Computers and Electronics in Agriculture 167 (2019) 105087

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

A spatio-temporal recurrent network for salmon feeding action recognition from underwater videos in aquaculture

Håkon Måløy[a,*], Agnar Aamodt[a], Ekrem Misimi[b]

[a] NTNU, Department of Computer Science, Sem Sælandsvei 9, Trondheim, Norway
[b] SINTEF Ocean, Brattørkaia 17C, 7010 Trondheim, Norway

ARTICLE INFO

Keywords:
Fish action/behaviour recognition
Fish feeding
Aquaculture
Convolutional neural network
Recurrent neural network
Action recognition
Video analysis
Optical flow

ABSTRACT

Recent developments have shown that Deep Learning approaches are well suited for Human Action Recognition. On the other hand, the application of deep learning for action or behaviour recognition in other domains such as animal or livestock is comparatively limited. Action recognition in fish is a particularly challenging task due to specific research challenges such as the lack of distinct poses in fish behavior and the capture of spatio-temporal changes. Action recognition of salmon is valuable in relation to managing and optimizing many aquaculture operations today such as feeding, as one of the most costly operations in aquaculture. Inspired by these application domains and research challenges we introduce a deep video classification network for action recognition of salmon from underwater videos. We propose a Dual-Stream Recurrent Network (DSRN) to automatically capture the spatio-temporal behavior of salmon during swimming. The DSRN combines the spatial and motion-temporal information through the use of a spatial network, a 3D-convolutional motion network and a LSTM recurrent classification network. The DSRN shows an accuracy that is suitable for industrial use in prediction of salmon behavior with a prediction accuracy of 80%, validated on the task of predicting Feeding and NonFeeding behavior in salmon at a real fish farm during production. Our results show that the DSRN architecture has high potential in feeding action recognition for salmon in aquaculture and for applications domains lacking distinct poses and with dynamic spatio-temporal changes.

1. Introduction

Feeding is an important part of the salmon breeding process. The feed is approximated to account for over half of the total fish farming costs in Norway (Fiskeridirektoratet, 2016). Therefore big financial gains can be made by optimizing the feeding process. Additionally, environmental gains from a reduction in feed spillage is also a promising outcome from an optimized feeding process. Traditional feeding processes are labor-intensive processes requiring manual observation and maneuvering of cameras within breeding cages. They rely on human expertise and the feeding schedule is largely dependent on the feeder responsible for the current feeding process. The use of automatic non-intrusive video-based methods has the potential to reduce the need for human labor and increase the welfare of salmon in breeding cages through more stable feeding cycles across all sites and improved feeding schedules based on the fish behavior. By using fish behavior rather than the amount of feed falling to the bottom of the cage as an indicator for when to stop the feeding process, the system can stop the feeding process at a more appropriate time. This can result in a reduced environmental impact and reduced feed costs as less feed is wasted.

Deep Learning is a sub-field in machine learning that has shown great promise in many forms of data analysis recently. Deep learning is the use of deep neural networks that use multiple processing layers to learn representations of data with multiple levels of abstraction (LeCun et al., 2015). These methods have shown remarkable improvements in speech recognition, image recognition and machine translation. Recently, deep learning has also been applied to agriculture in applications from classification of weeds to fruits counting (Kamilaris and Prenafeta-Boldú, 2018). The aquaculture industry has also seen the benefits from such methods through the use of deep learning and Support Vector Machines (SVMs) for segmentation of blood defects in cod fillet (Misimi et al., 2017).

In this work, we present an approach applicable to perform automatic action recognition in fish using deep learning, as shown in Fig. 1. Our approach is motivated by approaches from human action recognition, but modified for the new domain. The fish action recognition domain differs from the human action recognition domain in several fundamental ways. The human domain includes a vast variety of

* Corresponding author.
E-mail addresses: hakon.maloy@ntnu.no (H. Måløy), agnar@ntnu.no (A. Aamodt), ekrem.misimi@sintef.no (E. Misimi).

https://doi.org/10.1016/j.compag.2019.105087
Received 11 July 2019; Received in revised form 31 October 2019; Accepted 1 November 2019
Available online 12 November 2019

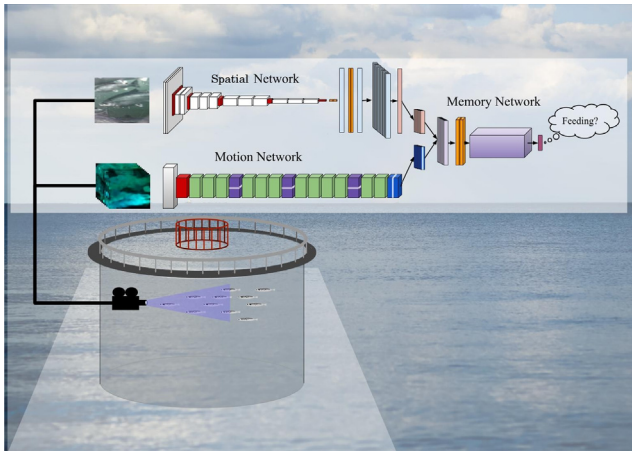


Fig. 1. An overview of the full prediction pipeline. The video is collected from within a breeding cage. Optical flow and spatial frames are generated before being fed into the Dual-Stream Recurrent Network, which performs prediction. An in-depth explanation is available in Fig. 3.

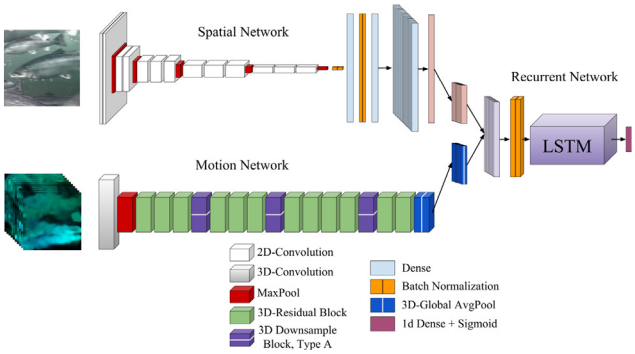


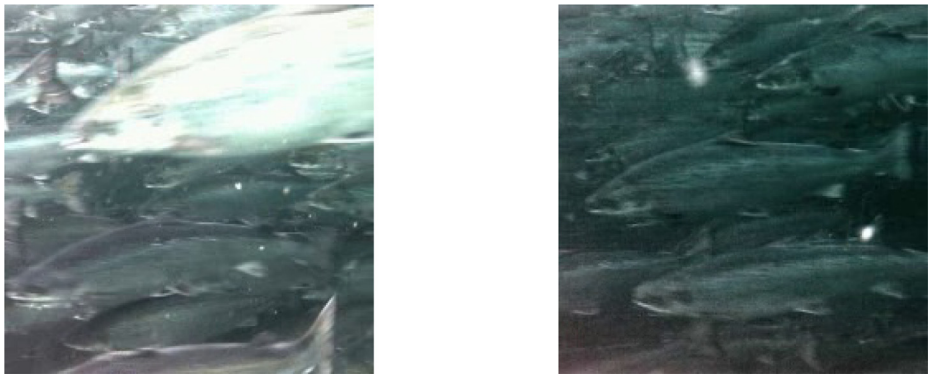
Fig. 3. The Dual-Stream Recurrent Network - DSRN. The spatial network takes single video frames as input and transforms them into feature vectors. 20 of these vectors are then averaged to create one high-level feature vector. The motion network takes a volume of 20 optical flow fields and transform them into a single high-level feature vector. 20 high-level feature vectors from each of the two networks are then concatenated to produce a sequence of 20 input vectors for the LSTM classification network. The LSTM classification network then produces a single classification for the entire sequence using a sigmoid activated unit.

surroundings from outdoor river rafting to indoor keyboard typing. This makes many of the actions immediately recognizable purely on the basis of their surroundings. Conversely, the fish domain is limited to underwater video from within a breeding cage at sea. Thus the surroundings in the fish videos are very similar, radically increasing the demand for robust action recognition capabilities. Underwater videos also introduce several other challenges. For example, when light hits the surface of the ocean it is reflected off the surface, while the light that does penetrate is refracted. This results in dimmer scenes with more uniform color distributions. Water also scatters and absorbs different wavelengths of light due to particles in the water, resulting in different shades of color based on the depth of the camera. These factors may be further enhanced by the surrounding weather. Direct sunlight produces very different lighting conditions than overcast weather. This is evident not only in the shades of color and amount of light in the scene but also in the amount of light reflected off the fish themselves. As more direct sunlight is present in the scene, more light is also specularly reflected off the fish, producing very bright areas in the frames as seen in Fig. 2. Another important aspect in the human action recognition domain is that human activities often can be classified purely based on discriminate poses of the subject. This has been utilized in Deep Learning approaches through the use of discriminative action poses to supplement videos during training (Ma et al., 2017). Fish, on the other hand, might not possess such defining poses. They are also often occluded by other fish, thus it might not be possible to discern whether fish are feeding or not, based on their pose alone. Action Recognition in fish is therefore much more dependent on the motion of the fish and how it changes through time. Additionally, fish often act together in a school, thus the behavior and pose of the school might be just as indicative of fish actions as individual fish behavior.

To account for these factors our approach relies heavily on motion from the salmons' movement temporal changes. Our approach consists of a 2D-convolutional spatial network and a 3D-convolutional motion network which act as spatial and motion feature extractors for a memory network which handles time-series data. Our full architecture is named a Dual-Stream Recurrent Network and can be seen in Fig. 3. We validate the DSRN using a data set consisting of underwater video of salmon, where the aim is to separate videos of Feeding salmon from videos of NonFeeding salmon. These videos are captured during feeding and non-feeding times at a real production fish farm, to create realistic Feeding and NonFeeding video data. We compare the DSRN to the performance of a regular CNN and a CNN combined with a LSTM and find that the DSRN accurately classifies 80% of the test videos, outperforming the other networks by a significant margin.

Contributions:

1. We propose a novel concept for the application of Action Recognition to salmon feeding behavior.
2. We collect a new data set of underwater video of salmon for Action Recognition from a real-world breeding facility.
3. We propose a general model for Action Recognition in situations



(a) A day with bright sunlight. (b) A day with little sunlight.

Fig. 2. Lighting conditions in the underwater video can vary based on many factors, such as weather, particles in the water and light diffraction. This produces higher demands on robust action recognition models with increased motion processing capabilities.

with a lack of distinct poses.
4. We propose a preprocessing strategy for underwater video containing specular reflections.
5. We introduce a new projection shortcut and show that it improves model performance over the architectures from (He et al., 2016a; Hara et al., 2017).

2. Related work

2.1. Human action recognition

Practice and theories for performing Human Action Recognition have been studied for many years (Poppe, 2010; Ali and Shah, 2010; Ji et al., 2013). This task involves processing videos and labeling them, using action labels. Early work explored several approaches for Human action recognition, including Unimodal Methods, Stochastic Methods and Rule-Based Methods (Vrigkas et al., 2015). However, recent advances in Deep Learning has led to the use of Deep Learning as a core methodology when developing action recognition approaches. Simonyan and Zisserman (2014a) train Convolutional Neural Networks (CNNs) as feature extractors before a final classification layer in their approach. The papers (Donahue et al., 2017; Yue-Hei Ng et al., 2015; Xu et al., 2016) improve upon this approach by adding Recurrent Neural Networks (RNNs) on top of CNNs for sequence handling. These Deep Learning approaches present the common approaches in human action recognition in recent years, with most of the approaches scoring well above 80% on the UCF-101 Action Recognition data set (Soomro et al., 2012). These methods have also been supplemented by the use of images with distinct poses, which also resulted in an increase in performance (Ma et al., 2017).

2.2. Animal action recognition

Automatic animal action recognition is a very recent field of study. With the advent of small wearable high-precision sensors such as gyroscopes and accelerometers in combination with video analysis tools such as deep learning, these methods promise to increase our understanding of animals and improve animal welfare through correct interventions. Peng et al. (2019) are able to classify cattle behavior by analyzing sensor data from IMU sensors using a Long Short-Term Memory (LSTM) RNN. They use sensor outputs from IMU's producing 3-axis accelerometer data, 3-axis gyroscope data and 3-axis magnetometer data. The 9-axis data is analyzed using a LSTM-RNN to produce a classification of the behavior. They find that they are accurately able to classify 8 types of behavior using an LSTM-RNN with a window size of 64 (3.2s). Yang et al. (2018) used a CNN to detect regions in images constituting pigs and their heads. They then develop an algorithm that is able to determine whether the pigs are feeding or not, based on the overlapping of the classified head region of each pig and the designated feeding zone. They are accurately able to classify pig feeding behavior from both still images and continuous video using this algorithm. Zhou et al. (2017) use near-infrared imaging to quantify variations in fish feeding behavior from a laboratory fish tank. They enhance images to a binary image and discard reflective frames using a Support Vector Machine (SVM) and a Gray-Level Gradient Co-Occurrence Matrix. They use fish centroids as a vertex in Delaunay Triangulation and use these results to calculate and quantify the flocking index fish feeding behavior (FIFFB). They find that they can accurately quantify and analyze variations in fish feeding behavior using the FIFFB.

2.3. Deep residual learning

The depth of neural network architectures is crucially important to their performance, however, deeper networks are harder to train due to degradation. In (He et al., 2016a), the concept of Residual Learning through Identity Mappings by Shortcuts is introduced. He et al. (2016a) create residual building blocks consisting of stacked convolutional layers. The output of the final convolutional layer in the block is then added to an identity mapping from the beginning of the block, creating an alternative route for the gradient to flow. They show that their residual networks outperform traditional CNNs with a significant margin on multiple classification tasks. They also show that they are able to build significantly deeper network architectures while still needing fewer trainable parameters than the traditional deep CNNs, thus reducing the need for computing power.

2.4. 3D-convolutional neural networks

Previous explorations of 3D-CNNs for action recognition already exist (Ji et al., 2013). However, this network is very shallow when compared to the recent 2D-CNNs (He et al., 2016a; Simonyan and Zisserman, 2014b). It is, therefore, reasonable to assume that the full potential of 3D-convolutions are not being utilized in these architectures. In (Hara et al., 2017), significantly deeper 3D-CNNs were explored for human action recognition in video. They trained their networks using 16 frame RGB video clips as input and showed an increase in performance compared to the architecture described by Ji et al. (2013). However, they do not explore the use of optical flow as was shown to increase performance in Varol et al. (2018).

2.5. Two-stream 3D-convolutional networks

2D-CNNs have shown significant performance gains in the field of image recognition and classification. They have also been used in video classification tasks, often supplemented by recurrent networks, with encouraging results. Xin et al. (2016) introduce an adaptive recurrent-convolutional hybrid network (ARCH) for Action Recognition tasks. The architecture consists of a Temporal-Spatial-Fusion CNN (TSF-CNN) which consists of a spatial CNN, a temporal CNN and a fusion network. The spatial CNN is structured similar to LeNet (LeCun et al., 1995) and takes the RGB video frames as input. The temporal CNN uses 3D-convolution to capture temporal data from optical flow input. Finally, the Fusion network fuses the two CNNs through a two-layer fully connected network. They implement a highly-distributed approach, where they train 10 parallel TSF-CNNs for 10 time-steps and input the scores of these networks into a sequence network, consisting of a RNN. When comparing the temporal ARCH to ARCH without the temporal CNN, they show that the temporal CNN significantly improves performance.

Our DSRN separates itself from the work in Xin et al. (2016) by using significantly deeper networks both in our 2D-CNN and 3D-CNN, giving our model a much higher capability of rich internal representations. We also only train one 2D-CNN and one 3D-CNN, showing that robust action recognition is possible on limited hardware. We also expand on the work in He et al. (2016b) by using a 3D-convolutional approach to their residual learning networks. Further, instead of using $1 \times 1 \times 1$ convolutions with a stride of 2 for the projection shortcuts in our downsampling layers, we introduce the use of $2 \times 2 \times 2$ pooling layers with a stride of 2. We show that this improves performance in our 3D-residual network while introducing no extra trainable parameters. We train deeper 3D-CNNs than those presented in Ji et al. (2013) and explore the effects of using optical flow instead of the RGB approach presented in Hara et al. (2017). Finally, we validate our approach on a data set consisting of underwater video of salmon. This data set is collected from a real facility during production, separating it from Zhou et al. (2017). The subjects in these videos might not possess the distinct poses explored in Ma et al. (2017) and we would therefore expect to rely more heavily on the model's ability to process motion and temporal features.

3. Dual-stream recurrent network

The Dual-Stream Recurrent Network architecture introduces a

network that combines the deep hierarchical visual feature extractor properties of 2D-CNNs and the temporal translation feature extractions of 3D-CNNs. It combines this with a network that can learn to understand the temporal dynamics of sequential data to produce a powerful and robust model for action recognition.

3.1. Spatial network

The spatial network consists of a 2D-CNN, which takes a single video frame as visual input and transforms it into a fixed length feature vector, ϕ_t through the use of hierarchically stacked convolutional- and densely connected layers. We use the popular VGG-16 architecture (Simonyan and Zisserman, 2014b) as a starting point and modify it, using batch normalization layers between every dense layer. To account for the challenging conditions from underwater video, we also introduce a new preprocessing strategy. This strategy involves grayscaling of images, equalizing the histograms, normalizing and zero-centering our data and removing specular reflections and dark areas from the frames by zeroing out all pixels above or below a threshold value. By first zero-centering and normalizing the data before performing the thresholding step, we force most of the image data to lie well within the fixed threshold value. Thus, only very bright or very dark areas are zeroed out by the thresholding step. The results from our preprocessing technique can be seen in Fig. 4.

3.2. Motion network

Previous experiments were done with two-stream architectures, using a 2D-convolutional motion network taking single optical flow fields as input. This gave little notable performance gains when compared to a single spatial network. We hypothesize this was due to the fact that the 2D-convolutions were only capable of analyzing a single optical flow field at the time. This limits the amount of motion information it could process to what is contained within that single optical flow field. Further experiments using optical flow fields generated from video frames with several frames apart yielded similar results. 3D-convolutions were therefore explored to improve the amount of motion information we could process in the DSRN motion feature extractor. For the 3D-convolutional motion network, we expanded on the architectures proposed by He et al. in He et al. (2016b), by increasing the dimensionality of the convolutional layers from 2D-convolutions to 3D-convolutions. This produces 3D-residual blocks. A 3D-residual block consists of an identity skip connection and two $3 \times 3 \times 3$ 3D-convolutional layers as seen in Fig. 6. For this purpose, we use the pre-activated modification from He et al. (2016a). This enables us to train very deep 3D-CNNs, without suffering from the drastic increase in trainable parameters usually encountered with this dimensional increase. The network takes a volume of optical flow fields, o_t, of size s as input $<o_0, o_1, ..., o_s>$ and transforms it to a single fixed-length feature vector θ_T. In our motion network we use 34 layers and exchange the 1×1 convolution downsampling layers for max pooling layers. We also use a volume size of $s = 20$ for our input to give the network enough temporal motion information to process. An example video frame and its resulting optical flow field is shown in Fig. 5. The motion network is shown in Fig. 7. In Section 5.4 we present an ablation study of our motion network done on the validation data set to evaluate the effect of our modifications compared to a straight forward dimension increased Residual Network.

Optical Flow. Optical Flow is a pattern illustrating the motion of image objects between two or more consecutive frames. These motions are usually caused by the movement of the objects in the frame, but can also be caused by the movement of the camera. Optical Flow requires that the pixel intensities of an object stay the same in frames following each other. It also assumes that pixels in very close proximity to each other have a similar motion trajectory.

A pixel $P(x_0, y_0, t_0)$ in frame 0 has moved a distance (dx, dy) in frame 1, taken after dt time, resulting in Eq. (1)

$$P(x_1, y_1, t_1) = P(x_0 + dx, y_0 + dy, t_0 + dt) \tag{1}$$

A Taylor series approximation and a removal of the common terms, followed by a division by dt gives us Eq. (2)

$$f_x u + f_y v + f_t = 0 \tag{2}$$

where f_x, f_y, u and v are given by Eq. (3). We see that f_x and f_y are the image gradients and that f_t is the gradient along the time dimension.

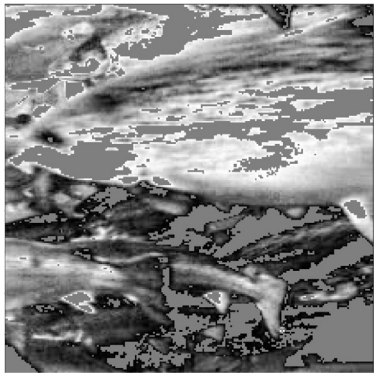
$$\begin{aligned} f_x &= \frac{\delta f}{\delta x}; \quad f_y = \frac{\delta f}{\delta y} \\ u &= \frac{dx}{dt}; \quad v = \frac{dy}{dt} \end{aligned} \tag{3}$$

There are several methods to calculate u and v, but a common one, also used in this study, is Gunnar Farnebäck's algorithm (Farnebäck, 2003). This algorithm produces a 2-channel array of Optical Flow vectors (u, v) with magnitude and direction as shown in Fig. 5.

3.3. Memory network

The final classification is performed by a memory network which consists of a 256-cell LSTM recurrent network. The inputs sequences were generated by concatenating the feature vectors from layer fc7 in the spatial network and the Global AvgPool layer in the motion network. To account for the fact that the motion network uses 20 flow fields to produce a single vector θ_T, we average 20 feature vectors from

(a) The original image.



(b) The image after applying our preprocessing technique.

Fig. 4. The preprocessing technique grayscales the image, equalizes the histogram, zero-centers and normalizes the image and removes specular reflections and dark areas by setting pixels above or below a certain threshold to 0.

(a) The original image. (b) The resulting opical flow field.

Fig. 5. The optical flow field resulting from the video. Green indicates leftward motion, Red indicates rightward motion and Blue indicates downward motion. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

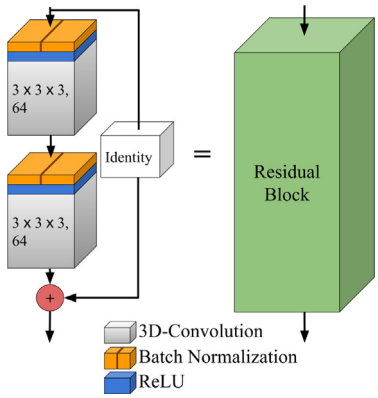


Fig. 6. A unit in the 3D-Residual Neural Networks. This is the 3D-Convolutional equivalent to the original 2D-residual unit, presented in He et al. (2016a) with the pre-activation modifications from He et al. (2016b).

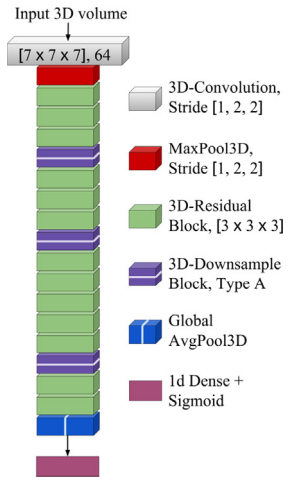


Fig. 7. 3D-Residual Network. It takes an input 3D volume, consisting of 20 consecutive stacked Optical Flow fields. The volume is downsampled using $2 \times 2 \times 2$ MaxPoolings and maintains temporal dimension deeper in the network through a temporal stride of 1 in the first convolutional and pooling layers.

the 2D-CNN to produce a fixed-size average vector $\phi_T = \frac{\phi_0 + \phi_1 + \cdots + \phi_{19}}{20}$ for the 2D-CNN. The input to the recurrent network is batch-normalized and transformed to a single classification output, using a single sigmoid activated, densely connected unit. A classification in the range [0.0, 0.5) is read as NonFeeding and a classification in the range [0.5, 1.0] is read as Feeding. An overview of the Dual-Stream Recurrent network is seen in Fig. 3.

3.4. DSRN implementation details

Since action recognition tasks are more complex than standard image recognition tasks, training networks for these tasks is also more difficult. Our data set is significantly smaller than the ImageNet data set (Deng et al., 2009), but only contains two possible classes, thus ensuring that the number of examples for each class is sufficient to expect training to converge. We use TensorFlow™ (Abadi et al., 2015) with the TFLearn library (Damien et al., 2016a) for our network implementations. We used a hardware set up consisting of two Intel Xeon 10 core processors and a single Nvidia GTX1080 GPU for training. The DSRN was trained using a network-by-network approach, meaning that each of the networks used in the complete model was trained individually, using its own classification layer. This was done because each model required the full hardware we had available during training.

For the spatial network we pre-trained the model using a publicly available VGG16 model (Damien et al., 2016b; Simonyan and Zisserman, 2014b) and then fine-tuned the parameters to our data set. We used stochastic gradient descent with a mini-batch size of 20 and a learning rate of 10^{-4} . The network was trained for 16 K iterations.

The DSRN motion network consists of a truly deep 3D-Residual Network, described in detail in Section 3.2. This network was trained using a 3D volume of stacked optical flow fields using the Farnebäck optical flow algorithm (Farnebäck, 2003) using the OpenCV implementation. We trained the network from scratch on our data set and used a similar training procedure as was used for the spatial network. We used a mini-batch size of 10 and a learning rate of 10^{-3} and the training was stopped after 13 K iterations.

The recurrent network was trained using sequences of concatenated feature vectors generated from both the spatial and the motion network. We used 20 feature vectors from the motion network per sequence, resulting in a total of 400 video frames per sequence. The network was trained using a mini-batch size of 20 sequence vectors and a learning rate 10^{-3}. Training was stopped after 2 K iterations.

4. Experimental setup

4.1. Evaluation protocol

We compare the DSRN to a spatial baseline model consisting of only the spatial network from the DSRN and a spatial recurrent network (SRN) consisting of the same spatial network as our spatial baseline, but with a 256-cell LSTM sequence classifier on top These networks were chosen to evaluate the value of motion information in our DSRN. We

Table 1
The table shows the different Action Snippet lengths for the three evaluated networks.

Network	Action Snippet length
Spatial Baseline	1
SRN	20
DSRN	400

introduce four performance measures with which we can compare the three architectures (PM1-PM4). The main measure of performance for all networks is PM2 - Video Action Recognition Accuracy. For the evaluation, we also report the Action Snippets performances for higher granularity in the results. Here, Action Snippets refers to the number of frames needed for a single classification to be made by the given architecture. Thus, the Action Snippet lengths vary between architectures and an overview of Action Snippet lengths are given for each architecture in Table 1. We use a total of four performance measures in our testing:

PM1: Individual Video Action Recognition accuracy. This measure gives a single binary classification for the entire duration of the video, where 1 means correctly classified and 0 means incorrectly classified. The predicted class for the video is calculated using the majority of the network predictions for that video. If the number of predictions is the same for both classes, we say the video is incorrectly classified.

PM2: Average Video Action Recognition accuracy for the entire test set. This measure gives the total percentage of correctly classified videos using Video Action Recondition accuracy. This is the main performance measure.

PM3: Average Action Snippet accuracy within individual videos. A video might consist of several Action Snippets, depending on the architecture, as seen in Table 1. This measure gives the average accuracy for all the Action Snippets within one video.

PM4: Average Action Snippet Action Recognition accuracy over the entire test set. This measure gives the average accuracy over the entire test data set using Action Snippets.

Thus a PM1 score of 1 with a PM3 score of 51% suggests that the model has misclassified a significant portion of the video, whereas a PM1 score of 1 and a PM3 score of 100% suggests that the model correctly classified the entire video. In a similar manner, a PM2 score 50% with a PM4 score of 53% indicates that the model is only able to correctly classify 50% of all videos, but 53% of all action snippets, showing that some misclassified videos contain correctly classified action snippets.

4.2. Data set

The data set consists of videos collected in the northern part of Norway, in November of 2016. They were collected from a salmon farming site using a standardized procedure to produce a consistent and representative data set. The camera was mounted looking inward towards the center of the cage during both Feeding and NonFeeding videos and captured at intervals of 2000, 5000 and 20,000 frames. The total data set consists of 76 videos, taken at a resolution of 224×224 pixels with RGB color channels and at 24 frames per second. The videos were labeled according to the feeding times, provided by the feeding operator at the farming facility. Each video contains either Feeding or NonFeeding action and there is no overlap between the actions within videos. We note that lighting conditions and turbidity are important factors impacting the quality of the videos, however as this is not within the scope of this study, we will not explore this further.

Data Set split. When splitting the original data set into training, validation and test sets, we split the videos based on dates. This ensures that all videos from a particular date are only present in one of the three subsets. The reasoning behind this split was twofold. First, the conditions for a particular day might enable the model to overfit on other factors than the behavior of the fish, such as how the camera moves or the light conditions for that day. Splitting on dates avoids this. Second, splitting the data set based on dates gives the best representation of the performance that can be expected if the model is deployed on a breeding site and starts receiving new video data. The training and validation data sets consist of a 50%/ 50% split of Feeding and NonFeeding videos. The data set split is shown in Table 2.

Table 2
The table shows the three data set splits with their respective amounts of frames and the percentage of the total amount of data.

Subset	# frames	data set fraction
Training	326768	64.51%
Validation	85760	16.93%
Test	94000	18.56%
Total	506528	100.00%

Test Data Set. The test data set consists of 20 videos ranging from 2000 to 20000 frames per video. Videos 0–10 are Feeding videos while videos 11–19 are NonFeeding videos. The distribution of videos and frames is shown in Table 3.

5. Results and discussion

5.1. Spatial baseline

We first measure the performance of the spatial baseline model to establish a baseline. From the results, given in Table 4, it is clear that the major part of the network performance comes from its ability to correctly classify NonFeeding videos (videos 11–19). Only one of the NonFeeding videos is wrongly classified, while six of the Feeding videos are wrongly classified. This indicates that Feeding behavior is harder to recognize than NonFeeding behaviour. This is also apparent in PM3, where we see that the accuracies are much higher for NonFeeding videos than they are for Feeding videos. In Fig. 8 we see the same tendency as the loss is much higher and variable for the Feeding section of the loss curve than it is for the NonFeeding section.

5.2. Spatial Recurrent Network (SRN)

Having evaluated the Spatial network and established a baseline, we now turn to the SRN. This network treats videos as sequences of image frames and is, therefore, more capable of learning from temporal transitions than the purely spatial baseline model. It uses 20 frames per prediction, resulting in a prediction for every 0.83 s of video. From the results in Table 4, we see that the inclusion of a recurrent network notably increases performance compared to the baseline. We also note that this increase in performance is exclusively due to the improved ability to correctly classify Feeding videos. In fact, the performance on NonFeeding videos is actually decreased from an average of 90.1%, in the baseline, to 89.1% in the SRN, using PM3. This indicates that accurate classification of Feeding behavior is highly dependent on temporal features in a way that NonFeeding behavior is not. Form Fig. 8 it is clear that more temporal information increases classification confidence for the SRN compared to the Spatial Baseline, as is indicated by

Table 3
The distribution of the number of videos and number of frames in the testing data set.

Label	# of videos	# of frames	Fraction
Feeding	11	43K	45.7%
NonFeeding	9	51K	54.3%
Total	20	94K	100.0%

Table 4
A comparison of the test results between the spatial baseline, the Spatial Recurrent Network (SRN) and the Dual-Stream Recurrent Network (DSRN) using the four performance measures. It is apparent that the DSRN outperforms both the spatial baseline and the SRN with a significant margin. This also becomes apparent from PM3, where the DSRN significantly improves upon the performance of the other networks on the Feeding videos (0–10).

	Baseline:				SRN:				**DSRN:**			
Video #	PM1	PM2	PM3	PM4	PM1	PM2	PM3	PM4	PM1	PM2	PM3	PM4
0	0		49.0%		1		97.0%		1		**100.0%**	
1	1		65.1%		1		74.0%		1		**100.0%**	
2	0		2.3%		0		0.6%		1		**100.0%**	
3	0		0.0%		0		0.0%		0		0.0%	
4	1		71.9%		1		76.4%		1		**100.0%**	
5	1		90.0%		1		100.0%		1		**100.0%**	
6	1		95.1%		1		100.0%		1		**100.0%**	
7	1		71.8%		1		99.6%		1		**100.0%**	
8	0		11.6%		0		16.4%		0		**16.7%**	
9	0	65.0%	32.6%	65.5%	1	75.0%	91.2%	75.3%	1	**80.0%**	**100.0%**	**82.5%**
10	0		9.1%		0		44.0%		1		**91.7%**	
11	1		99.9%		1		**100.0%**		0		50.0%	
12	1		100.0%		1		100.0%		1		100.0%	
13	1		100.0%		1		100.0%		1		100.0%	
14	0		**12.6%**		0		3.2%		0		0.0%	
15	1		99.6%		1		100.0%		1		**100.0%**	
16	1		100.0%		1		100.0%		1		100.0%	
17	1		99.3%		1		100.0%		1		**100.0%**	
18	1		**99.6%**		1		98.8%		1		91.7%	
19	1		100.0%		1		100.0%		1		100.0%	

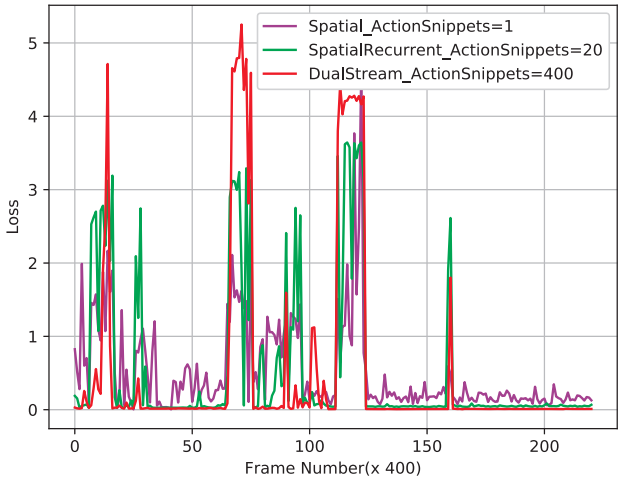


Fig. 8. The loss curves for the three architectures on the testing data set. The DSRN is very consistent through all action snippets, indicated by the flat curve. The low loss indicates that the DSRN is very "confident" in its prediction. The high spikes indicate that this is also the case even when the DSRN wrongly classifies action snippets.

the lower loss during correct classification. This increased "self-confidence" is also apparent when the model misclassifies action snippets as is seen by the large spikes in the loss during misclassification. The model remains confident in its classification, but since it is wrong the loss is very large.

5.3. DSRN

Finally, we evaluate the complete Dual-Stream Recurrent Network. This model combines a spatial network and a motion, 3D-convolutional, network as feature extractors for a 256-cell LSTM classification network. Through the use of both a spatial network and a motion network, this model is capable of interpreting temporal changes in both spatial- and motion-features. It uses a total of 400 video frames per prediction, resulting in a prediction for every 16.7 s of video. From the results in Table 4 we see a significant increase in performance compared to the baseline network. We again note that this performance gain comes exclusively from the ability to classify Feeding videos. Compared to the baseline, the DSRN classifies 4 more Feeding videos correctly, resulting in a final accuracy of 80.0%. Compared to the SRN, we see the same tendencies. The inclusion of motion features through 3D-convolutions and optical flow fields results in an increase in performance, largely due to increased capabilities to classify Feeding videos. In Fig. 8 we also see a further increase in model confidence, with lower loss curves for correct classifications and higher loss for wrong classifications. This not only further strengthens the evidence for the usefulness of temporal and motion information in video Action Recognition, but also indicates that the spatial and motion networks are complementary to each other. We also note that the DSRN compiles significantly more temporal information, per prediction, than the other models. This indicates the usefulness of temporal information for salmon video classification. The DSRN outperforms all other models on our data set with a significant margin, providing evidence for the benefits of deep 3D-CNNs and increased temporal information processing.

5.4. Motion network ablation study

Using the validation data set, we performed an ablation study of our motion network. This was done to evaluate the impact of our different modifications from a straight dimension increase to residual networks, as was done in Hara et al. (2017).

Network Depths. The depth of CNNs have been of significant importance for their performance in image recognition tasks. In (Simonyan and Zisserman, 2014b), Simonyan et al. showed that deeper networks resulted in better performance. We, therefore, explore two different network depths. The validation results for these are shown in Table 5. We vary depths from 18- and 34-layers. The results show that

Table 5
The average of validation accuracies for the different network depths.

Depth	Accuracy
18-layer	77.2%
34-layer	**80.7%**

Table 6
The average of validation accuracies for the two input volume sizes. The dimensions correspond to: *num_frames* × *frame_height* × *frame_width*.

Input dimensions	Accuracy
10 × 224 × 224	80.7%
20 × 224 × 224	**81.4%**

the 34-layer architecture produces the best results.

3D Input Volume Sizes. Experiments with 3D-convolutions have shown that increasing the temporal information processed by a model, through using more video frames, can lead to an increase in performance (Ji et al., 2013). We, therefore, compare 3D-Residual networks trained using 3D volumes of different sizes. The results are seen in Table 6 and show that increasing the number of optical flow fields used in the volume, from 10 to 20 leads to a performance increase.

Projection Shortcuts. In (He et al., 2016a), He et al. found that projection shortcuts, using 1 × 1 convolutions with a stride of 2 in downsampling blocks, improved performance when compared to regular downsampled identity shortcuts. Since pooling layers with a kernel size of 2 × 2 consider a larger region of the input than 1 × 1 convolutions, we explore two pooling options to the 1 × 1 convolutions. Pooling layers do not introduce any new trainable parameters in a network, and will, therefore, result in a decrease in the number of network parameters, compared to 1 × 1 convolutions. We compare the two different pooling approaches to the original convolution approach in our motion network:

Type A: 2 × 2 × 2 MaxPool-layer with stride 2 × 2 × 2.
Type B: 2 × 2 × 2 AvgPool-layer with stride 2 × 2 × 2.
Type C: 1 × 1 × 1 Conv-layer with stride 2 × 2 × 2.

The results are presented in Table 7 and show that the use of pooling layers significantly increase the model's performance compared to convolutional projection shortcuts.

5.5. Discussion

Since both the SRN and DSRN use several frames to produce a prediction, as seen in Table 1, the number of predictions made for a given video ranges from 100–1000 for the SRN and 5–50 for the DSRN. This results in almost categorical results in many videos as the number of predictions per video decreases, but as we showed in Fig. 8 the DSRN is very confident in its predictions. This is apparent by the low loss curves for correct classifications and high loss curves for the wrong classifications. This indicates that the model is indeed very certain when it gets 100.0% on a given video and is not only "hedging its bets" by producing predictions near the 0.5 mark.

In Table 4 we showed that more temporal information is closely correlated to an increase in classification performance. In Section 5.3 we also show that this increase in performance is due to an increased ability to classify feeding videos. We, therefore, hypothesize that feeding videos are harder to classify because of the much higher variance in swimming patterns during feeding. During Feeding the salmon will break out of the school in order to follow sinking pellets, resulting in different swimming patterns than during Non-Feeding. This might seem "confusing" to the models at first glance, but given more temporal data, the SRN and DSRN can make more and more sense of the swimming patterns, resulting in a Feeding classification.

Table 7
The average of validation accuracies for the three downsampling strategies.

DS strategy	Accuracy
20 × 224 × 224 Type A	**83.3%**
20 × 224 × 224 Type B	83.2%
20 × 224 × 224 Type C	81.4%
10 × 224 × 224 Type C	80.7%

In Section 4.2 we mention that our training, validation and test data are split on dates to give a representative performance of our model, given new data. However, this split also increases the challenge for our models. Traditionally machine learning data sets are split in a way to make the training data representative of the validation and test data. Otherwise, the models might learn features that are not present in one of the splits, thus negatively affecting model performance. By splitting our data on dates instead of within dates, we completely remove all information from those dates from our training data. Since salmon behavior could be affected by conditions only present within these dates, we risk degrading our models' performance due to a lack of exposure to such conditions. However, as seen in Table 4, our prediction results are satisfactory, despite the challenging data set split. We therefore conclude that the robustness to noise in our DSRN is very good, which indicates that the DSRN is well suited to general action recognition tasks.

Despite our positive results, there are some potential weaknesses to our approach. First, our DSRN was trained in a step-wise manner, meaning that each sub-network in the model was trained isolated from the other sub-networks. Since this training regime only allows each sub-network to learn important features during its own training, there is a very high likelihood that DSRN performance could be increased by training the full model in a final end-to-end fine-tuning. This would allow the gradient to flow through all the sub-networks, which could make the features produced by the spatial and motion networks even more usable for the memory network. Second, our data set is relatively small and only contains videos from a single cage at one fish farming facility. This means that the model might be overfitted to our specific cage and the fish and conditions within it, since it has not seen any other examples. This could mean that our model will perform much worse in a new cage, even at the same facility, due to different conditions and fish. However, this is mitigated by our data set split and the preprocessing strategy used for the spatial network. Since we split on dates and not within dates, our results are reported on never-before-seen conditions, thus we see that the DSRN generalizes well to changing conditions which suggests that the model itself is robust to such changes. A larger data set, from different cages and facilities would still be an important contribution to further prove the robustness of our approach. Third, the preprocessing strategy used for the spatial network was chosen to reduce the impact of differing lighting conditions and noise. We note that since the images are both zero-centered and normalized, most of the pixel data will lie well within the used threshold. However, a dynamical thresholding approach could further improve the preprocessing strategy and could be an interesting future direction for this work.

6. Conclusion

In this paper, we proposed the DSRN architecture for spatial temporal salmon action recognition for applications with a lack of distinct poses and dynamic spatio-temporal changes during motion. This architecture is validated on a data set of underwater salmon videos and resulted in a prediction accuracy on the test data set of (80.0%) which beats the baseline architectures by a large margin, as seen in Table 4. The underwater video salmon data set is in many ways more challenging than the data sets used for human action recognition due to challenging light conditions, turbidity, the lack of discriminative poses, dynamic and undiscriminating motion patterns. Through the use of very deep 2D- and 3D-CNNs, together with a LSTM, our model combines spatio-temporal and motion features to perform action recognition with an accuracy that is relevant for industrial use. The

performance of the DSRN indicates that the inclusion of motion features through the use of optical flow and a 3D-convolutional motion network provides a real benefit and that temporal processing of both spatial- and motion-features results in increased performance in video classification. Our findings demonstrate a robust model for feeding action recognition in salmon aquaculture which can be used for better management and optimization of feeding operations in aquaculture, since feeding is the most costly salmon farming operation. The architecture is also applicable to similar domains, such as livestock or animal action recognition, where the lack of distinct poses requires a higher reliance on motion for prediction and recognition.

For future work we aim to demonstrate the approach in large scale across several net cages and localities in Norway to capture the seasonal variations in both salmon behaviour and environmental conditions. With the recent advances in CNNs, it would be interesting to consider newer CNNs as both spatial and motion feature extractors. Transformers have been shown to outperform recurrent neural networks on many sequence tasks (Vaswani et al., 2017). It would, therefore, also be interesting to explore a model using a transformer instead of a recurrent network as the memory network. Finally it would be interesting to explore end-to-end trainable implementations of the DSRN to jointly optimize all components of the DSRN as this could further improve performance through improved gradient calculations for the entire architecture.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the Intelligent Project. The authors gratefully acknowledge the SINTEF Ocean RACE program and its chairman Dr. Arne Fredheim for financing the research in this work. In addition, the financial support for Dr. Ekrem Misimi from SalmonInsight project (RCN 280864) is greatly acknowledged. Preparation for this paper was also supported by the Centre for Research-based innovation in Aquaculture Technology, Exposed Aquaculture Operations (RCN 237790).

References

Ali, S., Shah, M., 2010. Human action recognition in videos using kinematic features and multiple instance learning. IEEE Human action recognition in videos using kinematic features and multiple instance learning 32 (2), 288–303. https://doi.org/10.1109/TPAMI.2008.284.

Damien, A. et al., 2016. Tflearn. https://github.com/tflearn/tflearn.

Damien, A. et al., 2016. Tflearn vgg-16 pretrained model. URL https://github.com/tflearn/models.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. Conference on Computer Vision and Pattern Recognition. IEEE, pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T., 2017. Long-term recurrent convolutional networks for visual recognition and description. In: Transactions on Pattern Analysis and Machine Intelligence., 4th. 39. IEEE, pp. 677–691. https://doi.org/10.1109/TPAMI.2016.2599174.

Farnebäck, G., 2003. Two-frame motion estimation based on polynomial expansion. Image Analysis 2749 (SCIA 2003. Lecture Notes in Computer Science) 363–370. https://doi.org/10.1007/3-540-45103-X_50.

Fiskeridirektoratet, 2016. Profitability survey on the production of Atlantic salmon and rainbow trout. pp. 1-77. Fiskeridirektoratet. https://www.fiskeridir.no/content/download/17237/244931/version/6/file/rap-lonnsomhet-akvakultur-2015.pdf.

Hara, K., Kataoka, H., Satoh, Y., 2017. Learning spatio-temporal features with 3d residual networks for action recognition, 2017, pp. 3154–3160. http://doi.org/10.1109/ICCVW.2017.373.

He, K., Zhang, X., Ren, S., Sun, J., 2016a. Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition. IEEE, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

Ji, S., Xu, W., Yang, M., Yu, K., 2013. 3D convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1), 221–231. https://doi.org/10.1109/TPAMI.2012.59.

Kamilaris, A., Prenafeta-Boldú, F.X., 2018. Deep learning in agriculture: a survey. Computers and Electronics in Agriculture 147 (nil), 70–90. https://doi.org/10.1016/j.compag.2018.02.016.

LeCun, Y., Jackel, L., Bottou, L., Cortes, C., Denker, J.S., Drucker, H., Guyon, I., Muller, U., Sackinger, E., Simard, P., et al., 1995. Learning algorithms for classification: a comparison on handwritten digit recognition. Neural Networks: The Statistical Mechanics Perspective 261–276.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436. https://doi.org/10.1038/nature14539.

Varol, G., Laptev, I., Schmid, C., 2018. Long-term temporal convolutions for action recognition. In: Transactions on Pattern Analysis and Machine Intelligence. IEEE. 40 (6) (2018) 1510–1517. http://doi.org/10.1109/TPAMI.2017.2712608.

He, K., Zhang, X., Ren, S., Sun, J., 2016b. Identity mappings in deep residual networks. In: Computer Vision – ECCV 2016. Springer International Publishing, pp. 630–645. https://doi.org/10.1007/978-3-319-46493-0_38.

Misimi, E., Øye, E.R., Sture, Ø., Mathiassen, J.R., 2017. Robust classification approach for segmentation of blood defects in cod fillets based on deep convolutional neural networks and support vector machines and calculation of gripper vectors for robotic processing. Computers and Electronics in Agriculture 139, 138–152. https://doi.org/10.1016/j.compag.2017.05.021.

Peng, Y., Kondo, N., Fujiura, T., Suzuki, T., Yoshioka, H., Itoyama, E., et al., 2019. Classification of multiple cattle behavior patterns using a recurrent neural network with long short-term memory and inertial measurement units. Computers and Electronics in Agriculture 157, 247–253. https://doi.org/10.1016/j.compag.2018.12.023.

Poppe, R., 2010. A survey on vision-based human action recognition. Image and Vision Computing 28 (6), 976–990. https://doi.org/10.1016/j.imavis.2009.11.014.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems, software available from tensorflow.org. https://www.tensorflow.org/.

Ma, S., Bargal, S. A., Zhang, J., Sigal, L., Sclaroff, S., 2017. Do less and achieve more: Training cnns for action recognition utilizing action images from the web, Pattern Recognition 68 (2017) 334 – 345. http://doi.org/10.1016/j.patcog.2017.01.027.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition, arXiv: 1409.1556.

Soomro, K., Zamir, A.R., Shah, M., 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv: 1212.0402.

Vaswani, A, Shazeer, N. , Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., N., Kaiser L., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008.

Simonyan, K., Zisserman, A., 2014a. Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, 27th. Neural Information Processing Systems Conference, pp. 568–576.

Vrigkas, M., Nikou, C., Kakadiaris, I.A., 2015. A review of human activity recognition methods. Frontiesr in Robotics and AI 2, 28. https://doi.org/10.3389/frobt.2015.00028.

Xin, M., Zhang, H., Wang, H., Sun, M., Yuan, D., 2016. Arch: Adaptive recurrent-convolutional hybrid networks for long-term action recognition. Neurocomputing 178, 87–102. https://doi.org/10.1016/j.neucom.2015.09.112.

Xu, Z., Hu, J., Deng, W., 2016. Recurrent convolutional neural network for video classification. In: International Conference on Multimedia and Expo (ICME). IEEE. pp. 1–6. http://doi.org/10.1109/ICME.2016.7552971.

Yang, Q., Xiao, D., Lin, S., 2018. Feeding behavior recognition for group-housed pigs with the faster r-cnn. Computers and Electronics in Agriculture 155, 453–460. https://doi.org/10.1016/j.compag.2018.11.002.

Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G., 2015. Beyond short snippets: deep networks for video classification. In: Conference on Computer Vision and Pattern Recognition. IEEE, pp. 4694–4702. https://doi.org/10.1109/CVPR.2015.7299101.

Zhou, C., Zhang, B., Lin, K., Xu, D., Chen, C., Yang, X., Sun, C., 2017. Near-infrared imaging to quantify the feeding behavior of fish in aquaculture. Computers and Electronics in Agriculture 135, 233–241. https://doi.org/10.1016/j.compag.2017.02.013.