Analysis of Shape Classification Techniques in Hydrodynamic Imaging with an Artificial Lateral Line

A. Brandsma
Supervisors: dr. S.M. van Netten & dr. B.J. Wolf

Abstract— **The lateral line is a unique fluid flow sensing organ found in fish. It has been used as a model to create a sensor array called an Artificial Lateral Line (ALL). This ALL can be used in hydrodynamic imaging for the classification of shapes, velocities or vibrations for objects moving under water. The ALL could be a replacement for situations where other sensors, like sonar or radar, might not function properly. Both the sensor and the classification technique are relatively novel and have promise for real world applications. This research focuses on analyzing the performance of two feature extraction techniques on two different flow datasets that have been created by an ALL. The two different feature extraction methods are used to retrieve computationally efficient features from the two flow datasets; we use a hand-picked feature extraction and an auto-encoder that automatically tries to detect which features to extract. These extracted features are then used for classifying the specific shapes using a classification algorithm called the Extreme Learning Machine. We also compare these feature extraction techniques with a raw time series input. The difference of the two flow datasets are characterized by two properties; the size of the sensor array and the size of the water volume in which the sensor array was placed. We found that the difference between the resulting classification scores of the two flow datasets was small on a per-run basis. However, there was a significant difference between the classification scores of individual windows between the two flow datasets. The auto-encoder used in this research resulted in non-generalizable features. These features resulted in the Extreme Learning Machine having a high classification score on the training set, whereas the test set had a very low classification score. The feature extraction methods mentioned in this paper are all promising and could lead to a fundamental enhancement of human senses in underwater environments.**

Keywords: *Artificial Lateral Line, Hydrodynamic Imaging, Extreme Learning Machine, Feature Extraction, Auto-encoder, Shape Recognition*

I. Introduction

Fish can make use of a lateral line organ that enhances their senses. This organ gives them sensory input from the water pressure around them [1]. Fish that are blind rely on the combination of this lateral line organ and touch in order to scope out their surroundings. In essence, this organ provides a 'touch at a distance' sense. Using the hydrodynamic wakes that are being measured by this organ, the fish is able to detect the local environment and react accordingly.

When an object is moving through a fluid, it leaves characteristic wakes that encode properties like shape, location, velocity and vibration of an object [10] [36]. These hydrodynamic wakes can be identified through the use of machine learning techniques in order to locate, track and classify the object.

Artificial Lateral Lines (ALLs) are a type of sensor array that can be used to mimic the function of the lateral line organ. Using a series of measurements, we can sample and analyze a spatiotemporal flow pattern from the hydrodynamic wake. This pattern can then be used to make predictions about the sources that created the flow pattern.

Once a classification technique has properly learned the representation of different characteristic wakes, it can then be used for object detection and collision avoidance in autonomous underwater vehicles (AUV) in murky water or with sonar blindspots. Given fast and reliable signal processing, which this research paper aims to show, this sensor and classification combination could, for instance, be a very viable approach to sense the surroundings of an AUV.

Two distinct feature extraction methods were tested in this thesis. For one, features were hand-picked and extracted manually [36]. For the other, we used an auto-encoder to automatically determine which features need to be extracted and extract those. This research, among other things, aims to validate the hand-picked feature extraction technique employed by Wolf et al. in a previous research [36].

The ALL dataset of Wolf et al., along with the ALL dataset used in another previous research [32], are used as input for the feature extraction methods. These two datasets have a few differences, among which the size of the area in which it was tested, the size of the sensor array itself and the shapes that were classified. For the remainder of this paper we will refer to the two flow datasets as the 'aquarium' dataset, for the dataset of [32], and the 'swimming pool' dataset, for the dataset of [36].

The features that are extracted from the two ALL datasets are passed through an Extreme Learning Machine (ELM). The Extreme Learning Machine is a type of machine learning algorithm that can be used for classification [14]. It requires only one training step in order to get a high classification score. It has also been shown to work well with ALL-generated data [33] [27] [34]. In order to compare the efficiency of the feature extraction methods we also use raw data input as done in [32]. Raw data means that the time series of water flow patterns are used without any feature extraction.

We discuss the viability of such an approach in real-world situations and try to come up with explanations for why the flow datasets might differ in their classification score when put through the same kind of classification technique. We also perform a parameter sweep for the feature extraction methods in order to get the most optimal results in terms of

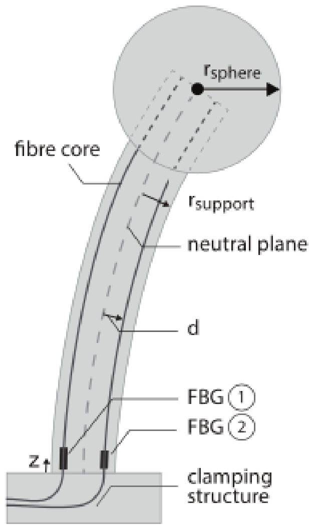


Fig. 1: Schematic of the fluid flow sensor. Fiber Bragg Gratings (FBG) are used to measure the deflection of the sphere based on reflecting certain light waves. [31]

classification score.

In short, this thesis aims to be an explorative analysis that looks at all facets of the two ALL datasets, the two feature extraction techniques mentioned above and the machine learning technique used. Our research question is two-fold:

- Will feature extraction methods outperform raw time series as an input for Extreme Learning Machines?
- Will a smaller sensor array, smaller surroundings and smaller objects have a significant positive effect on the classification results of the Extreme Learning Machine?

Our hypothesis is that the feature extraction method will indeed outperform the raw data as features are computationally more efficient for the ELM [3] [5]. We also expect the smaller sensor array/surroundings/objects to have a positive effect on the classification scores. The latter effect is expected for two reasons. Either the object moving through the water might create less turbulence in the surrounding fluid, or the sensor array registers less flow data when the objects moves in between the individual sensors of the larger sensor array.

II. Background

Hydrodynamic imaging [11] usually refers to measuring a projection of the hydrodynamic environment. Measuring a projection of the surrounding environment can be done using sensor arrays of different types [28]. We can use this projection for detecting properties of objects moving through the surrounding body of water, since the moving objects create a flow field that encode useful properties. The following subsections give information about the sensor array, the datasets used in this research, the machine learning techniques and the feature extraction methods.

A. Lateral Line

In order to sense the surroundings of a fish, the fish has a so-called lateral line organ which consists of multiple distributed neuromasts. A neuromast is a type of sensor that can sense the pressure gradient of the water that surrounds the fish. Multiple neuromasts are used to receive sensory input from all sides of the fish and create what is known as a ‘flow pattern‘. A flow pattern is defined as a single time step from multiple sensors that are combined. The cupula of a neuromast is a gelatinous structure that transmits the hydrodynamic forces produced by the surrounding flow of water to the connected hair bundles [20]. The hair bundles then generate a nervous response [10]. It has been shown that fish can use these neuromasts to detect dynamic and static objects around them [12][6]. Not only can they detect the existence of such an object, it is also possible to identify the location, vibration and shape of an object [33][8][10]. In the case of a static object, the pressure waves are generated by the movement of the fish or by turbulent surroundings. Dynamic objects on the other hand will generate measurable flow fields themselves.

A singular neuromast is enough to detect the surrounding changes in pressure gradients. Therefore, each individual neuromast can be seen as a full-fledged detector of ‘temporal patterns‘. Temporal patterns are defined here as multiple time steps from one sensor that are concatenated together. The fact that one sensor is enough to detect such patterns is shown as there is little to no mechanical coupling between the neuromasts in fish [10].

The combination of the flow pattern and the temporal pattern is called a spatiotemporal pattern. This spatiotemporal pattern is a representation of what is used by a fish to enhance its own perception of the surrounding waters.

B. Fluid Flow Sensor

In 2002, Fan et al. showed an initial implementation of an Artificial Lateral Line (ALL) [7]. Several implementations of ALLs have been created since. This thesis uses an ALL that was constructed by the LAkHsMI consortium[1]. This ALL has been inspired by the canal neuromasts found in fish [31]. The ALL consists of an array of deflection sensitive sensors placed along a straight line. This deflection is interpreted as fluid speed. The sensors used here are therefore a type of fluid speed sensor, also called an artificial neuromast.

The sensors from the ALL measure deflection through means of an optical signal. These optical signals are measured using Fibre Bragg Gratings and can be sent through optical lines to be measured at a different location. This makes the sensor especially useful in situations where it is preferable not to have a need for electricity at the location of the sensor.

The sensor design consists of the following: a fluid force recipient spherical body and a fiber support structure providing elastic coupling to a clamping structure. Inside of the fibre support structure there are Fiber Bragg Gratings (FBG)

[1] https://www.lakhsmi.eu/

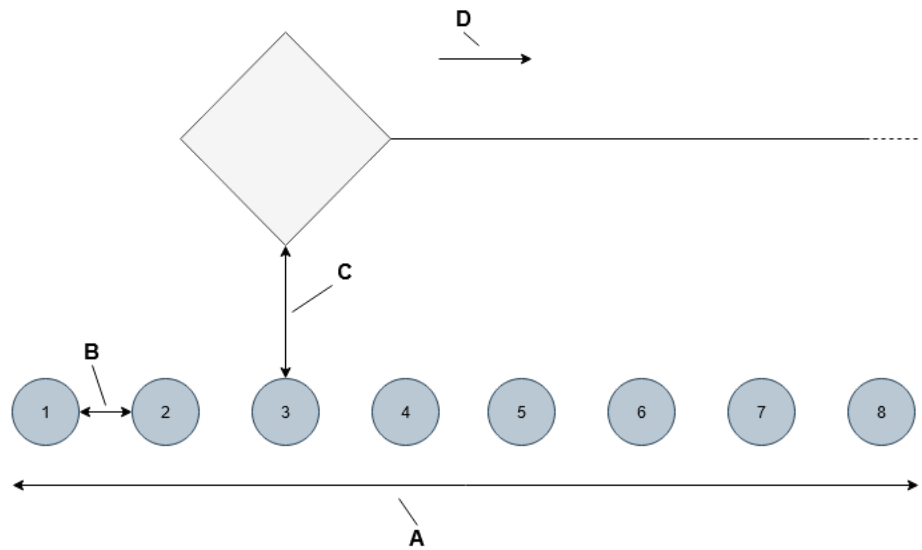


Fig. 2: A schematic overview of the sensor layout. The sensor layout was similar for the measurements of both flow datasets. An object, in this case a turned cube, was dragged past the sensor array. Each length and speed is named for the swimming pool and the aquarium respectively. A had a length of 3.5 m and 500 mm. B had a length of 0.5 m and 64 mm. C had a length of ~ 0.62 m and ~ 70 mm. D had a speed of 0.3 m/s and 127 mm/s.

which reflect light differently depending on the amount of strain put on the spherical body. This reflection of light can be measured to determine the forces on the sphere and thereby the velocity and direction of the surrounding fluid. A schematic of the sensor can be seen in Figure 1.

C. Extreme Learning Machine

An Extreme Learning Machine (ELM) is used for classification of the ALL data. The Extreme Learning Machine is a feed-forward neural network which only needs one learning step in order to reliably give high classification scores [14]. It has been shown that this type of machine learning model generally outperforms support vector machines in their classification capabilities and/or their training time [21]. Given that these models can provide a high classification score, while also providing a short learning time, gives enough reason to pick this model as an interesting machine learning candidate to study. In addition, it has also been shown that the ELM is fast and reliable in the field of hydrodynamic imaging with ALLs [33] [34]. Models such as multilayer perceptrons (MLP) or the echo state network (ESN) might outperform the ELM in classification score, but these systems take longer to train [27].

Perhaps a more important benefit for the ELM is its inherent ability to prevent overfitting due to the low number of hyper parameters of the system. Only the number of hidden neurons need to be decided upon. This makes it very straightforward to tune the neural network. The number of hidden neurons can be decided by trying several permutation of that hyper parameter and determining the classification scores of each of these ELM permutations. This poses no real problem due to the quick training time of the ELM.

D. Hand-picked Feature Extraction

Traditionally features are carefully selected based on their expected utility. Feature selection is done in order to remove redundancies in data, increase efficiency in learning tasks and improve our understanding of the learned results of a machine learning algorithm [3] [5].

For the purposes of this research, we specifically take a look at the feature selection in hydrodynamic environments. The features in the hand-picked feature extraction technique are based on earlier promising results in river flow conditions [29].

The hand-picked features consisted of direct current (mm/s), frequency bands (Hz), kurtosis and skewness. 16 frequency bands were selected from the ANSI half-octave band definition [2], using 0.25 Hz as a reference frequency [36]. The kurtosis and skewness describe the distribution of data in a time window and are added to the feature set as they have been used in previous fluid flow classifications [29] and sensor placement optimization [35].

Wolf et al. tested several subsets of the hand-picked feature sets. Using all frequency bands seemed to better help in classification than some traditional feature extraction techniques, like Lasso or FCBF [36].

E. Auto-encoder

Auto-encoders are neural networks that have the goal to recreate their input in their output with an intermediate step. This intermediate step is intended to create computationally efficient features from the input [13]. This type of neural network usually consists of one input layer, one hidden layer and one output layer. There do exist multilayer auto-encoders as well and stacking of auto-encoders is possible, which also essentially increases the number of hidden layers [16] [18].

Fig. 3: Schematic view (to scale) of the shapes used in [36].

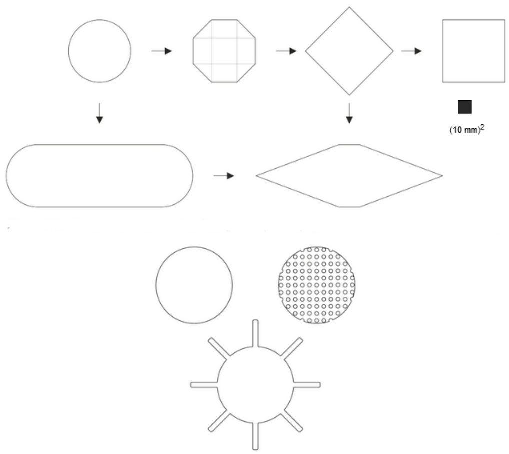


Fig. 4: Schematic view of shapes with different topology, elongation and surface types. From left to right the topology changes from the sphere to the square. From top to bottom the elongation changes. The objects with different surface type are shown separately. These shapes are the same as used in [32]

The hidden layer of an auto-encoder is usually restrained to a smaller dimension than the input in order to get features that have useful, discernable properties. Auto-encoders with hidden layers that are smaller than the input are called undercomplete. By training the undercomplete auto-encoder, it is forced to generate useful features, otherwise it would not be able to recreate the original input.

The loss function for the performance of the auto-encoder during training can be formulated as follows:

$$E = \underbrace{\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}(x_{kn}-\hat{x}_{kn})^2}_{\text{mean squared error}} + \lambda * \underbrace{\Omega_{\text{weights}}}_{L_2 regularization} + \beta * \underbrace{\Omega_{\text{sparsity}}}_{\text{sparsity regularization}}$$

Where λ is the coefficient for the L2 regularization term and β is the coefficient for the sparsity regularization term. Both terms are used as additional constraints on the output. These coefficients can be set to any value, but are usually selected to be low values. Besides these factors we can also tweak the sparsity proportion. This is the *desired* proportion of training examples a neuron reacts to. A low value for the sparsity proportion usually leads to each neuron in the hidden layer only giving a high output for a small number of training examples, which means a higher level of sparsity.

F. Datasets

Two flow datasets were used in this thesis. These datasets are derived from earlier experiments and will be briefly discussed here. While the scale of the environment, the sensor array and the objects is different for both experiments, the schematic overview is *roughly* the same. This schematic overview can be seen in Figure 2.

1) Swimming Pool Dataset: The swimming pool dataset consists of hydrodynamic flow data created by a large ALL in a relatively large body of water with large objects. It was obtained in the experiment by Wolf et al. [36].

Five different shapes were used in the swimming pool experiment. Also, a source was added where no object, only the towing platform, was pulled alongside the sensor array. The dumbbell and the barrel were chosen as they displace roughly the same amount of water, but have different topologies. The sphere was chosen since it is generally used in a lot of ALL-related studies [25]. And finally, there were two variants of capsules; one is a slender capsule, whereas the other is the same kind of capsule and has an inverted bowl attached to the end of it. A schematic view of these shapes can be seen in Figure 3.

The setup was built on the short side of a swimming pool with a size of 18 *m x* 25 *m*. The sensor array had a length of 3.5 *m* and was centered along the short side of the swimming pool. Each sensor of the sensor array was placed 0.5 *m* apart from each other.

Each object was towed six times past the sensor array. Three times in a forward motion and three times in a backward motion. A full motion of the towing platform took about 50 seconds. From these 50 seconds, 20 seconds were selected where the object was moving right in front of the sensor array.

2) Aquarium Dataset: The aquarium dataset consists of hydrodynamic flow data created by a small ALL in a relatively small body of water with small objects. It was obtained in the experiment by Römer [32].

Eight different shapes were chosen that differed in three different properties: topology, elongation and surface. Each changed property was expected to change its hydrodynamic signature. The topology differences have been defined by their extremes, namely the sphere and the cube. While in the middle of the spectrum a rhombicuboctahedron ('rhombus' for short) was defined. This could change the sensed signals due to sharper edges causing stronger vortices. The change of the elongation caused the object to pass the sensor for a longer time. Since the signal will prolong longer for the sensor it will most likely have a different hydrodynamic signature. Changing the surface texture has been inspired by the golfball. A golfball has little dents all over its surface which reduces the drag. The hypothesis was that a similar effect might occur underwater. A schematic overview of all different shapes has been provided (Figure 4). There is a change in topology from left to right and a change in elongation from top to bottom. The different surface types have been shown individually.

The ALL setup that was built for this experiment consisted of a water tank of $1200*800*260$ *mm* $(w*l*h)$ with an extended Makeblock plotter $(1040*1000)$ on top that could span across a large area of the water tank. The sensor array in this experiment had a total width of 500 *mm* and each sensor was placed 64 *mm* apart from each other. The sensor array was centered along the width of the water tank.

The second sensor in the array became faulty during testing and was therefore unreliable. Römer only used 5 sensors in his experiment. This was due to the faulty sensor and also because the first and last sensors seemed to pick up some unwanted distortions that most likely occurred due to the ramp-up and ramp-down phases of motion.

The dataset consists of 9 sources (i.e. the shapes) that have been dragged past the sensor array at 2 different distances. Each distance has been tested 20 times in both a forward and backward direction resulting in 80 different measurements per source. Note that the spiky sphere was not used at the closest distance due to the length of the spikes hitting the sensor array otherwise.

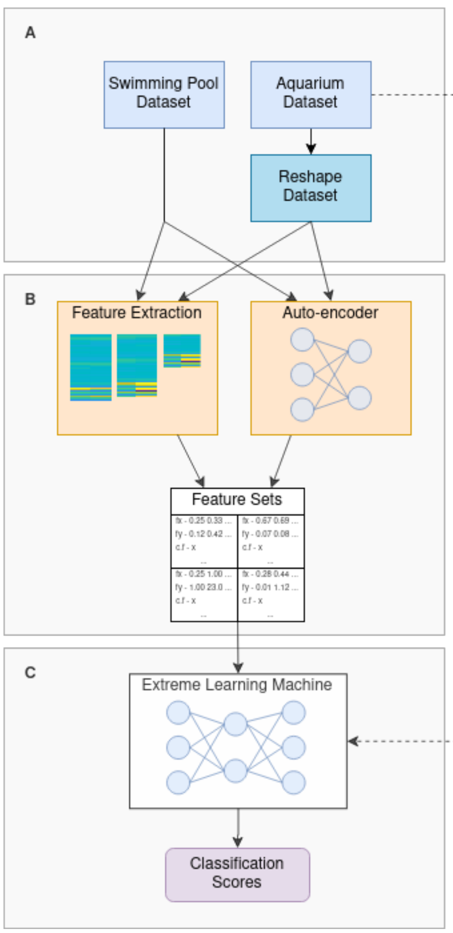


Fig. 5: A schematic overview of the steps taken in the experiment. Panel *A* shows the two datasets that are passed into the feature extraction methods. The aquarium dataset is also passed as raw data directly to the classifier (represented by the dotted line), which was done in [32]. Panel *B* shows the two feature extraction methods resulting in all the feature sets. Each dataset and feature extraction method has its own feature set. Panel *C* shows the classifier and the resulting classification score. Each input has its own respective classification score.

III. Methods

The goal of this thesis is to determine the efficiency and viability of several different feature extraction techniques on ALL-generated data. In the next several subsections, we provide details for both the feature extraction methods, the machine learning classifier, and how the results will be analysed. See also Figure 5, for an overview of the classification pipeline.

A. Data Transformation

In order to process the data in a similar fashion as the swimming pool dataset [36], we needed to reshape the aquarium flow data to the same form as the swimming pool data.

For each object in the aquarium dataset, 80 movements were recorded. Each movement consisted of approximately 1250 time samples, sampled at 250 Hz, for both an X and Y deflection in each array. Each movement was represented in a [1250, 16]-dimensional struct, where the first term represents the time samples and the second term represents the concatenation of the X and Y deflection of eight sensors. One sensor was removed since it outputted unreliable data.

The data from the swimming pool dataset took the data in a [1200, 8, 2]-dimensional struct, where the first element is representing the time samples, the second is representing the sensors and the third is representing the X and Y deflections. The X and Y deflection had to be split from each other for each sensor of the aquarium dataset.

B. Feature Extraction Methods

Two feature extraction methods were used in this experiment. The first method carefully hand-picks the features from the datasets (see *background*). The second method, explained here, tries to automatically determine computationally efficient features and extracts them. Both methods make use of a moving window approach.

1) Moving window approach: A time window of 4 seconds with a stride of 0.5 seconds and a downsampled sampling rate of 200 Hz was used. These values were found using cross-validation optimization. This also agrees with the theory that characteristic wakes and hydrodynamic stimuli are in the < 5 Hz range [36], as windows with a sampling rate of 200 Hz capture that frequency range well.

2) Auto-encoder: A simple auto-encoder consisting of three layers was used for this experiment. This means there is only one hidden layer in the auto-encoder. The auto-encoder was regularized to be a sparse variant. The sparse variant is encouraged, through means of an added penalty, to recreate the input data in a more constrained way than simply mimicking an identity function. This type of auto-encoder is usually used for classification, as seen in several researches across domains [17] [26] [23]. The hidden layer size was set to 250. There is no particular reason why the number of features should be the same as the hand-picked feature method. We do not want too large of a step to reduce the dimensions of the input data, therefore we have chosen to create more features.

The performance of the auto-encoder was tested during training using a mean squared error. This performance was then regularized using an L2 regularization of 0.01, a sparsity regularization of 4 and a sparsity proportion of 0.10. The training was performed using a scaled conjugate gradient. The loss function for the performance is described in the *Background* section.

C. Parameter Sweep

Several parameters were tweaked in order to manipulate the feature extraction methods. The different parameter values were tested on the hand-picked feature extraction method. The best parameter settings were then applied to both feature extraction methods for the final classification results. We will briefly discuss the parameters that were changed and why.

First of all, noise has been added to the dataset. The addition of noise lowers the chance of overfitting for the system, and therefore increases the generalizability [4] [24]. These beneficial aspects are especially the case for small datasets where the neural network has no choice but to overtrain on each example.

A subset of the time range was selected from which the features were extracted. Omitting certain parts of the dataset results in the classifier finding more discernable differences as some parts of the data are too similar.

Next the removal of certain sensors was considered. Sensor 2 had to be removed since it became unreliable during testing. The outer sensors were also considered for removal (see *background*).

The selection of proper window properties were also considered since these properties can have a significant effect on the resulting features. These window properties were the stride, the resample frequency and the window length. We know that characteristic wakes are usually found in the < 5 Hz range [36]. The window properties were adjusted in order to best capture the wakes in this frequency range.

We also applied normalization to the dataset. Normalization is not necessarily a parameter since it was always applied. The features that are created are therefore within a range of -1 to 1. There are a variety of practical reasons why standardized inputs are preferred in some applications of machine learning, among which its possibility to make training faster and it also reduces the chance to get stuck in local optima [15] [9].

Finally, scaling of the feature set is used to change the features to have larger or smaller values. Note that this is applied to the entire feature set at once, so there is no real relative change between features. The scaling here does not refer to 'standardization', as there is no need to change the unit of measurement for individual data points.

D. Classifier Training

The classifier used in this research is called an Extreme Learning Machine. The Extreme Learning Machine is a type of feed-forward neural network and only requires on hyper parameter to be set. This is discussed along with the input and output of the algorithm.

1) Hidden Layer Size Selection: An important hyper parameter for ELMs is the number of hidden neurons. We use a nested cross-validation optimization technique in order to determine the hidden layer size that has been optimized for classification score and overfitting reduction. The size of the hidden layers were generally around 200-300 neurons, but could change depending on the number of features that were used as input.

2) Input/output: The input of the ELM were the features that have been extracted in several different ways. The input layer, therefore had a size appropriate to the number of features, which was either 48 or 250 depending on which feature extraction technique was used. The output layer had a size equal to the number of different classes the machine learning algorithm could choose from. In this case, that were the object shapes producing each of the two flow datasets. This means the output layer had 6 and 9 different possible classifications across both flow datasets.

E. Data Analysis

1) Classifier Performance: The performance of the classifier has been measured by the F1-scores. This score considers both the precision and the recall of the test. An F1 score has been given for both the complete run and for each individual window. We consider a 'run' to be a completed motion, whereas a 'window' consists of parts of a complete motion. Since we use a k-fold cross validation, as will be discussed shortly, the F1-score is also coupled with a standard deviation. These metrics should provide the true error of a system and therefore the performance of each individual part.

Not only an F1 score has been used to measure the performance of the classifier. A confusion matrix has been provided as to give a more complete representation for the performance of the entire classification pipeline.

2) Statistical Inspection of Parameters: The different sets of parameters and their effect on the resulting F1 score were plotted in a box plot. This, together with significance and post hoc tests, is used to determine the best parameter setting with respect to efficiency, overfitting reduction and classification score.

F. k-fold cross-validation

A stratified 3- & 4-fold classification has been used to train, test and validate the neural network for the aquarium and the swimming pool dataset respectively. The advantage of this approach is that all data is used for both training and validating the model, while still retaining an accurate measure of the true error. The k-fold classification is therefore a useful method to evaluate a machine learning model on a limited dataset. The dataset for this research is not necessarily limited. However, with this method we use the dataset to its full potential.

The training and validation set for each flow dataset are split into k different subsets of the original data. Two of the k different subsets are used for validation, whereas the rest k-2 is used for training and testing of the model. This

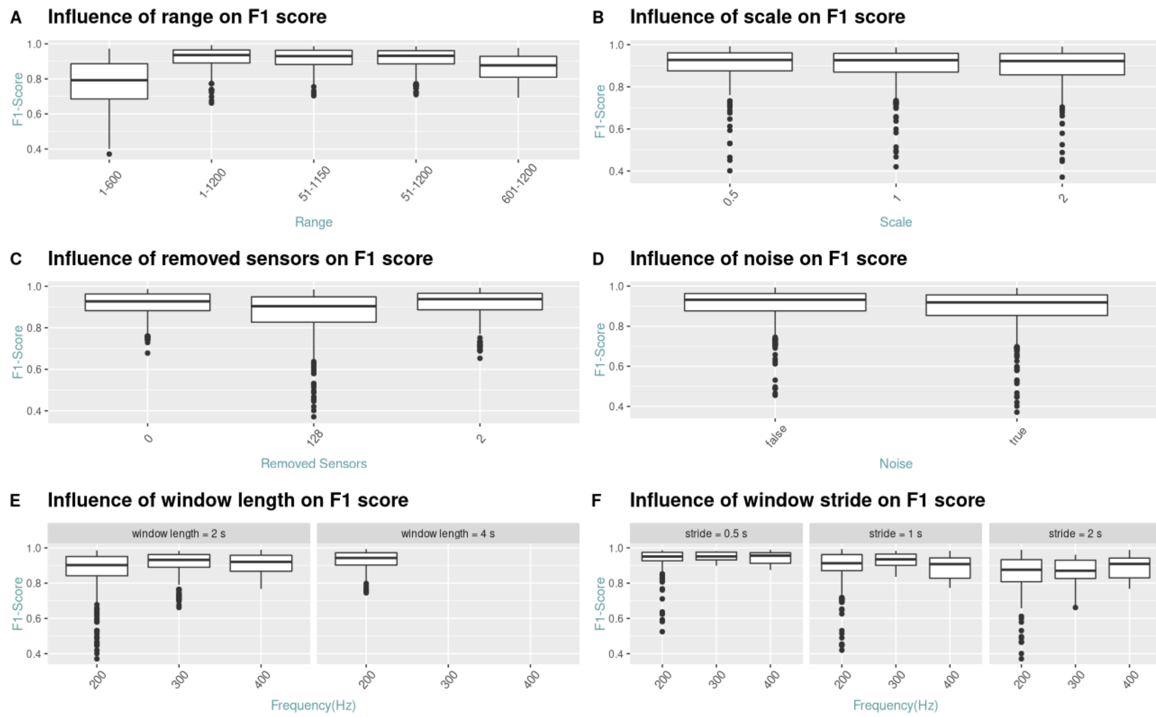


Fig. 6: Graphical representation of the different changes to the dataset that might influence the F1-score. Sub-figure A-F shows the effect of a specific parameter on the resulting F1 score of the ELM.

procedure is then repeated until all data has been used for both training, testing and validation. The resulting error of each validation will be averaged over all k times the model has been validated.

The selection of k should be chosen with some consideration. There are generally several techniques in which this can be done. For this kind of cross-validation, there is a bias-variance trade off for the choice of k. The higher the k, the lower the bias would be for the technique. Typically, cross-validation is chosen to be around $k = 5$ or $k = 10$. These values return error rate estimates that do not suffer from either high bias, or high variance [19]. We, however, have chosen for $k = 4$ and $k = 3$ for the aquarium dataset and the swimming pool dataset respectively. For this research, we split the two datasets based on the number of movements, which are 80 and 6 for the aquarium and the swimming pool dataset respectively. Since it is easy to divide 80 by 4 and 6 by 3, it is more convenient to select the subsets of data in this manner.

IV. Results

A. Preprocessing of the dataset

Figure 6 shows a box plot comparing the difference between the effects of each different parameter. The 7 different variables that have been changed in order to manipulate the feature extraction into a form better suited to generate the features are described below:

- *Range*: The selected subset of time samples
- *Scale*: The scaling of the feature space after feature extraction
- *Removed Sensors*: The sensors of which the data was omitted
- *Window Length*: The length (in seconds) of the moving window
- *Stride*: The step size (in seconds) with which the window moves
- *Frequency*: The frequency used in resampling the measured data
- *Noise*: The addition of a 6 dB SNR noise level

The dependent variable in this case was the F1 score that resulted from running the extracted features through the ELM. As mentioned earlier, this F1 score is an average of the k-fold cross-validation process. The ELM was therefore trained, tested and validated with the output of the dataset.

Since all possible permutation settings have been tested, some data might not be compatible. For instance, using a window length of 4 seconds with a sampling frequency of 400 Hz results in no output due to having too little data. A part of the outliers that are seen in Figure 6 could therefore be due to a combination of settings that are incompatible. However, it is usually the case that data that has significantly more outliers, are outliers that resulted from the choice of parameters.

Before any significance testing was done, a test for multicollinearity was performed. Using Pearson's r test, a negative correlation was found between frequency and window length, $r(1510)$ = -.41, $p < .01$. This is expected as these parameters both influence the granularity of the windows. No correlations were found for other variables.

In order to show the significance of each dataset change,

TABLE I: List of F1 scores for the different datasets. Each individual window score (w), as well as the total run score (r) is shown. Table A shows the F1 scores for the hand-picked feature extraction method. Table B shows the results of the auto-encoder. Table C shows the results of using raw data, this last approach does not use a moving window approach.

A: Hand-picked		train (w)	test (w)	train (r)	test (r)
	Dataset	F1$\pm$ σ (%)	F1$\pm$ σ (%)	F1$\pm$ σ (%)	F1$\pm$ σ (%)
	Aquarium	99.0 $\pm$ 2.1	98.1 $\pm$ 3.4	99.5 $\pm$ 0.6	99.1 $\pm$ 1.0
	Swimming Pool	71.0 $\pm$ 11.1	63.5 $\pm$ 12.5	98.6 $\pm$ 4.1	95.6 $\pm$ 9.4

B: Auto-encoder		train (w)	test (w)	train (r)	test (r)
	Dataset	F1$\pm$ σ (%)	F1$\pm$ σ (%)	F1$\pm$ σ (%)	F1$\pm$ σ (%)
	Aquarium	59.1 $\pm$ 13.2	34.1 $\pm$ 16.5	90.1 $\pm$ 9.4	48.8 $\pm$ 27.8
	Swimming Pool	20.1 $\pm$ 16.2	19.4 $\pm$ 15.8	25.1 $\pm$ 26.3	28.2 $\pm$ 30.1

C: Raw		train (r)	test (r)
	Dataset	F1$\pm$ σ (%)	F1$\pm$ σ (%)
	Aquarium	96.9 $\pm$ 2.86	95.1 $\pm$ 3.82

an ANOVA was conducted to compare the effects of range, scale, sensor removal, window length, stride, frequency and noise on the resulting F1 score in several conditions as depicted in Figure 6. We found a statistically significant difference (for $\alpha = 0.01$) in the average F1-score by the time sample range [F(4)=175.100, $p < 0.001$], the removed sensors [F(2)=94.044, $p < 0.001$], the addition of noise [F(1)=25.095, $p < 0.001$], the window length [F(1)=19.809, $p < 0.001$] and the stride [F(1)=369.611, $p < 0.001$]. The frequency and the feature space scale seem to have a significant effect.

A Bonferroni adjusted pairwise t-test revealed that the '1-1200' range resulted in a significantly higher F1 score than either the '1-600' or the '601-1200' range. The same goes for the '51-1150' and '51-1200' ranges. The three ranges that scored significantly higher did not significantly differ from each other. The removal of sensors was also found to be significant for removing sensors 1,2 and 8 as compared to only removing sensor 2. This was found to be significantly lower. There was no significant difference between removing no sensors and removing sensor 2. The addition of noise resulted in a significant increase of the resulting F-1 score. There was also a significant increase of performance for a window length of 4 seconds as compared to a window length of 2 seconds. Finally, all levels of the stride factor differed significantly, where a stride equal to 0.5 resulted in the highest F1 scores.

B. Comparison of classification scores

Here we compare the F1 scores of the current research with the F1 scores found using the techniques of [36] and [32]. All scores shown in Table I have been gathered during this research. The algorithms and flow datasets have been used to replicate the F1 score of the previous studies. The auto-encoder has also been added to this comparison of F1 scores.

Table I shows the results of the ELM for both the aquarium and the swimming pool datasets under optimal conditions. The optimal conditions for the aquarium were with the following parameters; a time sample range from 1 to 1200, noise was added, sensor 2 was removed, no feature space scaling was applied and the window properties were set to a length of 4 seconds, a stride of 0.5 seconds and a resampling frequency of 200 Hz.

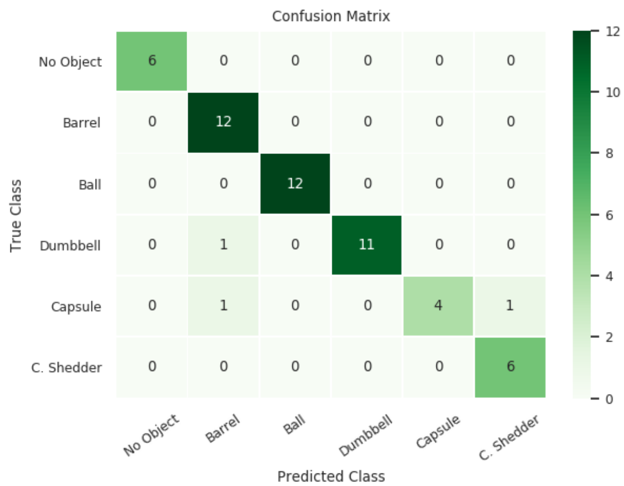
The optimal parameters are much the same for the swimming pool dataset. Two parameters are notably different, the time range was not limited since this had already been done to the dataset and the second sensor was not removed. The following parameter settings were used: noise was added no feature space scaling was applied and the window properties were set to a length of 4 seconds, a stride of 0.5 seconds and a resampling frequency of 200 Hz.

The table also shows the results of using the raw data in an ELM. This data only shows a full run, since no windowed approach is used when raw data is submitted.

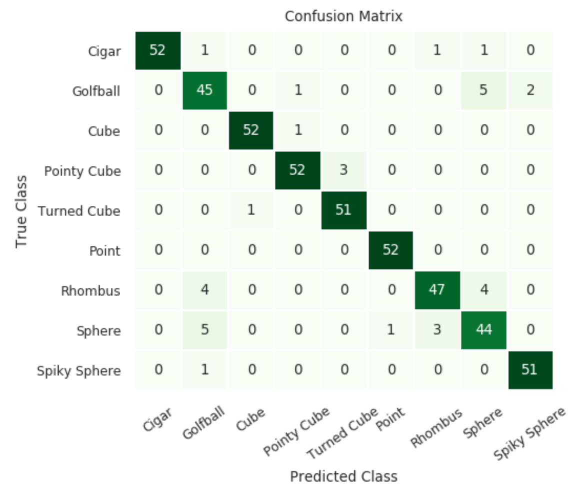
All two sample t-tests performed here use the F1 scores from the test scores of complete runs, unless specified otherwise. An independent samples t-test was conducted to compare the F1 score for classification using the features and the raw data, both from the aquarium dataset, as input. No significant effect was found according to the scores for the hand-picked features (M=99.1, SD=1.0) and the raw data (M=95.1, SD=3.82) conditions; t(5)=-2.064, p=0.09. Two independent samples t-test were performed to compare the aquarium and the swimming pool dataset for both feature extraction methods. First, for the hand-picked features, no significant difference seemed to be found between the F1 scores of the aquarium dataset (M=99.1,SD=1.0) and the swimming pool dataset (M=95.6, SD=9.4); t(5)=-0.764, p=0.48. Secondly, for the auto-encoder features, we also did not find a significant difference between the aquarium dataset (M=48.8, SD=27.8) and the swimming pool dataset (M=28.2, SD=30.1); t(5)=-0.983, p=0.39.

A notable difference can be found between the window F1 scores of the hand-picked features. There exists a significant difference between the window scores for the hand-picked features of the aquarium dataset (M=98.1,SD=3.4) and the swimming pool dataset (M=63.5, SD=12.5); t(5)=-5.437, p<.01.

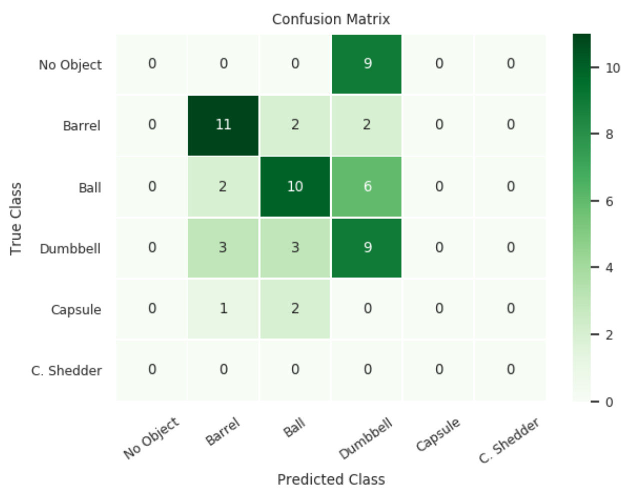
Four confusion matrices have been provided and can be found in Figure 7. These four confusion matrices show the predictions for the two flow datasets and both the feature



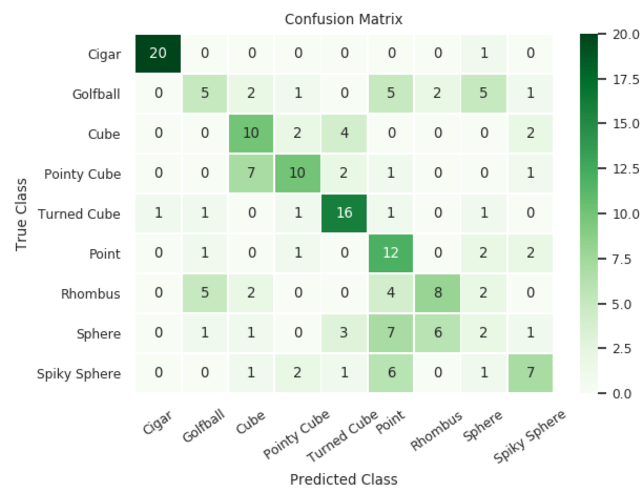
(a) Predictions of swimming pool dataset using the hand-picked feature extraction



(b) Predictions of aquarium dataset using the hand-picked feature extraction



(c) Predictions of swimming pool dataset using the auto-encoder feature extraction



(d) Predictions of aquarium dataset using the auto-encoder feature extraction

Fig. 7: Four confusion matrices that show the resulting classification of both the flow datasets and both the feature extraction techniques after being run through the ELM. Each confusion matrix shows the predictions per run. The raw data confusion matrix has not been added as it can be found in [32].

extraction techniques after running the resulting features through an ELM.

V. Discussion

We asked whether or not the features extracted from the raw data would outperform the raw data in and of itself when presented to the ELM and judged via the F1-score. We also asked whether a smaller sensor array, smaller surroundings and smaller objects will have a significant positive effect on the classification results of the Extreme Learning Machine. Finally, as a part of the initial research question, we also sought to determine the classification capabilities of the auto-encoder features.

Our hypothesis is that the feature extraction method will indeed outperform the raw data as features are computationally more efficient for the ELM [3] [5]. We also expect the smaller sensor array/surroundings/objects to have a positive effect since a smaller setup overall will most likely generate less disturbing wakes.

Now we have come to the conclusion that feature extraction does not necessarily outperform raw data. The F1-scores do not differ significantly, although the standard deviation is significantly lower. We can also conclude that a smaller sensor array, smaller surroundings and smaller objects have a partly significant positive effect on the classification scores. Namely, the classification scores increase significantly for windows, where less data is available, but not for complete runs. A discussion follows about the results from the parameter sweep, the two datasets and the two feature extraction methods.

A. Grouping Topology, Elongation and Surface Type

Looking at the confusion matrices in Figure 7, we can see that certain shapes are misclassified as certain other shapes more often. For instance, the sphere, the rhombus and the golf ball are relatively often mistaken for each other. This seems to imply that the topology of the rhombus and the surface type of the golf ball make a relatively small difference on the characteristic wake. These changes in topology and surface type do, however, still seem significant enough to properly identify the shape most of the time.

B. Feature and Raw Data Classification

Using the hand-picked or auto-encoder features has not shown to significantly increase performance over simply using raw data as an input for the ELM. The standard deviation is significantly lower though. It might be that the classification performance of the feature extraction is higher in theory, but that it does not show due to the high classification scores of both methods. Since both feature extraction methods show a high classification score, the differences between them are small and hard to attribute to any single (hyper)parameter.

While no significant difference has been found between the feature extraction and raw data, we have validated that the ELM is capable to efficiently process the data from an ALL. These features or raw data can then be used to correctly classify hydrodynamic wakes with high confidence. Selecting the correct parameter settings is essential for getting the high performance that the ELM got from the hand-picked features.

In all approaches careful tuning was applied through the use of nested cross-validation and noise. This resulted in minimal overfitting of the dataset. This is also signified by the small discrepancies between the training and testing sets.

C. Hand-picked and Auto-encoder Features

Auto-encoders have been shown to outperform hand-picked feature extraction methods for high-dimensional data [30][22]. It provides a promising way to generate labelled data. The generation of features for the aquarium dataset clearly works as the training classification score has an F1 score as high as 90.1%. However, the generalisability of those features are very low, since the classification score drops with about 50% for the test scores. The swimming pool dataset may have performed as badly as it did due to the high number of features generated by the auto-encoder, as there were only a handful of independent measurements.

Taking a look at the confusion matrices in Figure 7, we see that some objects are classified with relative certainty. For the swimming pool dataset (Figure 7c), the barrel, the ball and the dumbbell are classified relatively well. Similarly for the aquarium dataset, the cigar and the turned cube are classified relatively well. This seems to imply that there are some fundamental differences in the two flow datasets. These difference could be a result of the surrounding turbulence in the bodies of water.

D. Comparison of Runs and Windows

The classification of the windows was significantly better for the aquarium dataset. It seems that when there is less data overall, as in the windows, there is a significant increase in performance for the aquarium dataset as compared to the swimming pool dataset. We hypothesize that this is due to one of two reasons.

Firstly, there might be less turbulence in the body of water as it comes to rest more quickly than the swimming pool. This would imply that the measurements done for each individual shape is of a higher quality. This higher quality could result in a clearer distinction between characteristic wakes.

Secondly, the measurements might have less spatial gaps due to the small space in between the sensors of the sensor array. Spatial gaps happen when the object takes a long time to move in between the sensors and the sensors register less flow data. Having a small sensor array could result in a constant signal that encodes the characteristic wake, whereas the (sparse) sensor array for the swimming pool dataset might have spatial gaps of data while the object is moving from sensor to sensor.

E. Possible Shortcomings

Different shapes were used in both datasets. Due to the generalizability of the methods that were employed, this was not deemed a problem. However, for more reliable results perhaps using the same shapes in both environments would be better suited.

The F1-score might not be the best for comparing classifier performances. A higher F1 score of another classifier cannot necessarily be used as ”proof” that one is better than the other. Using more metrics, perhaps Cohen’s Kappa score, could be useful in future comparisons of classifiers. The Kappa score takes random classifications into account, and with that, tries to take the bias away from the actual distribution of the classification data.

The large hidden layer size of the auto-encoder does mean that there is still a large reduction of dimensionality which theoretically might result in an improvement for the auto-encoder.

Finally, the same kind of ALL system was used for the gathering of both flow datasets. However, through a faulty sensor the sensor array measuring the flow datasets were not equal. Any difference between the flow datasets might be due to the fact that a sensor was missing during measurements of the aquarium dataset.

F. Future Research

It has been shown that classification of hydrodynamic wakes can be done reliably and efficiently in different scenarios. This opens up the possibility for testing in various real-world applications. Also, there can be expanded further on this research by the following ideas.

An idea about using the random forest learning method came up during this research. Since data could be grouped according to their topology, surface type or elongation, perhaps a decision tree could be used. The decision tree could then be based on the characteristics of the shapes.

The auto-encoder was used for this research, but it seemed to give unreliable results in practice. These unreliable results mostly came in the form of strong overfitting or specific object misclassification. One of the possibilities is that the method employed reduced the data too drastically, losing a lot of significant information. Stacked auto-encoders could possibly give better results. The overfitting could also be a result of the fact that not enough data was used. Perhaps using more ALL-generated data in the auto-encoder would result in better classification scores and no overfitting.

VI. Conclusion

The current study illustrates the difference between the effects of sensor scale, environments and feature extraction techniques on hydrodynamic imaging using an artificial lateral line (ALL). By showing the effectiveness of the ELM classification under different situation, it has become convincing that this technique can be confidently used in underwater vehicles, canal tracking and other water related scenarios where other sensors might not work well. The system is clearly able to classify hydrodynamic wakes generated by objects that move past the sensor array.

VII. Acknowledgements

Special thanks to S.M. van Netten, B.J. Wolf and S. Römer for providing datasets, scripts and mental support during the writing of this thesis.

VIII. Data Availability

The datasets that were used in this thesis are available by contacting the authors of the original work [36][32].

References

[1] S. Dijkgraaf. "The functioning and significance of the lateral-line organs". In: *Biological Reviews* 38.1 (1963), pp. 51–105.

[2] American National Standards Institute. *Specification for octave-band and fractional-octave-band analog and digital filters*. 1986.

[3] P. Langley et al. "Selection of relevant features in machine learning". In: *Proceedings of the AAAI Fall symposium on relevance*. Vol. 184. 1994, pp. 245–271.

[4] G. An. "The effects of adding noise during backpropagation training on a generalization performance". In: *Neural computation* 8.3 (1996), pp. 643–674.

[5] M. Dash and H. Liu. "Feature selection for classification". In: *Intelligent data analysis* 1.3 (1997), pp. 131–156.

[6] J. Engelmann et al. "Hydrodynamic stimuli and the fish lateral line". In: *Nature* 408.6808 (2000), pp. 51–52.

[7] Z. Fan et al. "Design and fabrication of artificial lateral line flow sensors". In: *Journal of micromechanics and microengineering* 12.5 (2002), p. 655.

[8] J.P. Franosch et al. "Estimating position and velocity of a submerged moving object by the clawed frog Xenopus and by fishâa cybernetic approach". In: *Biological cybernetics* 93.4 (2005), pp. 231–238.

[9] S. Ali and K.A. Smith-Miles. "Improved support vector machine generalization using normalized input space". In: *Australasian Joint Conference on Artificial Intelligence*. Springer. 2006, pp. 362–371.

[10] B. Ćurčić-Blake and S.M. van Netten. "Source location encoding in the fish lateral line canal". In: *Journal of Experimental Biology* 209.8 (2006), pp. 1548–1559.

[11] Y. Yang et al. "Distant touch hydrodynamic imaging with an artificial lateral line". In: *Proceedings of the National Academy of Sciences* 103.50 (2006), pp. 18891–18895.

[12] S.P. Windsor et al. "The flow fields involved in hydrodynamic imaging by blind Mexican cave fish (Astyanax fasciatus). Part I: open water and heading towards a wall". In: *Journal of Experimental Biology* 213.22 (2010), pp. 3819–3831.

[13] G.E. Hinton, A. Krizhevsky, and S.D. Wang. "Transforming auto-encoders". In: *International conference on artificial neural networks*. Springer. 2011, pp. 44–51.

[14] G. Huang et al. "Extreme learning machine for regression and multiclass classification". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42.2 (2011), pp. 513–529.

[15] T. Jayalakshmi and A. Santhakumaran. "Statistical normalization and back propagation for classification". In: *International Journal of Computer Theory and Engineering* 3.1 (2011), pp. 1793–8201.

[16] J. Masci et al. "Stacked convolutional auto-encoders for hierarchical feature extraction". In: *International conference on artificial neural networks*. Springer. 2011, pp. 52–59.

[17] J. Deng et al. "Sparse autoencoder-based feature transfer learning for speech emotion recognition". In: *2013 humaine association conference on affective computing and intelligent interaction*. IEEE. 2013, pp. 511–516.

[18] J. Gehring et al. "Extracting deep bottleneck features using stacked auto-encoders". In: *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE. 2013, pp. 3377–3381.

[19] G. James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[20] S.M. van Netten and M.J. McHenry. "The biophysics of the fish lateral line". In: *The Lateral Line System*. Springer, 2013, pp. 99–119.

[21] J. Chorowski, J. Wang, and J.M. Zurada. "Review and performance comparison of SVM-and ELM-based classifiers". In: *Neurocomputing* 128 (2014), pp. 507–516.

[22] Jeremy Liu et al. "Autoencoder-derived features as inputs to classification algorithms for predicting well

failures”. In: *SPE Western Regional Meeting*. Society of Petroleum Engineers. 2015.
[23] J. Xu et al. “Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images”. In: *IEEE transactions on medical imaging* 35.1 (2015), pp. 119–130.
[24] K. Audhkhasi, O. Osoba, and B. Kosko. “Noise-enhanced convolutional neural networks”. In: *Neural Networks* 78 (2016), pp. 15–23.
[25] G. Liu et al. “A review of artificial lateral line in sensor fabrication and bionic applications for robot fish”. In: *Applied bionics and biomechanics* 2016 (2016).
[26] W. Sun et al. “A sparse auto-encoder-based deep neural network approach for induction motor faults classification”. In: *Measurement* 89 (2016), pp. 171–178.
[27] L.H. Boulogne et al. “Performance of neural networks for localizing moving objects with an artificial lateral line”. In: *Bioinspiration & biomimetics* 12.5 (2017), p. 056009.
[28] M. Alsalman, B. Colvert, and E. Kanso. “Training bioinspired sensors to classify flows”. In: *Bioinspiration & biomimetics* 14.1 (2018), p. 016009.
[29] A. Ristolainen et al. “Hydromorphological classification using synchronous pressure and inertial sensing”. In: *IEEE Transactions on Geoscience and Remote Sensing* 56.6 (2018), pp. 3222–3232.
[30] J. Sun et al. “Learning sparse representation with variational auto-encoder for anomaly detection”. In: *IEEE Access* 6 (2018), pp. 33353–33361.
[31] B.J. Wolf et al. “Bio-inspired all-optical artificial neuromast for 2D flow sensing”. In: *Bioinspiration & biomimetics* 13.2 (2018), p. 026013.
[32] S. Römer. “Underwater Object Classification using an Artificial Lateral Line and Extreme Learning Machine”. In: *MSc Thesis Rijksuniversiteit Groningen* (2019).
[33] B.J. Wolf and S.M. van Netten. “Hydrodynamic Imaging using an all-optical 2D Artificial Lateral Line”. In: *2019 IEEE Sensors Applications Symposium (SAS)*. IEEE. 2019, pp. 1–6.
[34] B.J. Wolf, S. Warmelink, and S.M. van Netten. “Recurrent neural networks for hydrodynamic imaging using a 2D-sensitive artificial lateral line”. In: *Bioinspiration & biomimetics* 14.5 (2019), p. 055001.
[35] D. Xu et al. “Sensor placement optimization in the artificial lateral line using optimal weight analysis combining feature distance and variance evaluation”. In: *ISA transactions* 86 (2019), pp. 110–121.
[36] B.J. Wolf et al. “Shape classification using hydrodynamic detection via a sparse large-scale 2D-sensitive artificial lateral line”. In: *IEEE Access* 8 (2020), pp. 11393–11404.