuals' actions in videos, thereby improving classification accuracy and effectiveness. The input data are downsampled using MaxPooling, thereby reducing the size of feature maps and, consequently, reducing computation and memory. Moreover, MaxPooling can extract salient features by selecting the most prominent features in the feature map for pooling, effectively capturing the key actions of people in videos. This feature selection and downsampling operation help the network better focus on individuals' actions, reducing interference from details and enhancing classification accuracy. Additionally, MaxPooling also possesses translation invariance, meaning that features in the feature map can be accurately captured regardless of their position. This translation invariance is crucial for recognizing human actions; individuals may appear in various positions in different video frames. However, their actions should remain consistent. By employing MaxPooling, the network can disregard the position information in the feature map and solely focus on the presence or absence of features, thereby improving its ability to recognize actions.

The LSTM model is a classic recurrent neural network utilized for processing sequential data [32]. In our experiment, we employed the LSTM model to extract temporal features from each frame of the ResNet model and input them into a fully connected layer for classification. Zengkai Wang et al. [13] have confirmed that LSTM+ResNet-152 performs the best on a publicly available sports dataset. In this study, we conducted

experiments on LSTM+ResNet-50 and LSTM+ResNet-152 with varying parameters on the squid fishing dataset that we constructed. Our findings reveal that LSTM+ResNet-50 outperforms LSTM+ResNet-152 when NUM_FRAMES=8. Based on the experimental results presented in Table 4, it is evident that the LSTM-ResNet-50 model performs better than the LSTM-ResNet-152 model when the NUM_FRAMES value is the same. Increasing the NUM_FRAMES for the ResNet-50 and ResNet-152 feature-extraction models leads to a decrease in both accuracy and recall rates of the classification. Figure 6 demonstrates that during the training phase, when NUM_FRAMES=8, the loss curve exhibits a significantly steeper slope, and the loss value is lower compared to higher NUM_FRAMES values. Generally, utilizing ResNet-50 as the feature-extraction model outperforms the employment of ResNet-152. This suggests that a larger model is not necessarily superior when dealing with specific datasets. This could be attributed to the strong capabilities of ResNet-152 in extracting image details. However, in the case of action-recognition tasks, judgment and classification heavily depend on the overall sequence of the crew's actions. Excessive focus on details for this type of task can potentially impact recognition accuracy, leading to a decrease in precision. The selection of NUM_FRAMES and the size of ResNet-152 have a substantial impact on the model's performance. Consequently, careful adjustment of hyperparameters is necessary when utilizing the LSTM-ResNet model to attain optimal performance.

The TimeSformer model is built upon Transformer architecture. According to the experimental results in Table 4, TimeSformer achieves a maximum $Top - 1$ acc of 80.01% on the squid fishing dataset. Bertasius et al. [16] achieved accuracies of 80.7% and 82.2% using TimeSformer-L on the Kinetics-400 and Kinetics-600 datasets, respectively (Bertasius et al., 2020). Despite the comparatively smaller size of the squid fishing dataset compared to the Kinetics-400 and Kinetics-600 datasets, TimeSformer demonstrates the capacity to learn and classify various actions, thus highlighting its ability to attain accurate classifications in short, similar videos. The four different configurations of the NUM_FRAMES parameter have a minimal effect on the $Top - 1$ acc and $Top - 5$ acc, as both metrics consistently maintain high values. However, as the NUM_FRAMES parameter increases, there is a slight decrease in the m_{ap} metric. This phenomenon may be attributed to the model's heightened focus on details and motion in the video sequence, which may not be decisive for certain categories of target objects. Based on the experimental results in Ref. [33], we can conclude that increasing the NUM_FRAMES parameter has a negligible impact on the $Top - 1$ acc and $Top - 5$ acc accuracies in the TimeSformer model, but it does affect the m_{ap} and $F1$ metrics to some extent. Furthermore, the training loss curve of the TimeSformer model (Figure 6) demonstrates a significant increase in the final stable loss value as the NUM_FRAMES parameter increases, suggesting that an appropriate NUM_FRAMES parameter can enhance model performance. To select an appropriate setting for the NUM_FRAMES parameter, one can comprehensively consider the aforementioned metrics, catering to specific requirements.

Based on the results obtained from the validation set, the accuracy and $F1$ scores of the three models have shown a slight decrease. This indicates that these models exhibit robustness in recognizing crew actions and have effectively learned the characteristics of various crew actions. However, due to variations in behavior habits and working environments among crew members in different positions, this has led to a decrease in the accuracy of all three models. To further enhance the performance of the models, collecting a larger-scale dataset is necessary, as well as training more stable models to mitigate this issue. Collecting more data enables a better understanding of the behavior habits and characteristics exhibited by crew members in different positions and environments, thus resulting in models with improved generalization ability.

5. Conclusions

This study analyzes EMS data records from squid fishing to categorize the crew members' workflow into 12 distinct behaviors. Using these behaviors as a basis, we constructed a dataset of EMS records in squid fishing and evaluated the performance of

three models: 3DCNN, LSTM-ResNet, and TimeSformer. Upon comparing the experimental results, the following conclusions were made:

1. In this article, the squid fishing dataset was constructed. The LSTM-ResNet-50 model, with NUM_FRAMES set to 8, achieved an accuracy of 88.47%, an $F1$ score of 0.8881, and an m_{ap} score of 0.8133. In comparison to other parameters, increasing the depth of the ResNet model did not improve the overall performance of the LSTM-ResNet model.
2. The 3DCNN-S model outperformed the 3DCNN-L model in terms of the 3DCNN architecture. Nevertheless, the small-scale 3DCNN-S model exhibited a negligible difference compared to the 3DCNN-L model, illustrating its competence in effectively handling the classification task of the squid fishing dataset built in this study.
3. In the TimeSformer model, the NUM_FRAMES parameter had minimal impact on the $Top - 1\ acc$ and $Top - 5\ acc$, but it did influence the m_{ap} and $F1$ metrics. Nevertheless, the TimeSformer model exhibited poor performance on the squid fishing dataset, which consists of short videos. Nonetheless, it possesses an advantage in training speed compared to the LSTM-ResNet and 3DCNN models. The transformer architecture has significant potential for video-recognition applications.

Author Contributions: Y.S. (Yifan Song): Conceptualization, Methodology, Software, Investigation, Writing—original draft, Writing—review and editing. S.Z.: Conceptualization, Methodology, Writing—review and editing. Y.S. (Yongchuang Shi): Resources, Data curation. F.T.: Formal analysis, Investigation. Y.W.: Visualization, Formal analysis. J.H.: Investigation, Resources, Supervision. Y.C.: Formal analysis, Data curation. L.L.: Formal analysis, Data curation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Laoshan Laboratory under Grant No. LSKJ202201804; National Natural Science Foundation of China under Grant No. 61936014.

Institutional Review Board Statement: The focus of this study is the identification of crew work behaviors and does not involve any commercial interests. The video surveillance data used in this study were obtained through legal means and did not interfere with the normal public order. Therefore, based on the “Notice on the Issuance of Ethical Review Methods for Human Life Science and Medical Research”, this study does not require ethical approval.

Data Availability Statement: As the data for this study are still being further collected and processed, a complete dataset is not available at this time. We recognize the importance of data and understand that other researchers may be interested in our study and would be happy to provide further support and assistance if they require data support or would like to communicate with us. We can be contacted by email or other appropriate means and will be happy to support other researchers with data or related research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, C.; Chen, X. Analysis of the research status in the field of offshore fishery based on bibliometrics. *Mar. Freshw. Fish.* **2020**, *175*, 108–119. [\[CrossRef\]](#)
2. Michelin, M.; Elliott, M.; Bucher, M.; Zimring, M.; Sweeney, M. *Catalyzing the Growth of Electronic Monitoring in Fisheries: Building Greater Transparency and Accountability at Sea*; California Environmental Associates: San Francisco, CA, USA, 2018.
3. Zhang, J.; Zhang, S.; Fan, W. Research on target detection of Japanese anchovy purse seine based on improved YOLOv5 model. *Mar. Fish.* **2023**, 1–15. [\[CrossRef\]](#)
4. Ruiz, J.; Batty, A.; Chavance, P.; McElderry, H.; Restrepo, V.; Sharples, P.; Santos, J.; Urtizbarea, A. Electronic monitoring trials on in the tropical tuna purse-seine fishery. *ICES J. Mar. Sci.* **2015**, *72*, 1201–1213. [\[CrossRef\]](#)
5. Pei, K.; Zhang, J.; Zhang, S.; Sui, Y.; Zhang, H.; Tang, F.; Yang, S. Spatial distribution of fishing intensity of canvas stow net fishing vessels in the East China Sea and the Yellow Sea. *Indian J. Fish.* **2023**, *70*, 1–9. [\[CrossRef\]](#)
6. Wang, S.; Zhang, S.; Zhu, W.; Sun, Y.; Yang, L.; Sui, J.; Shen, L.; Shen, J. Target detection application of deep learning YOLOV5 network model in electronic monitoring system for tuna longline fishing. *J. Dalian Ocean. Univ.* **2021**, *36*, 842–850. [\[CrossRef\]](#)
7. Zhang, J.; Wang, S.; Zhang, S.; Tang, F.; Fan, W.; Yang, S.; He, R. Research on target detection of engraulis japonicus purse seine based on improved model of YOLOv5. *Front. Mar. Sci.* **2022**, *9*, 933735. [\[CrossRef\]](#)
8. Wang, S.; Zhang, S.; Liu, Y.; Zhang, J.; Sun, Y.; Yang, Y.; Tang, F. Recognition on the working status of *Acetes chinensis* quota fishing vessels based on a 3D convolutional neural network. *Fish. Res.* **2022**, *248*, 106226. [\[CrossRef\]](#)

9. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 221–231. [\[CrossRef\]](#)
10. Wu, W.; Sun, Z.; Ouyang, W. Revisiting classifier: Transferring vision-language models for video recognition. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 2847–2855. [\[CrossRef\]](#)
11. Rafiq, M.; Rafiq, G.; Agyeman, R.; Choi, G.S.; Jin, S.I. Scene classification for sports video summarization using transfer learning. *Sensors* **2020**, *20*, 1702. [\[CrossRef\]](#)
12. Varol, G.; Laptev, I.; Schmid, C. Long-term temporal convolutions for action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1510–1517. [\[CrossRef\]](#)
13. Wang, Z.; Li, P.; Chen, B.; Ye, L. Research on sports video classification based on CNN-LSTM encoder-decoder network. *J. Jiaxing Univ.* **2021**, *33*, 25–34.
14. Selva, J.; Johansen, A.S.; Escalera, S.; Nasrollahi, K.; Moeslund, T.B.; Clapés, A. Video Transformers: A Survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–10.
15. Brattoli, B.; Tighe, J.; Zhdanov, F.; Perona, P.; Chalupka, K. Rethinking zero-shot video classification: End-to-end training for realistic applications. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4613–4623.
16. Bertasius, G.; Wang, H.; Torresani, L. Is space-time attention all you need for video understanding? *ICML* **2021**, *2*, 4.
17. Zhi, H.; Yu, H.; Li, S.; Gao, C.; Wang, Y. A video classification method based on deep metric learning. *J. Electron. Inf. Technol.* **2018**, *40*, 2562–2569.
18. Bo, P.; Tang, J.; Cai, X.; Xie, J.; Zhang, Y.; Wang, Y. High Altitude Video Traffic State Prediction Based on 3DCNN-DNN. *J. Transp. Syst. Eng. Inf. Technol.* **2020**, *20*, 39–46. [\[CrossRef\]](#)
19. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; Volume 30.
20. Rakhimov, R.; Volkhonskiy, D.; Artemov, A.; Zorin, D.; Burnaev, E. Latent video transformer. In Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Virtual, 8–10 February 2021; pp. 101–112.
21. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2010**, arXiv:2010.11929.
22. Kim, D.; Xie, J.; Wang, H.; Qiao, S.; Yu, Q.; Kim, H.S.; Chen, L.C. Tubeformer-deeplab: Video mask transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13914–13924.
23. Gabeur, V.; Sun, C.; Alahari, K.; Schmid, C. Multi-modal transformer for video retrieval. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Part IV 16. Springer International Publishing: Cham, Switzerland, 2020; pp. 214–229.
24. Girdhar, R.; Carreira, J.; Doersch, C.; Zisserman, A. Video action transformer network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 244–253.
25. Panda, R.; Chen, C.F.; Fan, Q.; Sun, X.; Saenko, K.; Oliva, A.; Feris, R. Adamml: Adaptive multi-modal learning for efficient video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 7576–7585.
26. Liao, J.; Wang, S.; Zhang, X.; Liu, G. 3d convolutional neural networks based speaker identification and authentication. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 2042–2046.
27. Liaojian Country. *Research on Speaker Recognition and Verification Technology Based on 3DCNN Lip Reading Features*; Shanghai Jiao Tong University: Shanghai, China, 2019. [\[CrossRef\]](#)
28. Affandi, A.; Sumpeno, S. Clustering spatial temporal distribution of fishing vessel based LON VMS data using K-means. In Proceedings of the IEEE 2020 3rd International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 24–25 November 2020; pp. 1–6.
29. Gilman, E.; Castejón, V.D.; Loganimoce, E.; Chaloupka, M. Capability of a pilot fisheries electronic monitoring system to meet scientific and compliance monitoring objectives. *Mar. Policy* **2020**, *113*, 103792. [\[CrossRef\]](#)
30. Ullah, I.; Chen, J.; Su, X.; Esposito, C.; Choi, C. Localization and Detection of Targets in Underwater Wireless Sensor Using Distance and Angle Based Algorithms. *IEEE Access* **2019**, *7*, 45693–45704. [\[CrossRef\]](#)
31. Su, X.; Ullah, I.; Liu, X.; Choi, D. A Review of Underwater Localization Techniques, Algorithms, and Challenges. *J. Sens.* **2020**, *2020*, 6403161. [\[CrossRef\]](#)
32. Teng, J.B.; Kong, W.W.; Tian, Q.X.; Wang, Z.Q.; Li, L. Multi-channel attention mechanism text classification model based on CNN and LSTM. *Comput. Eng. Appl.* **2021**, *57*, 154–162.
33. Wu, Z.; Yao, T.; Fu, Y.; Jiang, Y.G. Deep learning for video classification and captioning. *Front. Multimed. Res.* **2017**, *17*, 3–29.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.