2017 IEEE International Conference on Big Data (BIGDATA)

A Distributed Pipeline for DIDSON Data Processing

Liling Li, Tyler Danner and Jesse Eickholt
Department of Computer Science
Central Michigan University
Mt. Pleasant, MI, USA
eickh1jl@cmich.edu

Erin McCann and Kevin Pangle
Department of Biology
Central Michigan University
Mt. Pleasant, MI, USA
pangl1k@cmich.edu

Nicholas Johnson
Great Lakes Science Center
U.S. Geological Survey
Hammond Bay Biological Station
Millersburg, MI, USA

***Abstract*—Technological advances in the field of ecology allow data on ecological systems to be collected at high resolution, both temporally and spatially. Devices such as Dual-frequency Identification Sonar (DIDSON) can be deployed in aquatic environments for extended periods and easily generate several terabytes of underwater surveillance data which may need to be processed multiple times. Due to the large amount of data generated and need for flexibility in processing, a distributed pipeline was constructed for DIDSON data making use of the Hadoop ecosystem. The pipeline is capable of ingesting raw DIDSON data, transforming the acoustic data to images, filtering the images, detecting and extracting motion, and generating feature data for machine learning and classification. All of the tasks in the pipeline can be run in parallel and the framework allows for custom processing. Applications of the pipeline include monitoring migration times, determining the presence of a particular species, estimating population size and other fishery management tasks.**

***Index Terms*—surveillance, distributed processing, DIDSON, HDFS, classification**

I. Introduction

Within the field of ecology, there has been an increasing need to understand ecological systems at high resolutions both temporally and spatially [1], [2]. Over the years, researchers are more easily meeting these needs due to technological advances in sampling techniques that allow for increased data collection at much finer scales [3], [4]. However, a significant challenge with using relatively large ecological data has been data processing [5], [6], which can limit the scope of ecological inferences. Therefore, further improvements to data processing techniques could greatly benefit future large-scale ecological research.

Aquatic ecology is one branch of ecology that has greatly benefited from recent technological advances. For example, fisheries assessment utilizing traditional sampling methods such as stationary traps [7], [8] or seining (e.g., commercial fishing; [9]) are limited because these visual techniques do not typically generate high-resolution spatiotemporal data. Alternatively, modern fisheries assessment methods such as acoustic telemetry and sonar imaging have become very popular in recent years because they allow for more fine-scale assessment both temporally and spatially [10], [11] especially in areas where using traditional visual assessment techniques are challenging (i.e., oceans, large lakes or rivers). In particular, the use of Dual-frequency Identification Sonar (DIDSON) has been widely used for fisheries assessment and has provided large amounts of continuous surveillance data [11]. However, a significant challenge with using DIDSON is data processing because of the relatively large amount of data generated [12].

Because fish live in an aquatic environment, which is turbid and changeable, collecting data on fish presence and movement is relatively more difficult when compared to data collection for terrestrial organisms. DIDSON was originally developed to identify underwater intruders for a harbor surveillance system [13] but has widely been used in fishery management. It uses acoustic lenses to form narrow beams and receives acoustic echoes to form images. It has sufficiently high resolution and rapid refresh rate which are comparable to an optical system. Unlike optical surveillance systems, DIDSON is not affected by poor lighting conditions as it does not rely on light to work. It also has an excellent ability to acquire good images in turbid water because relatively high concentrations of suspended material in water have a very limited effect on acoustic beams [13].

To monitor fish movements, data collection may need to take place for an extend period of time (e.g., several months). A DIDSON recording device is capable of producing 30 GBs of data per day which can lead to 3 to 4 TBs of data being generated per deployment during an extended surveillance period. While real-time processing of DIDSON data may be possible in certain instances, varied types of analyses (e.g., species specific filtering and tracking) require the data to be processed multiple times and as a result it should be stored and processed off-line. This, coupled with the large amount of data, makes serial processing of the DIDSON data challenging. Consequently, we have developed a pipeline to process raw DIDSON data in a parallel fashion. The pipeline makes use of the HADOOP ecosystem [14] and can be used in conjunction with custom classifiers implemented in Keras [15].

II. Background

DIDSON has been used in fishery management for decades and a number of studies have been facilitated making use of DIDSON data. Baumgartner and his colleagues performed a study to assess the application of DIDSON on Australian water systems [16]. They concluded that the DIDSON is the most capable technology to monitor fish migrations, sample

978-1-5386-2715-0/17/$31.00 ©2017 IEEE

fish, and supplement existing assessment programs. Holmes *et al.* performed sockeye-salmon *(Oncorhynchus nerka)* counting with DIDSON data and determined that the counts produced by accompanying DIDSON software packages have high precision when comparing them with visual counts [17]. Both studies made use of proprietary DIDSON software packages and produced promising results which demonstrated the potential utilization of DIDSON data.

The DIDSON is commercial equipment and associated software for processing raw DIDSON data (e.g., Sound Metrics) have some restrictions and limitations. As a result, many researchers have developed their own assessment programs with more flexibility in processing the DIDSON data. Bothmann *et al.* developed a real-time fish classification system using DIDSON videos. The system consists of different modules for data preprocessing, feature construction, and classifier construction [18]. Through the comparison of 15 models with different features and 6 classification algorithms, they reported that a random forest tree with baseline, shape variation, and motion variables achieves the best performance on a fish classification task [18]. Butail and Paley applied an ellipsoid shape with a curvature coefficient to reconstruct 3D images of fish based on DIDSON data. Based on their model they were able to build a probabilistic framework to estimate position and shape of multiple fish in a school [19]. These studies show the fascinating potential of utilizing DIDSON images in fishery monitoring and managing tasks such as fish classification. Langkau *et al.* used the SHAPE analysis software to perform image banalization, Elliptic Fourier Analysis (EFA) and Principal Component Analysis (PCA) to assess the classification ability on DIDSON images. Through controlled experiments they found that the classification accuracy is almost 100% for large static fish templates and 83.9% for moving fish [20]. The results of the controlled experiments may not carry over to *in situ* settings, but they demonstrate the excellent capacity of DIDSON image for fish classification.

While a number of studies have been done using DIDSON data, it is important to note the limitations of these works. First, these studies often employed varied types of pre-processing of the data which may require manually marshaling the data through several different tools (e.g., target identification in a DIDSON viewer and target classification in an machine learning library). This is due to the fact that images extracted from DIDSON data often contain noise and additional artifacts such as acoustic echoes (see Figure 1 for DIDSON image containing a steelhead). Second, these works were mainly based on experimental data instead of long term observation data. It is difficult to envision how these current approaches and tool sets could scale to process a large amount of raw observation data and thus makes them difficult to implement for longer surveillance periods. This situation is exacerbated by the fact that the data may need to be processed multiple times for different analyses (e.g., different target species). To better utilize the DIDSON data and facilitate meeting fishery management demands, a semi-automated parallel data processing system is needed. Such a system should be flexible enough to support custom processing modules and fast enough to support multiple runs of off-line processing and analysis. Such a system is described here.

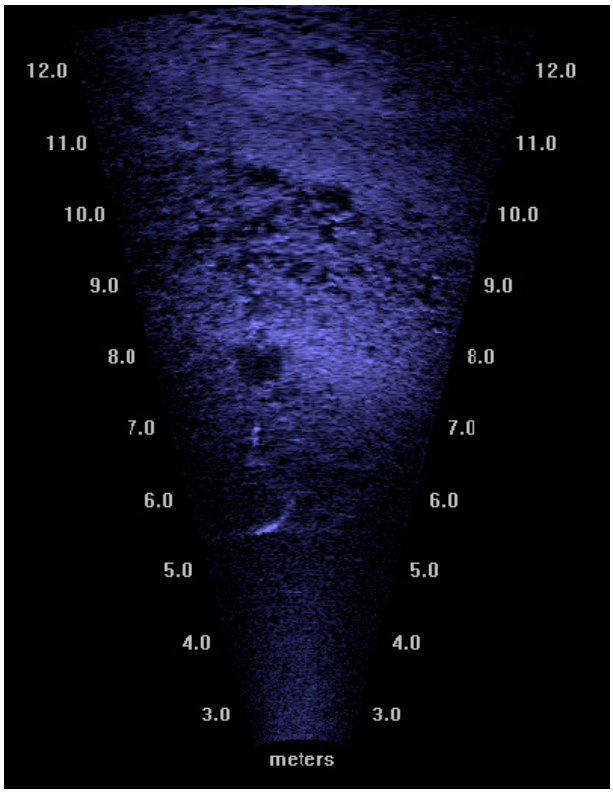

Fig. 1. Image extracted from DIDSON data showing a steelhead at approximately 5.5m from the camera. An acoustic shadow of the fish can be seen at proximately 8m.

III. Construction of the Pipeline

The driving design consideration was the need to process the data in parallel. Due to the large amounts of data and the likelihood that much of the data would be processed when being analyzed, it was decided that it would be best if the data was stored in a distributed fashion to support distributed processing. The Hadoop Distributed File System (HDFS) [21] was selected and is used to store ingested DIDSON data and intermediate results. Processing the data (e.g., filtering, tracking, classification, etc) is done through the MapReduce API and jobs dispatched through YARN.

The raw DIDSON acoustic data can be transformed and each set of sensor readings from the acoustic array can be viewed as a 2d image (i.e., a frame). Viewed sequentially these images form a video. This third dimension of the data (i.e., temporal) is useful for classification (e.g., it can be used to inform motion) and filtering (e.g., it can be used to remove noise). The pipeline retains temporal information by breaking the raw DIDSON data into blocks comprised of a temporal series of frames with a small amount of overlap. The overlap ensures that every frame existing on the HDFS has sufficient context (i.e., neighboring frames)

Figure 2 shows the overall architecture of the pipeline and its components. Raw DIDSON acoustic data (as produced by the DIDSON hardware) is converted into proper format by a Java program and placed on the HDFS. Filtering, tracking and target identification code run via YARN and the MapReduce API extracts possible targets as thumbnails and saves them to the HDFS. Feature generation code, run via YARN, converts thumbnails into a feature vector. Finally, the feature vector is run through a classifier (e.g., a classification model trained in Keras [15]) through a streaming job. Alternatively, the feature data can be read off the HDFS and accessed in a python script using hdfs3 [22] and then feed into a Keras model to make predictions in a serial fashion on a local machine. Once ingested, a few MapReduce jobs are needed along with shuffling some intermediate data through the HDFS. These tasks are managed by a shell script which drives the overall process.

A. Ingesting Raw Data

The input to the pipeline is data that is generated directly from the DIDSON device. The data is in binary format which consists of 512 bytes of metadata followed by the the acoustic data. The acoustic data is segmented by frames (i.e. a set of readings of the acoustic array at a particular time) and each is started with a header that contains information such as frame number, a time stamp, transmission mode, etc. The acoustic data represents reflectance of acoustic waves at a particular headings and time and as a result to be visualized it must be transformed to Cartesian coordinates and pixel intensities [23]. Existing Matlab code [24] was adapted and integrated with the Hadoop API to create a stand-alone Java program to convert acoustic data to a series of grayscale images. The grayscale images are stored as a record in a SequenceFile and saved directly to an HDFS. The conversion program runs serially but since the DIDSON is capable of generating a number of data files, one for each 30 minute interval, it is possible to ingest the data in parallel by running several instances of the program at the same time.

A SequenceFile is a Hadoop class used for grouping and storing key value pairs in an HDFS. As MapReduce works independently on key value pairs, an important design consideration was ensuring that the data was properly segmented and had sufficient context for processing (e.g., neighboring frames can be used for filtering and tracking). This is achieved by placing a series of frames in a SequenceFile record and overlapping the start and stop frames of the records. The key for each record can contain metadata such as the name of the source file, overall offset from the beginning of the source file, and frame number. The value is an array of bytes. The first 4 bytes of the byte array contain the height and width encoded as short integers and the remaining bytes represent the pixel values of the frames stored in row major order with the first dimension being the frame number. Figure 3 illustrates how a file produced from a DIDSON device (i.e., .ddf file) can be converted to a SequenceFile which contains a number of key-value records.

B. Data Filtering

Depending on deployment conditions, the images stemming from the DIDSON device can contain a significant amount of noise. This is to say that from one frame to the next there are often significant shifts in the value of a particular pixel, seeming without warrant. As a result it can be challenging to identify movement by techniques such as background subtraction due to the high variance of some pixels. To address this, a number of filtering schemes can be used and several filtering schemes were implemented and deployed within the pipeline. One approach is the median filter which calculates the median of neighboring pixel values as the filtered value. It works on each 2d frame (i.e., one image in the video stream) and removes noise to produce a smoothed effect and can be implemented efficiently [25]. Another approach is a varied MATLAB Wiener2 filter [26] which is designed to remove Gaussian white noise. The original MATLAB Wiener2 takes spatial neighboring pixels as input while the varied filter takes temporal neighboring pixels as input to estimate the local mean and standard deviation for each pixel. By defining neighbors on temporal dimension, the filter can reduce the white noise appearing in time series. This is possible since segments of frames are package together in one record during data ingestion and as a result a third dimension (i.e., time) can be used for data filtering and cleansing.

In addition to general filtering strategies, species-specific approaches can be used to limit the areas of a frame where a particular fish might be present. Consider as an example a sea lamprey*(Petromyzon marinus)*. As the DIDSON was deployed with a field of view perpendicular to the flow of water, upstream and downstream movements correspond moving straight to the left or right of the frame. Sea lamprey usually swim up or downstream and seldom move rapidly from bank to bank. Sea lampreys also have a slender, elongated shape. Therefore, when looking for movement a two stage clustering approach was used. In the first stage localized areas of movement were identified by counting the number pixels that differed significantly in value from the baseline average. This indicated an area of local movement. In the second stage, the number of small clusters in a 17 by 6 pixel area from the first stage were calculated. This was done due to the lamprey's slender shape and their usual left and right movements across the frame. Portions of the frame that exhibited these patterns of difference from the background were flagged as potential areas of interest. Further support was given by adjacent frames. If 40% of the neighboring frames also contained nearby areas of interest then the area deserved further attention and could be extracted for classification.

C. Processing Data and Custom Analyses

After filtering, the data can be processed via further use of the Hadoop API or streaming. Of particular interest was the ability to extract a series of small windows from frames and pass them through a machine learned model for classification. Areas of interest can be defined as areas of movement in a frame. Large amounts of concentrated movement in the same

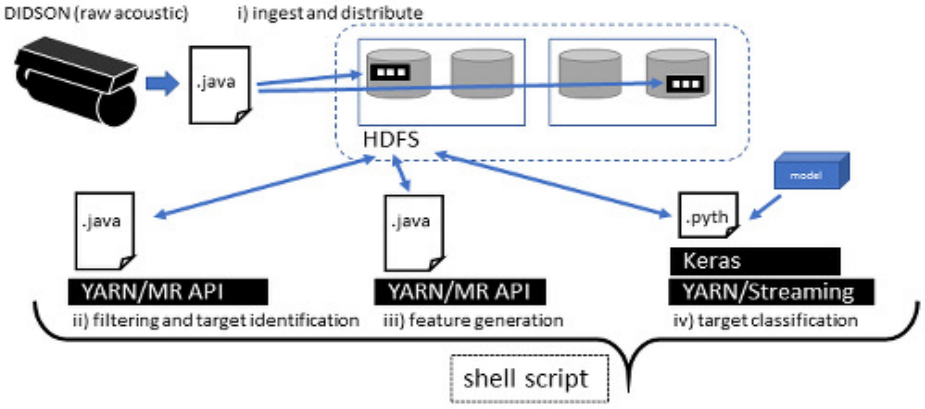

Fig. 2. A pipeline for parallel processing of DIDSON data.

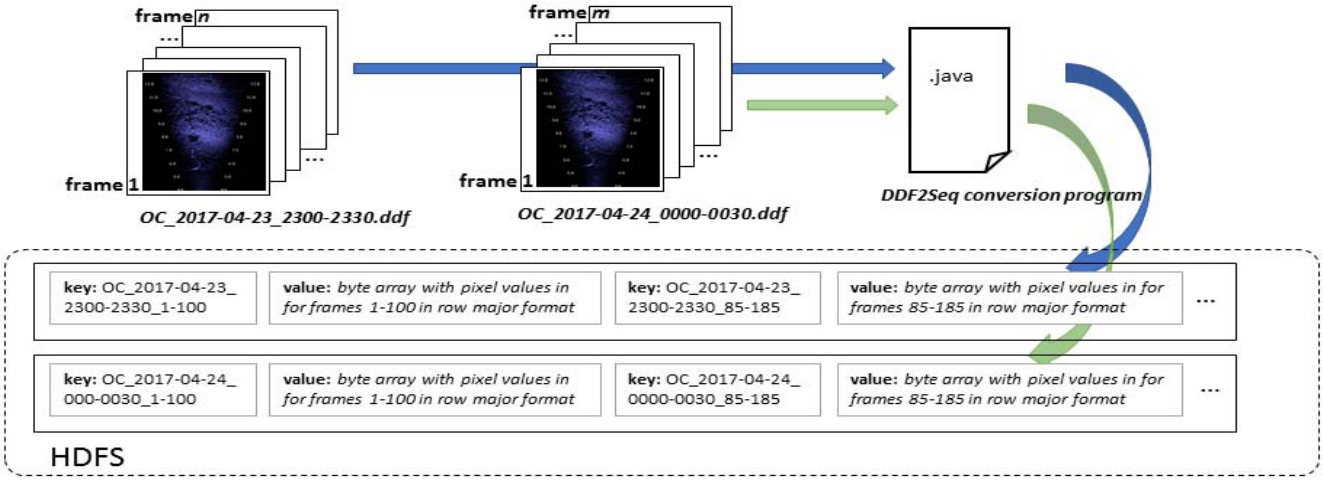

Fig. 3. An illustration of coverting raw acoustic data to key-value pairs containing overlapping frames stored in a byte array.

area over several frames provides further support that the movement could be caused by a fish. A box 100 by 35 is placed around the center of movement and thumbnail is extracted. Thumbnails for the preceding 3 frames and following 3 frames are extracted and therefore can inform movement. Thumbnail generation can be accomplished through a distributed job making use of the Hadoop MR API. The extracted thumbnails are saved as key-value records in a SequenceFile. The value is a byte array with each byte representing the grayscale intensity of a pixel. The key contains contents of the source key (i.e., source file name, frame number) and also the location (i.e., Cartesian coordinates of the center of the thumbnail box along with the frame number of the center thumbnail).

From the thumbnail data, a feature vector can be generated. The feature vector is the input to a classifier and represents data, if properly formatted, that is useful for the classification task at hand. At present, the feature file generated is a text file in which each line represents one set of thumbnails. The metadata from the key is prepended to the feature data so that it accompanies the data through the pipeline. The feature file is saved on the HDFS so that classification can be performed in parallel, if desired. The needed Keras libraries, model and python code to apply the model can be packaged and distributed through a Hadoop streaming job. Alternatively, the feature file can be read directly from the HDFS using the libhdfs3 library [22] and classification can be performed in serial fashion on a single machine. Species classification is one example of the type of data processing which can be accomplished with this pipeline.

IV. Discussion

A. Benefits of the Pipeline

This pipeline offers many benefits over existing approaches which utilized DIDSON data for fishery management. First, in terms of speed and flexibility, this pipeline greatly exceeds what was previously possible. The commercially available software (e.g., SoundMetrics) is ideal for viewing and analyzing small amounts of data but limited in its ability to process large amounts of data (i.e., more than a few days worth of

surveillance data). It is also limited in terms of extensibility to support customized tracking or classification. Previously, support for advanced customized filtering was largely limited to a process in which the raw DIDSON data was encoded to video (e.g., AVI) and then video processing libraries such as OpenCV [27], [28] were used for advanced filtering, target tracking, or target classification. Second, with the pipeline presented here, a data analysis process can be scripted or automated in ways which are not possible when making use of a GUI tool.

B. Applications of the Pipeline

This pipeline can be viewed as a general framework for distributed processing of DIDSON data. After ingestion and storage on the distributed file system, any number of applications are possible. By storing the source filename, date, time and start frame of each clip as the key for a given record, specific frame numbers and locations can be provided and specific thumbnails extracted. This was useful for curating training data to create classifiers for specific aquatic species. With the help of experts and additional means of validation (e.g., monitoring fish with passive integrated transponders, cameras), a list of known targets was created and placed on the HDFS. A MapReduce program was created to use the list and extract feature data for these targets in parallel. This provides a quick and easy means to make changes to the feature data (e.g., apply different filtering schemes, change the size of thumbnails, change the number of frames used). This approach can also be used to quickly cull the data and create videos of areas of interest (e.g., complete view of a target being classified as a sea lamprey, particular times in a day, etc.). Based on this pipeline we developed an application which took a list of targets classified as sea lamprey by the pre-trained classification model and created an MP4 video comprised of segments of DIDSON derived video with all of the the targets highlighted. This provides sufficient context and easy of access for experts to view the footage and visually check for sea lamprey activity over a long period without viewing the entire video. It also aids in the validation of machine learning classifiers developed for species identification and other tasks.

From the ecological system perspective, researchers largely have been limited by the inability to handle big data and there are many applications that could benefit if more data is available. As an example, consider the ecological implications of global commerce. Ecological systems, including aquatic system, have lost their natural isolations and many species have been introduced into new territories. These could cause species invasions which pose great harm to aquatic ecosystems and lead to implications for fisheries management and invasive species control [29], [30]. For example, the sea lamprey *(Petromyzon marinus)*, a parasitic fish native to the Atlantic Ocean, is a high impact invasive species in Laurentian Great Lakes, greatly contributing to the collapse of important fisheries during its peak abundance in the 1960s [31], [32]. Given that fish invasion has been and continues to be a major hazard to the Great Lakes fishery, government, academia, and fishery commissions are working together to control invasive species to benefit the fishery and local ecosystem. Within those fishery management techniques and measurements, fish observation and classification are among the basic procedures to provide fish abundance estimation, invasive species control, and endangered fish protection. A distributed processing platform such as the one described here can help digest large amounts of surveillance data and give timely feedback for decision making. The capability to process big data could also scale the fisheries management techniques and allow higher resolution analysis, which is importance to advance fishery management and environmental protection.

While the pipeline described here was developed to handled DIDSON data, it could be adapted to process other types of surveillance data with minimal effort. This is because only the first step in the pipeline (i.e., the format conversion) is specific to the DIDSON data format and the remaining modules are general to the SequenceFile format which is binary data representing video data in a three dimensional array. Any other type of video can be easily converted to such a format so as to make use of the pipeline.

C. Limitations and Future Developments

There are a few notable limitations associated with this type of processing pipeline. In practice it requires a Hadoop cluster to process the data and most do not have a Hadoop cluster on premise. Cloud based options ameliorate this situation to a point but are costly. Furthermore, the pipeline only provides a framework for processing and customizations to filtering and thumbnail extraction need to be done at the source code level.

Further points of development for this work include conversion of shell scripts used to drive the pipeline to Scala and easing distribution. Scala is a natural fit for the driving logic of the pipeline since it can integrate seamlessly with the existing Java code and is executable on Spark. This will reduce the amount of intermediate data generated during execution of the pipeline and by refactoring the filtering code, most changes to data processing could be accomplished via function calls and chaining in the driver script. At present, changes require adjustment to the source code for individual pipeline components. To increase access and ease of installation, a containerized version of pipeline could be created making use of a stand-alone deployment of Spark. This would allow the pipeline to be used to script processing on smaller datasets. Processing of larger datasets would still require a distributed file system (e.g., HDFS).

While the pipeline was designed primarily as a means to apply custom filtering, tracking, and classifiers to large amounts of DIDSON data, it can be extended for additional functionality. Real-time applications of target identification and classification could be accomplished through streaming in raw DIDSON data. The rate at which the data is generated (e.g., 10 frames per second) and the speed at which existing classifiers can be applied would make real-time surveillance of waterways possible.

V. Conclusion

We have presented a distributed pipeline for processing DIDSON data. The pipeline utilizes the Hadoop ecosystem and performs processing tasks through a series of YARN jobs. In terms of processing capacity, the pipeline presented here is a stark improvement over existing tools for processing DIDSON data, which largely work in a serial fashion and not capable of dealing with the amount of data that a DIDSON device can generate over an extended deployment. The pipeline is capable of ingesting raw DIDSON data, transforming the acoustic data to images, filtering the images, detecting motion, extracting targets, generating features for machine learning and classification. All of the tasks in the pipeline can be run in parallel and the framework allows for custom processing. This pipeline is not only limited to DIDSON data processing but can be adapted to other types of surveillance data and supports higher resolution analysis in many fields.

Acknowledgment

The authors would like to acknowledge Michigan Sea Grant, which supported Erin McCann during her time at Central Michigan University. The authors would also like to thank Emmaleigh Wilson for her work in prototyping some image filtering and target tracking algorithms.

Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S Government. We thank the Great Lakes Fishery Commission for funding the collection of the DIDSON data.

References

[1] S. E. Hampton, C. A. Strasser, J. J. Tewksbury, W. K. Gram, A. E. Budden, A. L. Batcheller, C. S. Duke, and J. H. Porter, "Big data and the future of ecology," *Frontiers in Ecology and the Environment*, vol. 11, no. 3, pp. 156–162, 2013. [Online]. Available: http://dx.doi.org/10.1890/120103

[2] "Macrosystems ecology: big data, big ecology." *Frontiers in Ecology and the Environment*, vol. 12, p. 3, 2014.

[3] F. Cagnacci, L. Boitani, R. A. Powell, and M. S. Boyce, "Animal ecology meets gps-based radiotelemetry: a perfect storm of opportunities and challenges," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 365, no. 1550, pp. 2157–2162, 2010. [Online]. Available: http://rstb.royalsocietypublishing.org/content/365/1550/2157

[4] C. J. Brown, S. J. Smith, P. Lawton, and J. T. Anderson, "Benthic habitat mapping: A review of progress towards improved understanding of the spatial ecology of the seafloor using acoustic techniques," *Estuarine Coastal and Shelf Science*, vol. 92, pp. 502–520, May 2011.

[5] Y. Ma, H. Wu, L. Wang, B. Huang, R. Ranjan, A. Zomaya, and W. Jie, "Remote sensing big data computing," *Future Gener. Comput. Syst.*, vol. 51, no. C, pp. 47–60, Oct. 2015. [Online]. Available: http://dx.doi.org/10.1016/j.future.2014.10.029

[6] A. Karmas, A. Tzotsos, and K. Karantzalos, "Geospatial big data for environmental and agricultural applications," in *Big Data Concepts, Theories, and Applications*. Springer, 2016, pp. 353–390.

[7] T. R. Binder, R. L. McLaughlin, and D. G. McDonald, "Relative importance of water temperature, water level, and lunar cycle to migratory activity in spawning-phase sea lampreys in lake ontario," *Transactions of the American Fisheries Society*, vol. 139, no. 3, pp. 700–712, 2010.

[8] E. M. McCann, N. S. Johnson, and K. L. Pangle, "Corresponding long-term shifts in stream temperature and invasive fish migration," *Canadian Journal of Fisheries and Aquatic Sciences*, 2017.

[9] J. H. Clark, A. McGregor, R. D. Mecum, P. Krasnowski, and A. M. Carroll, "The commercial salmon fishery in alaska," *Alaska Fishery research Bulletin*, vol. 12, no. 1, pp. 1–146, 2006.

[10] N. E. Hussey, S. T. Kessel, K. Aarestrup, S. J. Cooke, P. D. Cowley, A. T. Fisk, R. G. Harcourt, K. N. Holland, S. J. Iverson, J. F. Kocik, J. E. Mills Flemming, and F. G. Whoriskey, "Aquatic animal telemetry: A panoramic window into the underwater world," *Science*, vol. 348, no. 6240, 2015. [Online]. Available: http://science.sciencemag.org/content/348/6240/1255642

[11] F. Martignac, A. Daroux, J.-L. Bagliniere, D. Ombredane, and J. Guillard, "The use of acoustic cameras in shallow waters: new hydroacoustic tools for monitoring migratory fish population. a review of didson technology," *Fish and fisheries*, vol. 16, no. 3, pp. 486–510, 2015.

[12] M. L. Keefer, C. C. Caudill, E. L. Johnson, T. S. Clabough, C. T. Boggs, P. N. Johnson, and W. T. Nagy, "Inter-observer bias in fish classification and enumeration using dual-frequency identification sonar (didson): A pacific lamprey case study," *Northwest Science*, vol. 91, no. 1, pp. 41–53, 2017.

[13] E. Belcher, W. Hanot, and J. Burch, "Dual-frequency identification sonar (didson)," in *Proceedings of the 2002 Interntional Symposium on Underwater Technology (Cat. No.02EX556)*, 2002, pp. 187–192.

[14] The Apache Software Foundation. (2017) Hadoop. [Online]. Available: http://hadoop.apache.org/docs/r2.7.4/

[15] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[16] L. J. Baumgartner, N. Reynoldson, L. Cameron, J. Stanger *et al.*, "Assessment of a dualfrequency identification sonar (didson) for application in fish migration studies," *NSW Department of Primary Industries-Fisheries Final Report Series*, no. 84, 2006.

[17] J. A. Holmes, G. M. Cronkite, H. J. Enzenhofer, and T. J. Mulligan, "Accuracy and precision of fish-count data from a dual-frequency identification sonar(didson) imaging system," *ICES Journal of Marine Science*, vol. 63, no. 3, pp. 543–555, 2006.

[18] L. Bothmann, M. Windmann, and G. Kauermann, "Realtime classification of fish in underwater sonar videos," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 65, no. 4, pp. 565–584, 2016.

[19] S. Butail and D. A. Paley, "3d reconstruction of fish schooling kinematics from underwater video," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2438–2443.

[20] M. Langkau, H. Balk, M. Schmidt, and J. Borcherding, "Can acoustic shadows identify fish species? a novel application of imaging sonar data," *Fisheries Management and Ecology*, vol. 19, no. 4, pp. 313–322, 2012.

[21] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, May 2010, pp. 1–10.

[22] Continuum Analytics, "hdfs3," http://hdfs3.readthedocs.io/en/latest/, 2016.

[23] S. Negahdaripour, "Calibration of didson forward-scan acoustic video camera," in *OCEANS, 2005. Proceedings of MTS/IEEE*. IEEE, 2005, pp. 1287–1294.

[24] "nilsolav:arisreader: Code for reading soundmetric aris data into matlab," https://github.com/nilsolav/ARISreader, 2017.

[25] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 1, pp. 13–18, feb 1979. [Online]. Available: https://doi.org/10.1109/tassp.1979.1163188

[26] MathWorks, "2-d adaptive noise-removal filtering - matalab winer2," http://www.mathworks.com/help/images/ref/wiener2.html, 2017.

[27] *The OpenCV Manual*, 3rd ed., http://docs.opencv.org/master/, 2017.

[28] "Open source computer vision library," https://github.com/opencv/opencv, 2017.

[29] K. S. Simon and C. R. Townsend, "Impacts of freshwater invaders at different levels of ecological organisation, with emphasis on salmonids and ecosystem consequences," *Freshwater biology*, vol. 48, no. 6, pp. 982–994, 2003.

[30] F. J. Rahel, B. Bierwagen, and Y. Taniguchi, "Managing aquatic species of conservation concern in the face of climate change and invasive species," *Conservation Biology*, vol. 22, no. 3, pp. 551–561, 2008.

[31] B. Smith and J. Tibbles, "Sea lamprey (petromyzon marinus) in lakes huron, michigan, and superior: history of invasion and control, 1936–78," *Canadian Journal of Fisheries and Aquatic Sciences*, vol. 37, no. 11, pp. 1780–1801, 1980.

[32] G. C. Christie and C. I. Goddard, "Sea lamprey international symposium (slis ii): advances in the integrated management of sea lamprey in the great lakes," *Journal of Great Lakes Research*, vol. 29, pp. 1–14, 2003.