EasyChair Preprint

№ 10957

Video-Based Recognition of Aquatic Invasive Species Larvae Using Attention-LSTM Transformer

Shaif Chowdhury, Sadia Nasrin Tisha, Monica McGarrity and Greg Hamerly

September 23, 2023

Video-Based Recognition of Aquatic Invasive Species Larvae Using Attention-LSTM Transformer

Shaif Chowdhury[1], Sadia Nasrin Tisha[1], Monica E. McGarrity[2], and Greg Hamerly[1]

[1] Baylor University, Waco TX 76706, USA https://www.baylor.edu
[2] Texas Parks and Wildlife Department https://tpwd.texas.gov/

Abstract. Aquatic species like zebra and quagga mussels are invasive in United States waterways and cause ecological and economic damage. Due to the time-consuming nature of conventional early detection methods, there is a need for automated systems to detect and classify invasive and non-invasive species using a video-based system without any human supervision. We present a video classification model for rapidly recognizing invasive and non-invasive mussel larvae from plankton or water sample videos.

Many recent video recognition models are transformer-based and use a combination of spatial and temporal attention, often with large-scale pre-training. We present a model with a CNN-based patch encoder and transformer blocks consisting of temporal attention with LSTM that is end-to-end trainable and effective without pre-training. Based on detailed experiments, the Attention-LSTM model significantly improves over state-of-the-art video classification models, classifying invasive and non-invasive larvae with 99% balanced accuracy. Our code is available at https://anonymous.4open.science/r/AttLSTM-10CF/

Keywords: Recognition · Video Recognition · Attention-LSTM · Transformer · Aquatic Invasive Species · Dreissenid · Quagga Mussel · Zebra Mussel

1 Introduction

Zebra and Quagga mussels are native to Eurasia but have become widely introduced and invasive into North American waters causing ecological disruption[24]. These organisms fight for resources causing the extinction of other freshwater mussels[26]. Dreissenid mussels spread rapidly, forming large colonies and restricting water flow and impeding power generation from water systems, clogging pipes, and other machinery[8]. In the United States, dreissenid cause several hundred million in damages to power plants, water systems, and industrial water intakes annually. dreissenid mussels are relatively easy to detect, but they spread quickly laying millions of eggs a year. Once adult zebra mussels have established a presence in a water body, with reproducing adults present,eradicating

or controlling their growth is populations is not possible and impacts on water infrastructure are imminent. That means it is imperative to monitor the presence of such invasive species at the larval stage[17]. The conventional methods of detecting veliger presence are to collect plankton or water samples and then examine the selection using cross-polarized light microscopy[17] or environmental DNA[27]. Both of these methods are costly, time-consuming, and require human experts. For this reason, it is vital to develop an automated procedure to visually monitor the veliger of invasive species from water sample videos.

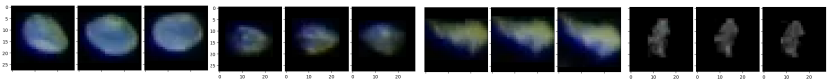

Fig. 1. Example of invasive dreissenid and non-invasive species larvae in our dataset. There are four different organisms in this image, with the first six columns containing images of two different invasive larvae (three for each organism) and the following six containing images of two different non-invasive larvae.

This research aims to classify invasive dreissenid and non-invasive larvae from videos of water samples. We track objects in the video across frames and then extract a cropped image for each tracked object from each frame in which it appears. Every object has a sequence of images that must be classified as invasive or non-invasive. A set of invasive and non-invasive images are shown in Figure 6. Previously invasive species recognition[6] has been done using a VGG-based CNN and an autoencoder-based feature fusion strategy. Invasive organisms often have very distinguishable movement compared to non-invasive larvae[25], which makes it crucial to model both spatial features and temporal relations between different frames. This paper introduces an Attention-LSTM-based model for end-to-end video-based classification of invasive and non-invasive organisms.

1.1 Attention-LSTM

Recognition is a fundamental challenge in computer vision, both in image and video recognition. Based on the success of transformers in natural language processing[32], many transformer-based architectures have been proposed for image and video recognition[10,21,1]. The Video Vision Transformer has shown to be effective at classifying videos in multiple video recognition datasets[1] using a space-time attention transformer.

But transformer-based models are generally more effective when large datasets are available for pre-training[32,10]. On the other hand, LSTMs have been known to be very efficient in modeling sequential information[15,9]. CNN-based architectures have also proven to be good at extracting spacial features[11,9]. Another recent development is the Sequencer[30] architecture by Tatsunami et al. that used Bi-directional Long-Short-Term Memory (LSTM) for image classification. The authors have conducted detailed experiments showing that the Sequencer

outperforms Vision Transformers with comparable parameters on ImageNet1k. This indicates that the recurrence mechanism of LSTM can model long-range sequences in the same way self-attention layers can.

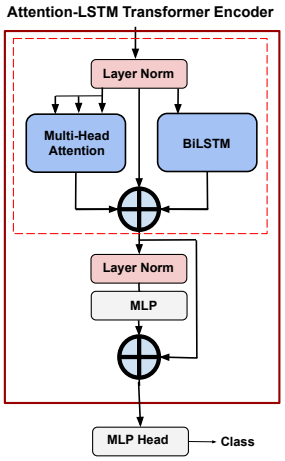

Fig. 2. The Attention-LSTM transformer.

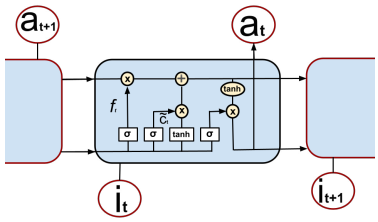

Fig. 3. An LSTM unit.

Our architecture is based on a novel Attention-LSTM transformer block combining bi-directional LSTM and multi-head attention layers. The introduction of the LSTM layer and self-attention are efficient for modeling fine-grained features in the video. We propose a hybrid model for video classification that extracts features from the video frames using a ConvNet. We use transformer blocks to encode the spatial features, consisting of layers of multi-head attention and Bi-LSTM. The sequential output of multi-head attention and Bi-LSTM are combined and passed to a feed-forward network. Then, we perform layer normalization similar to the Vision Transformer Encoder. To perform classification, we attach a global average pooling layer at the end of transformer blocks and pass the output to a linear classifier with a softmax activation function. We test our model on classifying invasive larvae from water sample videos and compare it with state-of-the-art video classification models.

2 Related Work

Invasive dreissenids mussels have been spreading in the United States for decades, but application of machine learning in this area has been limited. Tracking invasive species[7] is generally done manually using a microscope with cross-polarized light[17]. Due to the rapid spreading of invasive dreissenid larvae[28], it is necessary to use a video-based recognition system to check for Zebra and Quagga mussel larvae early and often[22]. Our invasive species recognition is based on video samples collected from the Colorado River, Davis Dam (AZ)[31].

Since the introduction of ConvNets[20] there has been a lot of research work on classifying underwater images. Many of these underwater object recognition frameworks are based on popular CNN models like AlexNet[20], ResNet[16] etc. But the classification of dreissenid veligers has some unique challenges due to microscopic size and features that are hard to distinguish even by human annotators. And dreissenid veligers can also be rare, depending on the season. As a result, there needs to be more data imbalance in recognizing invasive larvae. Our dataset is created from a video of water samples, where invasive and non-invasive larvae often have different types of movement. So, we have decided to treat this problem as a video classification problem using cropped frames taken from the video. The annotation is done by experts inspecting the tracked objects on the video and the cropped images. In the next section, we will look at recent video classification approaches and provide the background for our video recognition framework that achieves state-of-the-art in our Quagga mussel dataset.

Since the success of ConvNets[20,14] in widespread Computer Vision problems like Image classification[20], Object Detection[13] and Segmentation[2], there has been growing interest in applying them to Video recognition. Due to the massive growth of online video-based data, several large video datasets have emerged, like Kinetics[18], Moments-in-time[23] etc. Unlike images, videos require modeling spatial features and temporal relations between other frames[33]. This makes it challenging to apply traditional CNN-based frameworks for Video classification.

More recently, transformer-based networks[32] have achieved state of the art in several key areas of machine learning, including text summarizing[12], image classification[10], segmentation[4], detection[5] etc. Transformers are based on two main ideas: 1. Self Attention mechanism to model long-term dependencies between sequences. and 2. Pre-training on a large dataset and then fine-tuning on a smaller dataset, which significantly improves the accuracy for fine-tuned tasks[19]. Generally, transformer-based video recognition frameworks feed frame-level patches to the transformer with temporal attention or use a combination of spatiotemporal attention, often using CNN to create patches[1,3].

The recently introduced Vision Transformer (ViT) by Dosovitskiy et al.[10] creates multiple patches from 2D images, performs linear projections to get 1D tokens, and then utilizes transformer blocks with a final MLP layer for classification. Transformer-based video recognition frameworks generally use a similar approach to create patches from every video frame and use a space-time attention-based transformer[1]. Gedas Bertasius et al.[3] have compared the different types of attention like Space Attention, Joint Space-Time Attention, and Axial Attention for video recognition in their Timesformer model, etc.

It is generally well-understood how much attention layers contribute to ViT's success. But, LSTM-based model Sequencer[30] has tried token mixing in vision architectures using only LSTM and achieved state of the art on ImageNet classification benchmark. One attractive property of LSTM is that it learns to map an input sequence of variable length into a fixed-dimensional vector representation. Standard LSTMs are generally better at classifying sequential features

than an MLP[29], especially with long-range sequences. At the base level, a video representation framework must encode the spatial features and understand the temporal relation between frames. So, to model the material connection between different frames, we use transformer blocks consisting of multi-head attention followed by an LSTM layer.

In this paper, we develop a video recognition framework for classifying invasive species. Our model is based on frame-level patches fed to an attention-LSTM-based transformer. We propose a variant of our model that is convolution-free and faster to train while achieving comparable accuracy. We compare the performance of our model with space-time transformer-based architecture ViViT[1] and Long-term Recurrent CNN[9].

3 Proposed Method

In this section, we introduce the Attention-LSTM model as shown in Figure 2. Firstly, we discuss the Vision transformer architecture and preliminary background on LSTM, introduce the Attention-LSTM model and its components, and based on that, we develop several architectures for video classification.

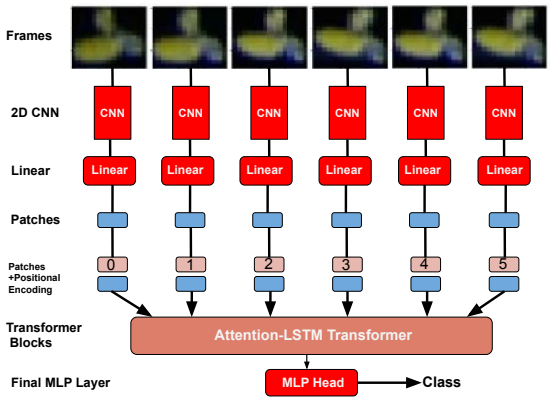

Fig. 4. Model Overview: We extract features from video frames using a Convolution Neural network, add positional encoding to the flattened patches and feed them to the Attention-LSTM transformer. We add a final MLP layer to classify from the resulting sequence.

The original transformer introduced by Vaswani et al.[32] received input as a 1D sequence of tokens. For 2D images, Vision transformer (VIT)[10] creates N patches of 2D tokens, flattens the tokens, and then employs a trainable linear projection to get tokens of D dimensions. Along with that, standard 1D positional encoding is added to the tokens. These tokens are passed through transformer encoders consisting of alternating layers of multi-headed self-attention,

layer normalization, and MLP blocks. The MLP contains two layers with a GELU non-linearity.

If we consider z_l as a sequence of tokens given as input to the transformer, then the behavior of the transformer is described by the following equations: $y^l = \text{MSA}(\text{LN}(z^l)) + z^l$ and $z^{(l+1)} = \text{MLP}(\text{LN}(y^l)) + y^l$

Here MSA is multi-headed self-attention, LN is layer normalization, and MLP is a multi-layer perceptron. In case multiple transformer layers $z^{(l+1)}$ can be used as an input to the next transformer block.

We can think of a video as temporally ordered sequence of image frames like $(x_1, x_2, x_3, \dots)$ and $x_1 \in \mathbb{R}^{(H \times W \times C)}$, where (H, W) is the resolution of the images and C is the number of channels. Our model would use input in the form of $V \in \mathbb{R}^{(H \times W \times C \times F)}$, where F is the number of frames. We would create F 1D patches with positional encoding and pass them to the Attention-LSTM-based transformer. Figure 4 provides an overview of the model.

LSTM

LSTMs excel in sequential data tasks like speech recognition, image, and video captioning. Before Transformers, LSTMs were NLP leaders. They're also crucial in video recognition, with Convolutional LSTMs predicting frames. LSTMs use hidden units for memory, employing three gates: input gate i_t, forget gate f_t, and output gate o_t. The forget gate f_t decides what information to discard from the cell state. The input gate i_t combines values to update and candidates to add. The output gate o_t determines the unit's output. This structure enables LSTMs to retain long-term dependencies. While CNNs are great at processing spatial features, LSTM models are great at retaining temporal association. So, for sequential data, the LSTM recurrence mechanism helps achieve better performance from Transformers.

Multi-Head Attention

MultiHead Attention is crucial in the Vision Transformer model, excelling in various Computer Vision tasks. It enables the model to focus on different patches and grasp frame relationships. The process involves projecting input into query (Q), key (K), and value (V) transformations, using a parameter d_k for key dimension. The attention score is then computed according to theis equation : [32] : $\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V$

here, T denotes transpose. Output is computed by multiplying Attention scores with value V, maintaining input shape. It's replicated for longer sequences, attending to parts differently.

3.1 Model Architecture

This section will provide details of the Attention-LSTM model architecture, which has two main components: the Patch Encoder and the Attention-LSTM transformer block.

Patch Encoder :

The Vision Transformer feeds the transformer's linear projections of flattened patches along with positional encoding. In our case, we create patches from every

temporal video frame. The goal here is to develop compact features that capture the visual information for every frame. For transformer-based video recognition, several architectures have proposed 2D CNN[3,10] or 3D CNN[1] based models for patch encoding. Due to the rapid development of deep learning, CNNs[20] had great success in large-scale image recognition problems making it the best candidate for feature extraction. So, for patch encoding, we apply a ConvNet for every frame. We add temporal positional encoding to flattened feature vectors and feed them to the Attention-LSTM transformer.

For video processing, we decompose each frame into N non-overlapping patches, each of size $A \times B$, such that the N patches span the entire frame. In case the number of frames in a video is larger than N, we uniformly randomly select N frames, maintaining their increasing temporal order. We apply ConvNet to create features from these patches, where each patch x is given by $x_t = \text{ConvNet}(f_i)$, where f_i is of size $(A, B, 3)$. We flatten these patches into vectors $x_t \in \mathbb{R}^{(A \times B \times 3)}$. Here, $t \in \{1, \ldots, N\}$ denotes the temporal location of the frame. We linearly map each patch $x(t)$ into a flattened embedding vector z. Finally, the sequence of tokens going to the transformer encoder is as follows:

$$z = [z_{cls}, x_1, x_2, \ldots, x_N] + p \tag{1}$$

where the projections $x_1, x_2, \ldots$ are created by a convolutional operation. An optional learned classification token z_{cls} is prepended to this sequence, similar to the BERT Transformer. A learnable positional embedding, p is added to the tokens to retain positional information.

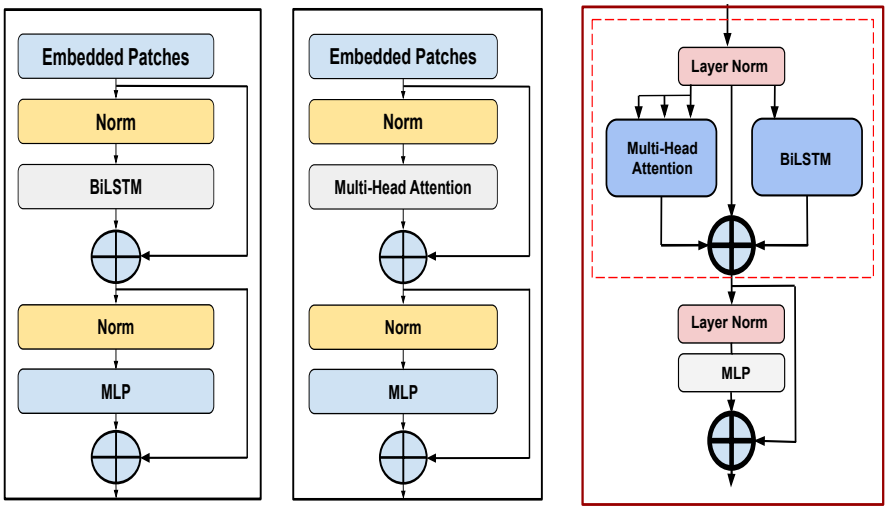

Fig. 5. 1. Sequencer, 2. Transformer, and 3. Attention-LSTM Block. 1. A Sequencer Block consists of a BiLSTM layer. 2. In contrast Transformer block consists of Multi-Head attention. 3. Attention-LSTM combines MultiHead Attention and LSTM layer.

3.2 Attention-LSTM Layer

In the last few years, several different variants of transformers have appeared. The sequencer model has replaced attention with Bi-LSTM. This paper tries to combine the strength of LSTM and attention to create a practical transformer module for long-range fine-grained features. So, we propose a new architecture combining a BiLSTM with multi-head attention and concatenating the results with a residual connection. Like Vision Transformer, MLP block is applied at the end of the transformer and residual connections, layer normalization after every transformer block. Incorporating an attention mechanism with LSTM makes our model highly effective—a comparison between different diagrams is shown in Figure 5.

The tokens (z) created by the patch encoder are passed through layers of multi-headed self-attention (MSA), BiLSTM, layer normalization (LN), and multi-layer perceptron (MLP):

$$y^l = \text{MSA}(\text{LN}(z^l)) + z^l + \text{BiLSTM}(\text{LN}(z^l)) \tag{2}$$

$$z^{(l+1)} = \text{MLP}(\text{LN}(y^l)) + y^l \tag{3}$$

Like ViT, multiple transformer layers $z^{(l+1)}$ can be used to input the next transformer block. At the end of transformer blocks, we merge the patches using a global average pooling layer and perform the final classification using a softmax activation. We used a cross-entropy loss function with softmax activation in the final layer.

Table 1. Model Details: Details of the Transformer model variants, the number of transformer layers, transformer heads, and the number of parameters for an image of size $(28, 28, 3)$.

Model Type	Parameters	Layers	Heads	Patch size
Att-LSTM-S (2D CNN)	$47k$	2	2	32
Att-LSTM-S (Linear)	$183k$	2	2	32
Att-LSTM-S (2D CNN)	$446k$	2	2	64
Att-LSTM-S (Linear)	$292k$	2	2	64

3.3 Model variations

In this section, we present several different model variants and evaluate them on different datasets. We have used model variants depending on the size of the dataset.

ConvNet-based : Our base model is based on one convolution layer, followed by linear projections for patch creation. We used a filter size of (3×3)

and a stride size of 1, then created flattened patches with linear projections and added learnable positional embedding.

Linear Projections : Here we use linear projections on flattened frames to create patches and add positional encoding for every temporal step.

We create several variations of our model for training on different datasets by changing the number of transformer layers and attention heads. We use a similar naming convention to that of Vision Transformer. We consider Attention-LSTM-Small ($L = 2, H = 2$), Attention-LSTM-Base ($L = 6, H = 6$) with patch size of 32 & 64 and present the number of parameters for each of them in Table 1. Here the number of layers is given by L, and the number of heads in the attention layer is provided by H.

4 Invasive Species Dataset

Our dataset is processed from videos of water streams. We have used a Kalman Filter-based proprietary algorithm for tracking and cropping larvae images from videos. So a set of frames is available for every organism. The dataset has two types of objects: Invasive and non-invasive. It contains cropped images of 6,905 organisms, with 1,220 invasive organisms (quagga mussels) and 5,685 non-invasive organisms. There are a total of 221,702 images across two organisms. The dataset is imbalanced towards non-invasive species as it takes around 85 percent. So, along with accuracy, we report F1 Score for invasive species as evaluation criteria. Every organism has a minimum of 6 to a maximum of 42 frames. In our classification model, we used six frames with a size of $(28 \times 28 \times 3)$. We used 70% of the data for training and validation and 30% as a test.

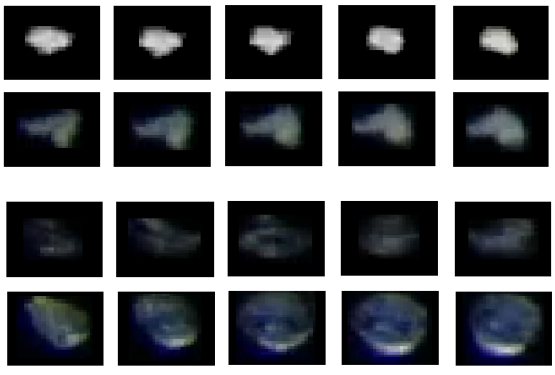

Fig. 6. These are five images of the same organism plotted in each row. The first two rows are from non-invasive organisms, and the next two are from dreissenid veligers. Notice the movement of dreissenid veligers as they progress through the water. This shows the importance of taking motion cues of invasive species into account while modeling fine-grained features of the organisms.

5 Empirical Evaluation

We train models with patch sizes of 64, 32. We train all models with the Adam optimizer and cross-entropy loss with a batch size 32. The Adam optimizer is applied with an initial learning rate of 0.001 and an exponential decay rate of 0.9, which we found especially useful for training larger datasets.

We evaluate the Attention-LSTM model primarily on our quagga mussel dataset, test it with three different backbone architectures, and present the detailed results in Table 2. We offer the results with different variants, like Attention-LSTM-Small, and Attention-LSTM-Base, with varying sizes of patch and backbone structures. We report the test accuracy for the invasive species dataset and compare the results with ViVIT and LRCN models and a single image VGG CNN.

Table 2. Comparison with state of the art on invasive larvae dataset. Results are based on ten experiments with 100 training epochs, using H and L for transformer head and layer size. Training time per epoch is labeled as Time. VGG CNN relies on a single image, while accuracy for organisms is calculated via majority voting. All the other models are based on the first five images of an organism. The F1 score is the key metric for estimating invasive species' presence and prevalence. The results show that Attention-LSTM significantly improves and outperforms other video recognition models across comparable parameter bands.

Method	Patch Size	H	L	Time	F1 Score(Invasive)	Accuracy
Att-LSTM-S (Linear)	32	2	2	1s	99.18±0.84%	99.71±0.30%
Att-LSTM-S (2D CNN)	32	2	2	2s	99.15±1.5%	99.51±0.87%
Att-LSTM-B (Linear)	64	2	2	1s	98.72±1.95%	99.34±1.22%
Att-LSTM-B (2D CNN)	64	2	2	4s	98.56±2.71%	99.51±0.87%
Att-LSTM-S (Linear)	32	6	6	1s	99.65±0.37%	99.87±0.13%
Att-LSTM-S (2D CNN)	32	6	6	4s	99.75±0.2%	99.32±1.89%
Att-LSTM-B (Linear)	64	6	6	3s	98.28±3.83%	99.27±1.53%
Att-LSTM-B (2D CNN)	64	6	6	4s	99.46±0.93%	99.86±0.33%
ViVIT	32	2	2	1s	92.39±1.64%	97.33±0.54%
ViVIT	32	6	6	3s	92.78±3.52%	97.48±1.5%
ViVIT	64	2	2	2s	93.57±1.28%	97.75±0.44%
ViVIT	64	6	6	3s	94.75±1.35%	98.18±0.43%
LRCN	-	-	-	3s	89.26±2.57%	96.19±0.77%
VGG CNN	-	-	-	-	90.8 ± 0.21 %	93.6 ± 0.11 %

The results demonstrate that Attention-LSTM-based video recognition achieved 99% accuracy in classifying invasive and non-invasive images. Attention-LSTM improves the accuracy significantly compared to other video recognition frameworks like ViVIT, LRCN, or VGG-based single image CNN, while also being faster to train.

6 Conclusion

Invasive species have a detrimental impact on the aquatic environment, leading to infrastructural damage. We present an Attention-LSTM-based transformer for video-based, end-to-end recognition of aquatic invasive species larvae. The combination of LSTM and Multi-Head Attention allows our model to recognize more fine-grained features from videos. We achieves a remarkable 99% F1 score in accurately identifying invasive larvae from water sample videos. Future efforts will focus on categorizing dreissenid veligers based on their life stage and providing recommendations to address them effectively. These methods show significant potential for enhancing the effectiveness of early detection programs for invasive dreissenid mussels.

References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: CVPR. pp. 6836–6846 (2021)
2. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)
3. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: ICML. vol. 2, p. 4 (2021)
4. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.: Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)
5. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. arXiv preprint arXiv:2205.08534 (2022)
6. Chowdhury, S., Hamerly, G.: Recognition of aquatic invasive species larvae using autoencoder-based feature averaging. In: International Symposium on Visual Computing. pp. 145–161. Springer (2022)
7. Churchill, C.J., Baldys, S.: USGS zebra mussel monitoring program for north Texas. US Department of the Interior, US Geological Survey (2012)
8. Connelly, N.A., O'Neill, C.R., Knuth, B.A., Brown, T.L.: Economic impacts of zebra mussels on drinking water treatment and electric power generation facilities. Environmental management **40**(1), 105–112 (2007)
9. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR. pp. 2625–2634 (2015)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2020)
11. Gao, Z., Tan, C., Wu, L., Li, S.Z.: Simvp: Simpler yet better video prediction. In: CVPR. pp. 3170–3180 (2022)
12. Guo, M., Ainslie, J., Uthus, D., Ontanon, S., Ni, J., Sung, Y.H., Yang, Y.: Longt5: Efficient text-to-text transformer for long sequences. arXiv preprint arXiv:2112.07916 (2021)
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)

14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
16. Jiang, Z., Zhao, C., Wang, H.: Classification of underwater target based on s-resnet and modified dcgan models. Sensors **22**(6), 2293 (2022)
17. Johnson, L.E.: Enhanced early detection and enumeration of zebra mussel (dreissena spp.) veligers using cross-polarized light microscopy. Hydrobiologia **312** (1995)
18. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
19. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. ACM computing surveys (CSUR) **54**(10s), 1–41 (2022)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM **60**(6), 84–90 (2017)
21. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: CVPR. pp. 10012–10022 (2021)
22. Lucy, F., Muckle-Jeffs, E.: History of the zebra mussel/icais conference series. Aquatic Invasions (2010)
23. Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S.A., Yan, T., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C., et al.: Moments in time dataset: one million videos for event understanding. IEEE transactions on pattern analysis and machine intelligence **42**(2), 502–508 (2019)
24. Nalepa, T.F., Schloesser, D.W.: Quagga and zebra mussels: biology, impacts, and control. CRC press (2013)
25. Nichols, S.J., Black, M.: Identification of larvae: the zebra mussel (dreissena polymorpha), quagga mussel (dreissena rosteriformis bugensis), and asian clam (corbicula fluminea). Canadian Journal of Zoology **72**(3), 406–417 (1994)
26. Schloesser, D.W., Metcalfe-Smith, J.L., Kovalak, W.P., Longton, G.D., Smithee, R.D.: Extirpation of freshwater mussels (bivalvia: Unionidae) following the invasion of dreissenid mussels in an interconnecting river of the laurentian great lakes. The American midland naturalist **155**(2), 307–320 (2006)
27. Sepulveda, A.J., Amberg, J.J., Hanson, E.: Using environmental dna to extend the window of early detection for dreissenid mussels. Management of Biological Invasions **10**(2) (2019)
28. Stokstad, E.: Feared quagga mussel turns up in western united states (2007)
29. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. Advances in neural information processing systems **27** (2014)
30. Tatsunami, Y., Taki, M.: Sequencer: Deep lstm for image classification. arXiv preprint arXiv:2205.01972 (2022)
31. Turner, K., Wong, W.H., Gerstenberger, S., Miller, J.M.: Interagency monitoring action plan (i-map) for quagga mussels in lake mead, nevada-arizona, usa. Aquatic Invasions **6**(2), 195 (2011)
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
33. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European conference on computer vision (ECCV). pp. 305–321 (2018)