Applying Machine Learning Algorithms on Hydroacoustic Data for Fish Species Classification

Alice Zhang, Hunter Chen, Lewei Er, Yihan Wang, Yihang Luo, Yu Zhang

Department of Statistical Sciences, University of Toronto

STA490Y: Statistical Consultation, Communication, and Collaboration

Faculty Supervisors: Dr. Vianey Leos-Barajas & Dr. Jessica Leivesley

Teaching Assistant: Mandy Yao

April 5, 2024

1. Introduction

Recreational fishing contributes significantly to the livelihood of many communities across Canada. To ensure a strong local economy, governments need to maintain a healthy fish population to safeguard sustainability. Fish population surveys are regularly conducted to obtain information on the health of these lakes. The surveys are typically conducted at a high cost financially and ecologically. The availability of advanced hydroacoustic transducers and measurement technologies opens up the potential of classifying fish species underwater from sound wave data using machine learning models without capturing them. Being able to identify fish species underwater enables more efficient monitoring of lake health and can reduce the environmental and financial cost of fish population surveys.

While previous hydroacoustic techniques could evaluate the sizes of individual fish, they lacked the ability to distinguish between different species. Recent advancements in technology offer a promising solution. Wideband acoustic transducers, which emit a broad range of frequencies in single sound pulses, have been developed. These wideband acoustics, when combined with machine learning techniques, could open up the possibility of classifying fish species solely by analyzing hydroacoustic data.

In terms of fish species classification using hydroacoustic data, algorithms such as Random Forest have been explored and achieved promising results (Gugele et al., 2021). We took a step further to explore connectionist classification models in this project, specifically, neural network's potential in performing the same task.

We collected data containing acoustic responses from two fish species by tethering individual fish under transducers emitting 249 frequencies between 45kHz and 170kHz. Each data point contained measurement of the reflection of the transducer-emitted sound waves at a time point. The primary variable of interest for each observation was a list of target strengths, indicating the intensity of sound waves reflected at certain depths, with a one-to-one correspondence with the frequency range. After exploring three different machine learning algorithms (convolutional, recurrent, and residual neural networks) to acoustic measures and testing their ability in correctly distinguishing between two fish species, we found that all three methods achieved over 80% balanced test accuracy. Moreover, extracting features from the layers of the convolutional neural network provided interpretations of salient activation patterns and the range of frequencies for each fish species.

2. Methodology

2.1 Data

Data Collection

Hydroacoustic frequency response data was collected from 34 individuals from each of the two species using three transducers in Lake Opeongo in the Algonquin Provincial Park. Each of the transducers covered different frequency modulation (FM) frequencies and worked simultaneously. Data was collected within a 35-minutes period for each fish where sound beams were emitted continuously.

Data Structure & Important Covariates

The raw data was processed and then exported with Echoview. The processed data received for analysis consists of 56,816 rows each describing one single ping, and a total of 302 columns collecting entries for five unique fish species, specified by the variable *spCode*. However, this project concentrated specifically on performing binary classification on the species Lake Trout and Smallmouth Bass, thus the two were filtered to be kept during the model training process.

In addition to columns that identified and described basic morphological characteristics of each fish, 251 of the columns presented the target strength at each of the 251 frequencies emitted from the transducers. Emitted frequencies ranged from 45kHz to 170kHz. Supplementary observational notes on the physical condition of each fish throughout the data collection process were used to evaluate the reliability of measured responses and to support data cleaning. Data were linearized and normalized with the following implementation by Dr. Jessica Leivesley (2023) as outlined in the project summary for identifying fish species using hydroacoustic data.

$$(1)\quad \sigma_u(f) = e^{\frac{TSu(f)}{10}}$$

$$(2)\quad \sigma_u(f)_{norm} = \frac{\sigma_u(f) - \sigma_u(f)_{MIN}}{\sigma_u(f)_{MAX - \sigma_u(f)_{MIN}}}$$

Where f represents frequency emitted, and $TSu(f)$ represents the uncompensated target strengths at frequency f.

Data Cleaning

For all models, individuals with missing transducers were removed to avoid missing values. Next, the 90kHz and 90.5kHz frequency columns were removed for all individuals because data collection for these two frequencies was not completed for every individual. The data was cleaned with stricter criteria when training the 1D CNN model. That is, lake trout fish that were dead or in poor conditions during data collection were removed — decisions were made based on observational notes. Second, ranges of time pings where the fish exhibited rhythmic changes in depth and abnormal activities were also removed for 1D CNN training.

2.2 Machine Learning Methods

i. Overview of the Three Models

Introduction to Neural Networks in general

We chose to explore different types of neural networks to perform fish species classification. They are usually made up of multiple layers of neurons, including one input, one output and one or more hidden layers in between. Having the ability to capture non-linearity in data, they are broadly used in classification tasks nowadays. For example, previous studies have used neural networks to classify bird species using sounds (Gupta et al., 2021). Neural network models can be optimized by adjusting model parameters. Moreover, model performance can be adjusted by modifying the weights of connections between neurons in different layers based on error signals.

We explored three different implementations of neural networks, they were: 1D Convolutional Neural Network, Recurrent Neural Network and Residual Neural Network (ResNet).

Model 1: 1-D CNN

The Convolutional Neural Network (CNN) is a computer vision model that directly takes structured grid data, i.e. spectrograms, as input. It contains convolutional layers with feature matrices that enable automatic and adaptive learning of the spatial structures of features from the input. However, the traditional use of CNN often requires spectrograms to have equal time intervals. The main problem in our data is the missing time pings, making the construction of data input more challenging (see Model 2 for data reorganization approaches and how we addressed this challenge).

As a result, our study employed a similar approach as Rieger et al. (2023) that takes 1D arrays as input data of the CNN to classify chemical properties. Mapping onto our data, we took each time ping as one data point. Each value in the 1D input array was a target strength that corresponds to one level of frequency. This input structure required minimum data reorganization and less computational power to fit 1D arrays into the model. We hoped that 1D CNN could capture some features in spatial structure from our input (e.g. peaks, tails, etc.), which would allow the classification of fish species.

Explainability

The structure of 1D CNN allowed us to extract features from its layers to visualize the most activating target strength patterns per convolution channel, as well as ranges of frequencies that show more salient responses for a certain type of fish.

To identify the most activating target strength patterns of each channel, we first extracted the output of the final convolution layer of our optimized model on the test dataset. Each row represents a channel, which was trained to identify a specific pattern. Each column is a feature vector that corresponds to a segment in the spectrogram. Then, for each channel, we examined the top 100 typical examples of subsets of a spectrum that maximize the corresponding feature variable in that channel. In the top row of Fig.1, we overlapped the receptive fields for these 100 examples and displayed 5 of them chosen at random below. These patterns were the parts that resulted in high activations.

The examination of salient receptive fields for fish classification involved investigating the classifier layer of our model. Through extracting the weights, we could learn that activations/patterns in which frequency ranges may have a higher correlation with a certain type of fish.

Model 2: Recurrent Neural Network (RNN)

A Recurrent Neural Network (RNN) is a deep learning model equipped with both feedforward and feedback connections (Baughman, D., & Liu, Y., 1995). It employs sequential and time series data which perfectly suits the hydroacoustic data we are dealing with. RNN recognizes connections between inputs and incorporates accumulated knowledge from prior inputs when processing new information (Boufeloussen, 2022). The particular structure of inputs to RNN allows us to capture the hierarchical nature in the data. In doing so, RNN has the capacity to perform binary classifications while also providing knowledge on the between and within group variations for fish regions.

Input Structure

The original data naturally exhibited a hierarchical structure with response data from each fish being grouped into regions (identified with *Region_name*). For all regions from each individual, every 5 consecutive time pings were extracted to be constructed as a new input data point. Remaining pings that were insufficient to form groups of 5 were discarded for each region, generating a collection of inputs stored in a large data array as matrices with dimensions of 5x249. That is, 5 rows of ping times and 249 columns of emitted frequencies with each entry describing the response target strength. This effectively expanded the small sample while keeping some of the hierarchical characteristics of the data. See Appx. Fig.1 for an outlined visualization.

Model Structure

We employed Long Short-Term Memory (LSTM) units of RNN to allow the alleviation of vanishing gradient issues common in RNN — the loss of information due to continuous recurrence of connections (Wang, 2021). LSTMs give more weight to early inputs, countering the progressive reduction in the gradient caused by continuous backpropagation through time (Chauhan, 2022).

Long Short-Term Memory RNNs are more robust as they deal with long-term dependencies in the data more efficiently.

The initial layer in the model was an LSTM layer, the most essential layer of our RNN model. Inputs were structured as explained above, in sequences of 5x249 matrices. The leaky version of a Rectified Linear Unit (Leaky ReLU) activation function was applied following the LSTM layer, hoping to prevent neurons from becoming inactive and outputting only zeros. Batch normalization followed to normalize previous outputs, improving stability and efficiency in training. Our RNN model also included fully connected (dense) layers and L2 regularization to prevent overfitting during the learning process. The number of layers and number of neurons as well as LSTM output units were optimized as outlined in section (ii) below. The final layer of the model was a feedforward dense layer with two output channels. The layer used a sigmoid activation function which asymptotes to 0 and 1 (Ali, 2023); thus, the outputs range within [0, 1] and are naturally interpreted as probabilities, suitable for the binary classification task required.

Model 3: ResNet

We also chose to use a ResNet in conjunction with one-dimensional Convolutional Neural Network (1D CNN) layers for the classification of fish species based on hydroacoustic data. The choice of ResNet was motivated by its proven effectiveness in handling complex classification tasks, as demonstrated by Ji et al. (2023). Their model using ResNet, which processed time-domain signals of underwater acoustic targets' radiated noise as input, achieved a recognition accuracy of 97.1% on the full ShipsEar dataset (Ji et al., 2023). This high level of accuracy underscores the potential of ResNets in hydroacoustic data analysis.

Moreover, the incorporation of 1D CNN layers in our model allowed for efficient processing of our one-dimensional hydroacoustic data. CNNs are particularly effective at automatically and adaptively learning spatial hierarchies of features, which is beneficial for our task of fish species classification. The combination of ResNet with 1D CNN layers in our model provided a robust and efficient architecture for our specific task.

Furthermore, the shortcut connections we integrated not only add the output of a preceding block to its subsequent block but also implement skips over multiple layers. Mathematically, if we denote the desired mapping as $H(x)$, we let the stacked non-linear layers fit another mapping of $F(x) := H(x) - x$. The original mappings recast into $F(x) + x$, allowing ResNet to fit a residual mapping (He, Zhang, Ren, & Sun, 2015).

ii. Optimization

This section outlines the various parameters adjusted for and the optimization procedures applied on each of the three models.

Model 1 - 1D CNN

The data set was randomly split into training, validation, and test sets with a 70%, 15%, and 15% split, respectively. We based our model optimization on the validation set to find the best model weights and reported the model performance according to the test set. In our optimization procedures, we considered the number of 1D convolutional layers and the number of convolutional channels to start. After deciding on them, we attempted to add regularizations to prevent overfitting, primarily through dropout layers and L2 regularization.

Model 2 - RNN

90% of the data were retained for training. Observations were grouped by *fishNum* to ensure no individual fish appeared multiple times in the different subsets of the data. Stratified sampling was ensured based on fish species. The remaining 10% were used for testing. We employed 5-fold cross-validation for model training; we ensured that no observations from the same individual were repeated across folds to prevent data leakage. The model was trained for 50 epochs with a batch size of 1000. Class weights were assigned in training to compensate for the imbalance between lake trout and bass. The classes were skewed 2:1, thus the minority class, i.e. smallmouth bass, was assigned 2 times the weight of lake trout.

For optimization, we conducted a random grid search for parameters including the number of LSTM units, number of neurons for dense layers, L2 regularization parameter, and dropout rates specifically for potential models with two or three dense layers. One restriction applied on the parameters was that the number of neurons in each dense layer must be smaller than or equal to the output size of its previous dense layer. I.e. suppose the search subset for node number of dense layer 1 contains {256, 128, 64}, then the largest value in that of dense layer 2 must not be greater than 64. Regularization rates were selected between {1e-6, 1e-5, 1e-4} and dropout rates were selected between {0, 0.1, 0.15}. 20 subsets of parameters were randomly sampled from all possible combinations of different parameter values. We attempted models with one, two and three dense layers following one LSTM layer. We extracted the subset of parameters and the appropriate number of dense layers that allowed the model to achieve the best performance.

Model 3 - ResNet

Similar to 1D CNN, our data was randomly split into training, validation, and testing sets with a 70%, 15% and 15% proportion, respectively. This split was done while keeping time pings from the same fish within a single set. The ResNet model was optimized on the number of filters, kernel size, activation function, and the residual blocks structure. In addition to the residual block structure, we employed a technique known as shortcut connections. By combining both low-level and high-level features extracted by the model, the model could leverage a richer representation of the target strengths, improving its ability to make accurate predictions. The configuration we have chosen for ResNet is in Table 1.

iii. Statistical Metrics

We evaluated the models' performance based on the balanced testing accuracy and testing AUC (Area-Under-Curve) values of the model trained on the test dataset. AUC values were calculated by the area under the ROC curve, which simultaneously displays the False Positive rate and True Positive Rate for thresholds ranging from 0 to 1. It summarizes the overall performance of the classifier over all thresholds.

Specificity is defined as the proportion of positive classes that are correctly predicted; in our models (except for ResNet), the positive class referred to Smallmouth Bass, so specificity was interpreted as the accuracy of predicting Smallmouth Bass. Similarly, sensitivity calculates the proportion of negative classes that are correctly predicted; as the negative class represented lake trout for 1D CNN and RNN, this measure corresponded to the accuracy of predicting lake trout. Different from the two, ResNet assigned lake trout as the positive class in training.

Balanced accuracy was used for evaluation instead of the traditional accuracy to account for the misleading performance measure resulting from an imbalanced number of samples from each class in the training data. It is calculated as an average of sensitivity and specificity and extracted from the confusion table.

$$Sensitivity = \frac{n_{TP}}{n_{TP} + n_{FN}}$$

$$Specificity = \frac{n_{TN}}{n_{TN} + n_{FP}}$$

$$Balanced\ Accuracy = \frac{Specificity + Sensitivity}{2}$$

Where n_{TP} represents the number of true positives, n_{FN} the number of false negatives, n_{TN} the number of true negatives and n_{FP} the number of false positives.

3. Results

After removing deceased fish and time pings where fish exhibited abnormal behaviours, the data set contained 20 lake trout and 30 Smallmouth Basses, giving 32,954 time pings in total. For each ping, 249 columns represented the target strength at frequencies ranging from 45kHz to 170kHz (excluding 90kHz and 90.5kHz). For 1D CNN, the data set was further split into the train, validation, and test sets with a 70%, 15%, and 15% split respectively. All performances were evaluated based on the test set.

For RNN, the dimension of each input data was 5x249, i.e. target strengths for 5 consecutive time pings across 249 emitted frequencies. The processed data contained 36,729 observations in total. 90% of the data was used for training through cross-validation and the remaining 10% for testing. In the training set, 4,107 time pings originated from lake trout and 2,032 were from the Smallmouth bass. Class weights were assigned accordingly.

For the ResNet, the data was further reduced to balance the number of pings in each class, i.e. lake trout and small white bass. 9,134 pings per class were kept including 6,378 pings per class for training, 1,382 pings per class for validation, and 1,374 pings per class in the test set. The time pings of fish in the three sets were balanced with respect to the species.

3.1 Optimization results

Table 1: Model Architecture after Optimization

Model	**Architecture**	**Notes**
1D CNN	Conv(1, 16, 3, 1), ReLU, Dropout(0.1), BatchNorm, Conv(16, 16, 3, 1), ReLU, Dropout(0.1), BatchNorm, MaxPool(2), Conv(16, 32, 3, 1), ReLU, Dropout(0.1), BatchNorm, Conv(32, 32, 3, 1), ReLU, Dropout(0.1), MaxPool(2), BatchNorm, Conv(32, 64, 3, 1), ReLU, Dropout(0.1), BatchNorm, Flatten, Dense(3648, 2). (With a global L2 regularizations on all parameters at λ = 0.001)	For the convolutional layers Conv(i, o, k, s): i, o, k, s, represent the number of input channels, output channels, kernel size, and stride respectively. For the MaxPool layers, the numbers indicate the kernel size and stride. For each Dropout layer with a value of 0.1, 10% of the input element were dropped out

RNN	LSTM(5*249, 256), LeakyReLU, BatchNorm, Dense(256, 128), LeakyReLU, Dense(128, 2).	For the long-short term memory layer LSTM(i, o) and fully-connected layers Dense(i, o): i represents the input shape, o the output shape. L2 regularization was applied on the dense layer with λ = 0.000001.
ResNet	Block 1, 2: Conv(1, 16, 3, 1), ReLU Block 3, 4, 5: Conv(16, 16, 3, 1), ReLU, Conv(16, 16, 3, 1), ReLU, Flatten, Dense(3970, 2) Skip connections between block 1, 3 and 3, 5.	Notation representations within the architecture are identical with what was outlined above in 1D CNN. The use of shortcut connections (Skip Blocks) in the architecture is to enhance deep network training, so we employed shortcut connections. These connections skip blocks by adding the input of the first block directly to the outputs of subsequent blocks. By combining information across different levels of representation, the model can recognize complex patterns that require the conjunction of high and low level features.

3.2 Model Performances

Table 2: Model Performance Results

Model	**Balanced Testing Accuracy**	**Test AUC**
1D CNN	0.82851	0.88425
RNN	0.80010	0.93466
ResNet	0.89047	0.93131

Fish species classification is a multi-class problem with mutually exclusive labels. Since the data set consisted of two fish species, we naturally evaluated the model based on specificity, sensitivity, balanced testing accuracy, and AUC (area under the curve).

1D CNN

Our final model achieved noticeable performance in all metrics. As shown in Table 2, the model performance was comparable to other alternative methods regarding test accuracy and AUC, while being less stringent on input structure and requiring less computational cost. We also observed that the model performed almost equally well in classifying both species, as the test sensitivity and specificity were very close (see Appx. Table 1). In other words, the model was learning patterns specific to a particular species and could distinguish species well.

RNN

The optimal model obtained consists of one LSTM layer and a single hidden dense layer, with 265 units of output from the LSTM layer and 128 neurons from the dense layer. Furthermore, the L2 regularization parameter was optimized at 1e-6. As outlined in Table 2, the optimized RNN model achieved a relatively higher testing AUC value when comparisons were made. However, it obtained the lowest balanced accuracy among all models. Along with the high sensitivity and relatively lower specificity (see Appx. Table 1), this suggests that the model tends to recognize smallmouth bass (recall that this is the positive class in RNN training) better than lake trout when classifying.

ResNet

Utilizing 1D CNN layers within the residual blocks, without drastically increasing the number of layers, we have achieved an increase in performance evident in both the balanced accuracy and AUC. The addition of skip connections slightly increased the performance. ResNet achieved a relatively lower sensitivity (see Appx. Table 1), suggesting that the model classifies smallmouth bass better than lake trout (recall that ResNet assigned lake trout as the positive class).

3.3 Additional results

1D CNN’s Explainability

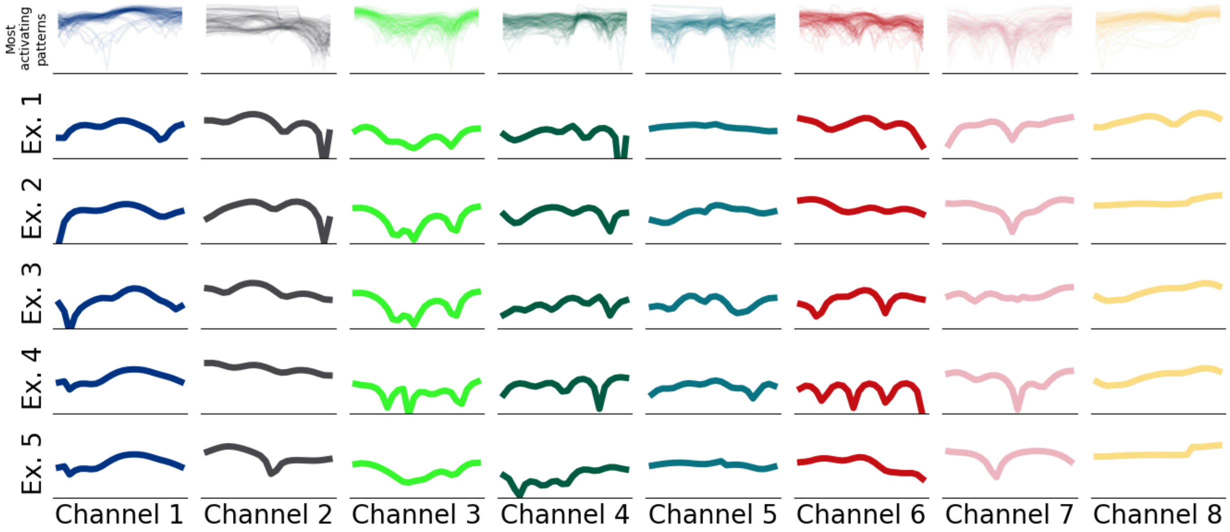


Figure 1. The most activating part of the spectrogram for the first eight channels

In our model, each column is 24 elements wide with a stride of 4, effectively representing an interval of 12kHz in the spectrogram. The output of the final convolution layer has a shape of 64 channels x 57 features. The patterns in Fig.1 illustrate the most activating part of the spectrogram for the first eight channels. For a complete visualization of the most activating patterns for all 64 channels, we refer to Appx. Fig.2. Since our receptive fields have a size of 24 with a stride of 4, each field represents 24 elements in the input vector or a range of 12kHz. As a result, the patterns in Fig. 1 usually contain more than one trough at various depths. Although the example patterns were not identical, we can still observe some general patterns within a channel.

We then combined the most activated patterns with their locations in the spectrogram to further interpret our model. Fig.2 provided heat maps showing the classifier weights when predicting both species; each neuron in the classifier corresponds to different channels and locations in the spectrogram. A higher-weight neuron is colored yellow, whereas a lower-weight neuron is colored purple. For Lake Trout, although the target strengths are noisy, we could still identify frequencies from 45kHz to 101kHz to carry salient weights in classification. As for the Smallmouth Bass, frequencies from 45kHz to 75kHz, from 83kHz to 101kHz, and from 109kHz to 123kHz are the most important regions for classification. These heat maps could aid us in understanding what sound wave frequencies are characteristic of certain fish species.

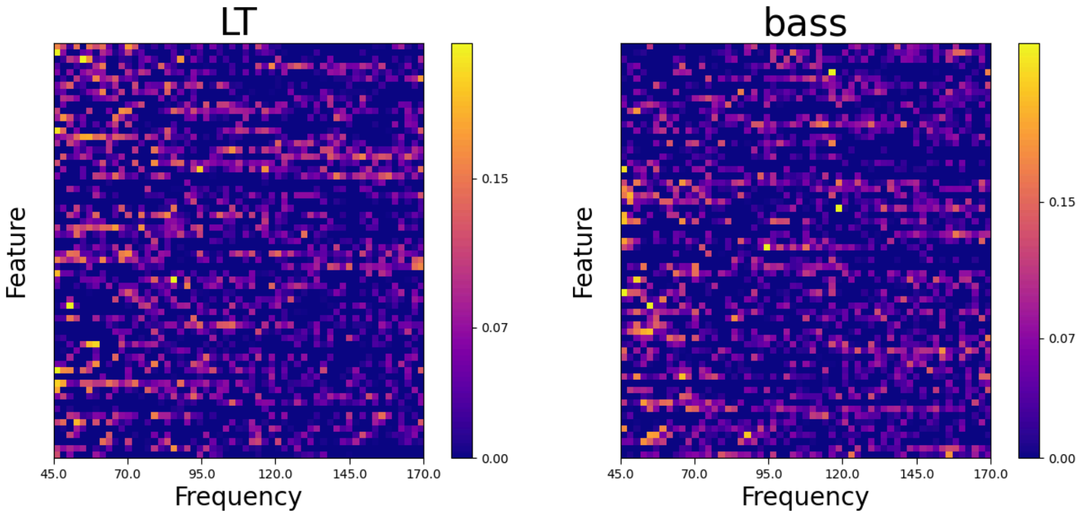


Figure 2. Saliency in classifier weights for each fish species

By locating the high-weight neurons and their corresponding channels, we can combine the information from Fig.1 and Fig.2 to provide more insights into the segments of the spectrogram our model focused on. For instance, patterns (channels 1, 2, 14, 36, 51, 53, 54) that are close to a nearly horizontal line with few shallow troughs activate the network the most at frequencies from 45kHz to 57kHz when predicting Lake Trout. As for Smallmouth Bass, the network would be activated the most when it detects a similar pattern as for Lake Trout (channels 20, 24, 31) plus the general pattern of one deep trough (channels 23, 27, 28, 39) at frequencies from 45kHz to 57kHz. The discrepancy in the most activating patterns of species allows us to have more insights into how species respond differently to sound waves of various frequencies.

4. Discussion

The hydroacoustic data was collected using emerging technologies, in which little research has been done on performing and optimizing classification tasks in a compatible way. Therefore, our task was to investigate various algorithms that are potentially compatible with our data and to identify the best-performing ones. We examined three models — 1D CNN, RNN, and ResNet, to perform binary classification on fish species.

Beginning with 1D CNN, the model could identify common activation patterns and the salient range of frequencies that are more distinctive for each fish species. Model training had a lower demand on computation power compared to the other two methods. In terms of general application, this may be useful in further hydroacoustic measurements for fish population sampling, as the data collection can focus on a smaller range of frequencies, reducing expense by using fewer transducers and saving time spent on data pre-processing. We can also look for specific patterns in the raw data for an initial understanding of what fishes are being sampled, for example, frequencies range from 45kHz to 123kHz for both species based on our results. The salient patterns of activations and frequency ranges may be more informative if we obtain data for more fish species and perform classification in future. However, 1D CNN also has the limitations of not capturing the hierarchical structures for the input data since the input array is based on individual observations (pings).

In contrast, RNN could preserve information from each individual with multiple pings being considered collectively in each input to the model. RNN allows the retention of information from prior inputs, allowing further analysis of the variance in predictive accuracy within and between fish regions. Limitations of RNN include it being computationally expensive, demanding much more computational capacity than 1D CNN. Moreover, direct outputs from training RNN lacked interpretability when compared to alternative models. With the above being said, although its interpretability is relatively lower, RNN is the most accurate at performing what we intend to do — classifying fish species using frequency response from individual fishes. In applications of species classification, it is rare that individual time pings from the same fish are inputted singly for training. We wish to maintain all information held in the frequency data for one individual rather than treating each time point as a distinct fish.

ResNet is another powerful model that can be used for our classification task. It has two factors in mind. One is that we expected the model complexity to cause the vanishing gradient problem, the other is that we can utilize skip connections to force our network to take both low-level features extracted in early layers and high-level features present in recent layers into consideration. As for its general application, ResNet can be highly beneficial during the training process. It can capture the hierarchical structures in the input data, which could potentially lead to more accurate classification results. This is especially useful when the input data is not based on individual observations, but rather on a sequence of observations where the order and relationship between the observations matter. However, it's important to note that ResNet requires more computational power compared to 1D CNN. Therefore, the use of ResNet might increase the cost and time of data processing.

In conclusion, each model has its strengths and weaknesses, and the choice of model depends on the specific requirements of the task, including computational resources, the nature of the input data, and the desired outcomes. Further studies and experimentation with more fish species could provide more informative patterns of activations and frequency ranges. We intend to investigate more complex architectures such as RNN with Residual Blocks, Convolutional Recurrent Neural Networks (CRNN) and other advanced architectures and techniques. We hope that these techniques can further improve the classification performance and provide more insights into distinguishing between fish species.

References

Ali, M. (2023, November 9). *Introduction to activation functions in neural networks*. DataCamp. https://www.datacamp.com/tutorial/introduction-to-activation-functions-in-neural-networks

Boufeloussen, O. (2022, December 22). *Simple explanation of recurrent neural network (RNN)*. Medium. https://medium.com/swlh/simple-explanation-of-recurrent-neural-network-rnn-1285749cc363

Baughman, D., & Liu, Y. (1995). Fundamental and Practical Aspects of Neural Computing. *Neural Networks in Bioprocessing and Chemical Engineering,* 21-109. https://doi.org/10.1016/B978-0-12-083030-5.50008-4

Chauhan, N. S. (2022, June 7). *Introduction to RNN and LSTM*. The AI Dream. https://www.theaidream.com/post/introduction-to-rnn-and-lstm

Gugele, S. M., Widmer, M., Baer, J., DeWeber, J. T., Balk, H., & Brinker, A. (2021). Differentiation of two swim bladdered fish species using next generation wideband hydroacoustics. *Scientific Reports*, *11*(1), 10520. https://doi.org/10.1038/s41598-021-89941-7

Gupta, G., Kshirsagar, M., Zhong, M., Gholami, S., & Ferres, J. L. (2021). Comparing recurrent convolutional neural networks for large scale bird species classification. *Scientific Reports*, *11*(1), Article 1. https://doi.org/10.1038/s41598-021-96446-w

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385. Retrieved from https://arxiv.org/pdf/1512.03385.pdf

Ji, F., Ni, J., Li, G., Liu, L., & Wang, Y. (2023). Underwater Acoustic Target Recognition Based on Deep Residual Attention Convolutional Neural Network. *Journal of Marine Science and Engineering, 11*(8). doi:10.3390/jmse11081626

Rieger, L. H., Wilson, M., Vegge, T., & Flores, E. (2023). Understanding the patterns that neural networks learn from chemical spectra. *Digital Discovery*, *2*(6), 1957–1968. https://doi.org/10.1039/D3DD00203A

Wang, C. (2021, December 7). *The vanishing gradient problem: The problem, its causes, its significance, and its solutions.* Medium. https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

Appendix

A.1 Tables

Table 1: Evaluations of model performance

Model	**Balanced Testing Accuracy**	**Test AUC**	**Sensitivity**	**Specificity**
1D CNN	0.82851	0.88425	0.84941	0.80761
RNN	0.80010	0.93466	0.95018	0.64998
Resnet	0.89047	0.93131	0.8261	0.9549

A.2 Figures

Figure 1: Visualization outlining the processing of input structure specific to RNN

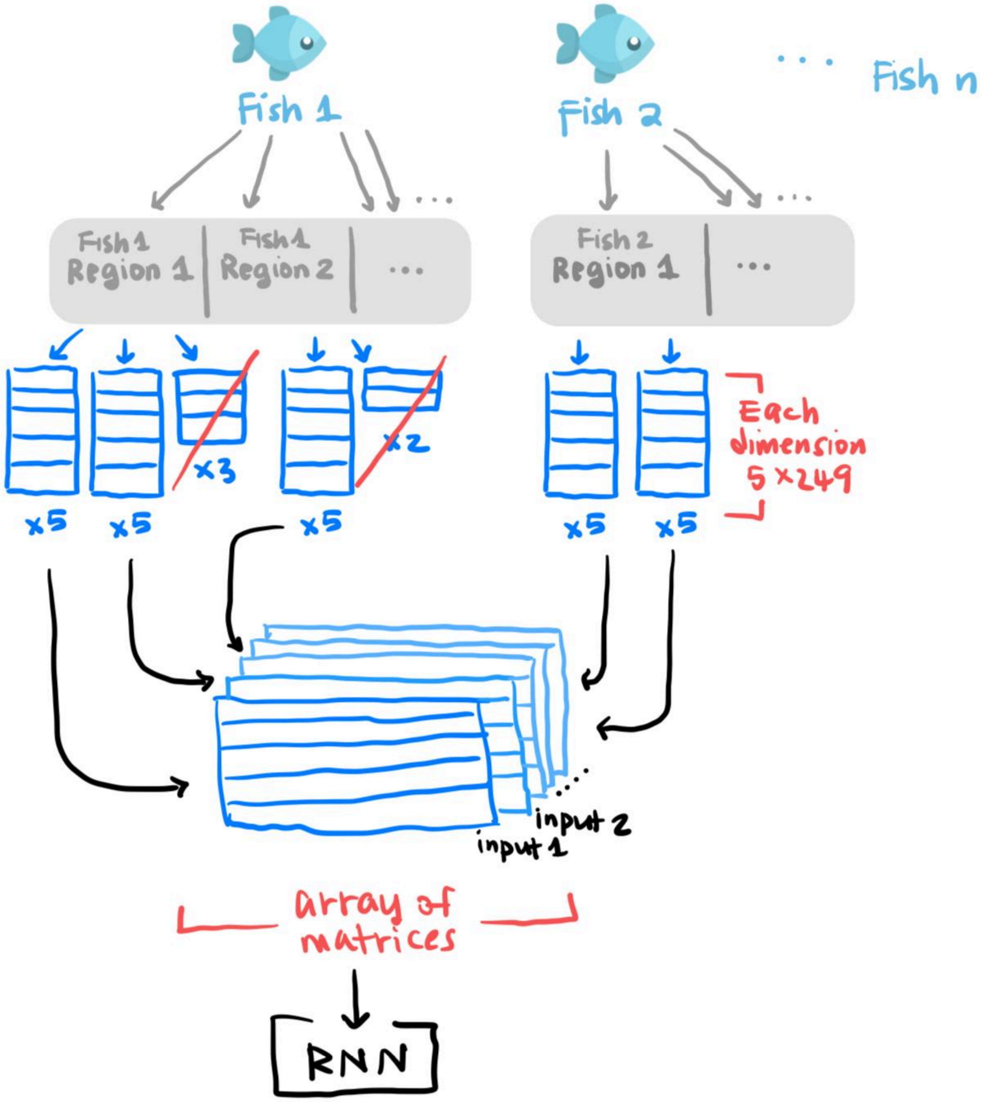
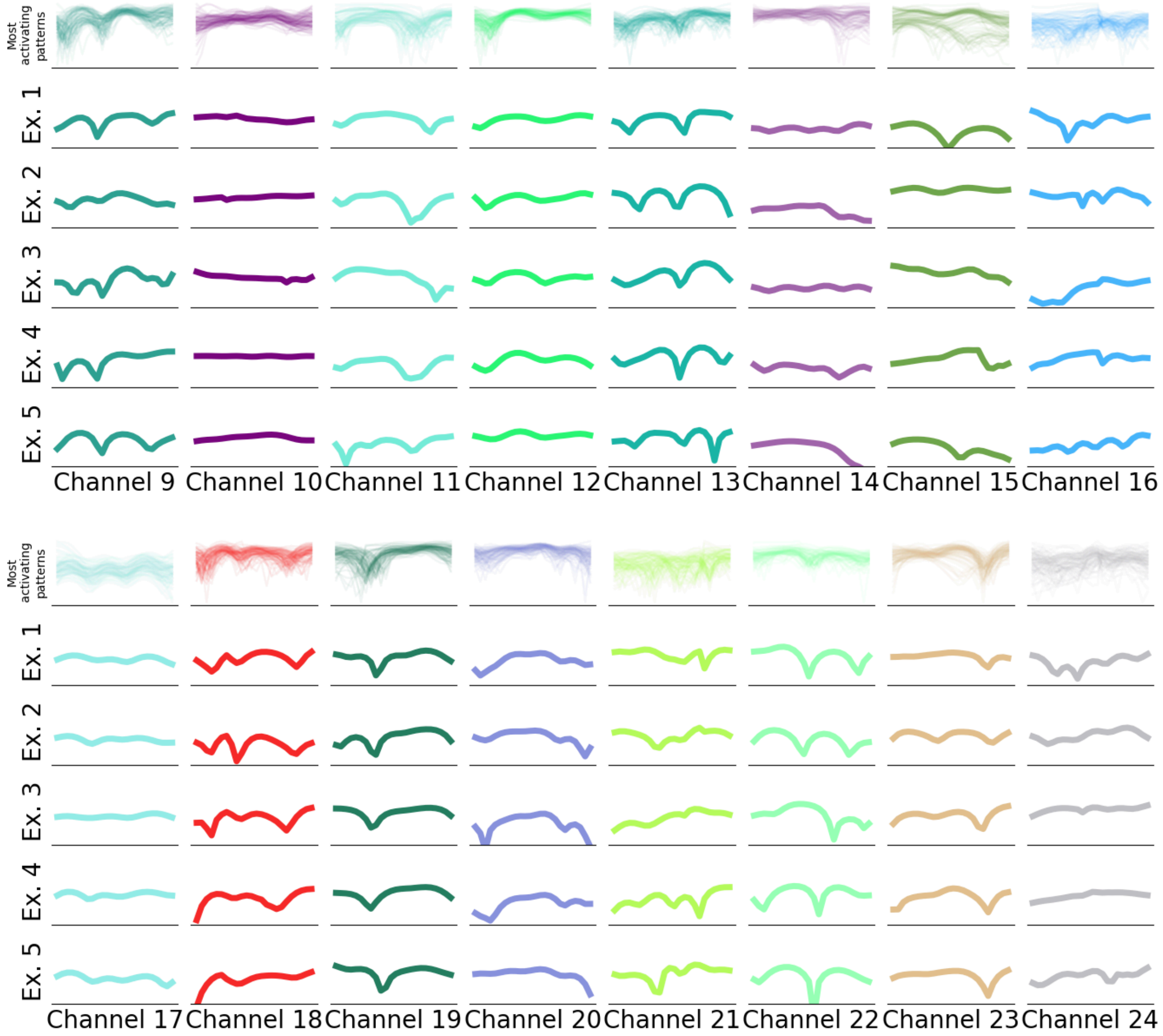
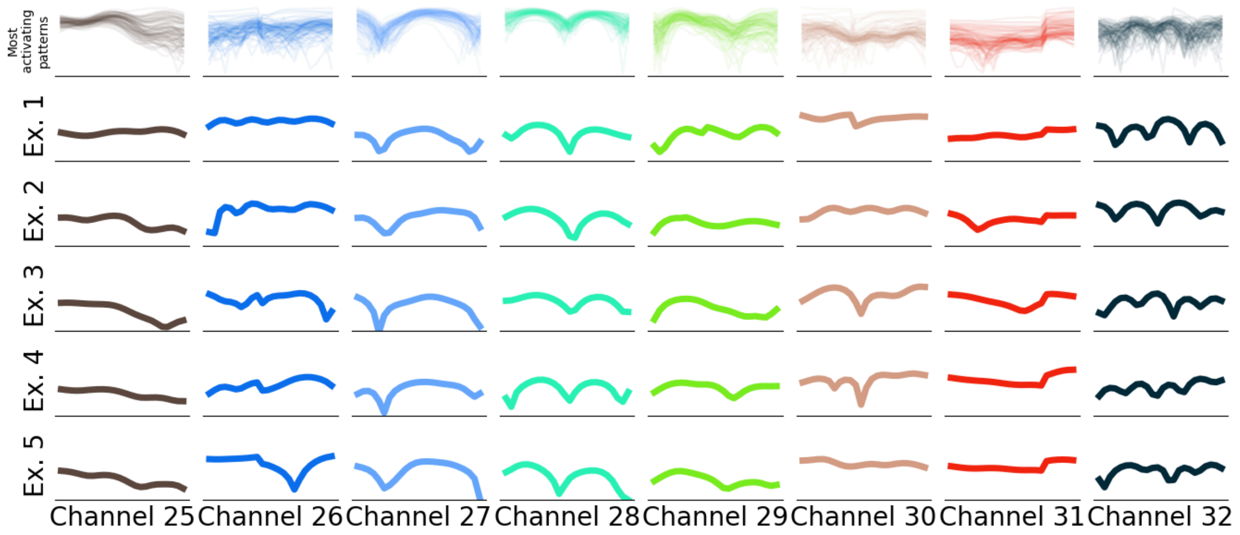
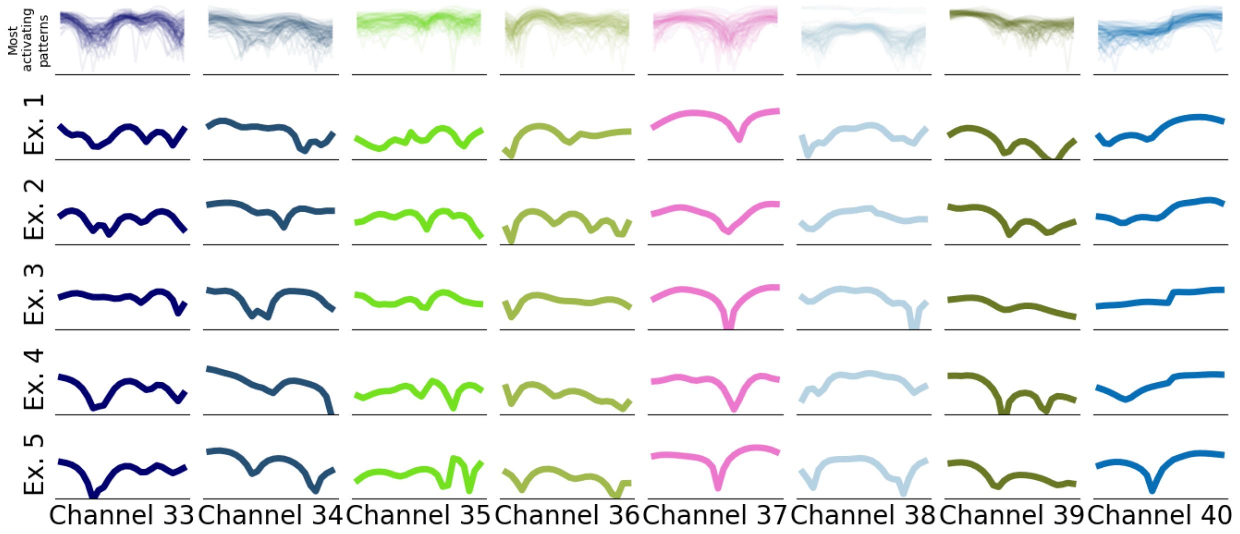
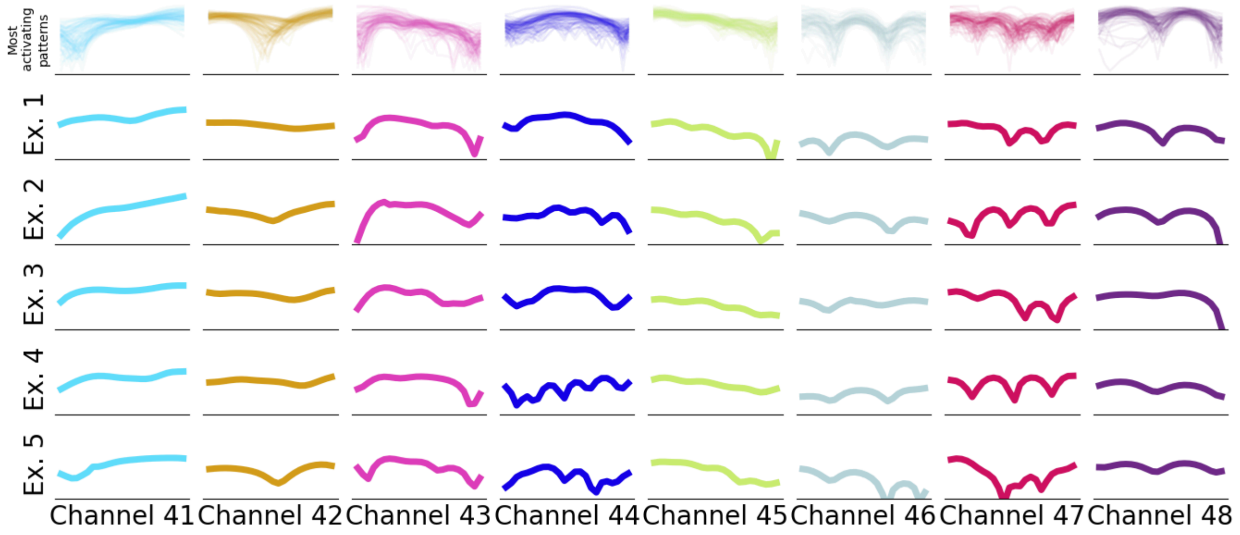


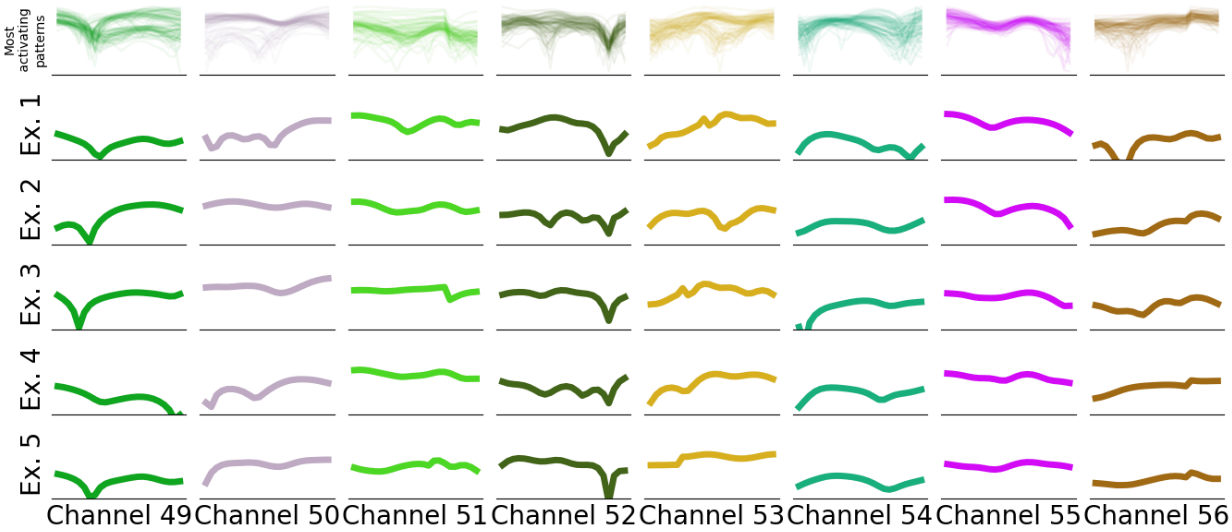
Figure 2: Visualization of the most activating patterns for all 64 channels from the final convolutional layer of the 1D CNN

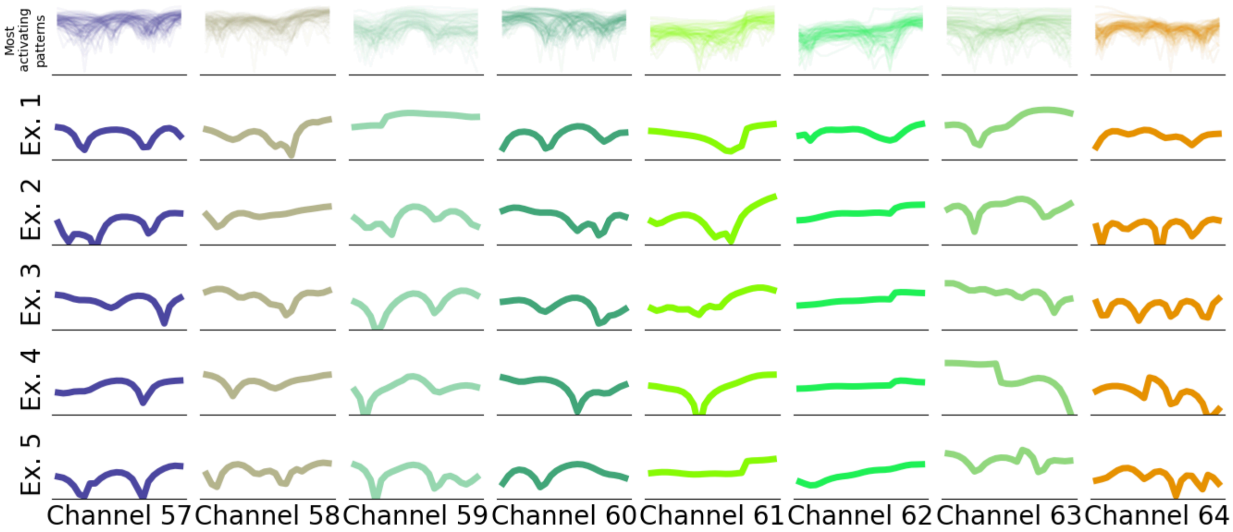



Most activating patterns
Ex. 1
Ex. 2
Ex. 3
Ex. 4
Ex. 5
Channel 25
Channel 26
Channel 27
Channel 28
Channel 29
Channel 30
Channel 31
Channel 32


Most activating patterns
Ex. 1
Ex. 2
Ex. 3
Ex. 4
Ex. 5
Channel 33
Channel 34
Channel 35
Channel 36
Channel 37
Channel 38
Channel 39
Channel 40


Most activating patterns
Ex. 1
Ex. 2
Ex. 3
Ex. 4
Ex. 5
Channel 41
Channel 42
Channel 43
Channel 44
Channel 45
Channel 46
Channel 47
Channel 48


Most activating patterns
Ex. 1
Ex. 2
Ex. 3
Ex. 4
Ex. 5
Channel 49
Channel 50
Channel 51
Channel 52
Channel 53
Channel 54
Channel 55
Channel 56


Most activating patterns
Ex. 1
Ex. 2
Ex. 3
Ex. 4
Ex. 5
Channel 57
Channel 58
Channel 59
Channel 60
Channel 61
Channel 62
Channel 63
Channel 64