A SEMI-SUPERVISED MACHINE LEARNING APPROACH FOR ACOUSTIC MONITORING OF ROBOTIC MANUFACTURING FACILITIES

by

Jeffrey Bynum

A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
In Partial fulfillment of
The Requirements for the Degree
of
Master of Science
Electrical Engineering

Committee:

_____	Dr. Jill Nelson, Thesis Director
_____	Dr. David Lattanzi, Committee Member
_____	Dr. Vasiliki Ikonomidou, Committee Member
_____	Dr. Monson Hayes, Chairman, Department of Electrical and Computer Engineering
_____	Dr. Kenneth Ball, Dean for Volgenau School of Engineering

Date: _____ Summer 2019
George Mason University
Fairfax, VA

A Semi-supervised Machine Learning Approach for Acoustic Monitoring of Robotic Manufacturing Facilities

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

By

Jeffrey Bynum
Bachelor of Science
George Mason University, 2016

Director: Dr. Jill Nelson, Professor
Department of Electrical and Computer Engineering

Summer 2019
George Mason University
Fairfax, VA

Dedication

I dedicate this thesis to my parents. Your love and encouragement made this possible.

Acknowledgments

I would like to thank the members of the Lattanzi Research Group for your invaluable mentorship, guidance, and feedback. In addition, portions of this work were supported by a grant from the Office of Naval Research (No. N00014-18-1-2014). Any findings or recommendations expressed in this material are those of the author and does not necessarily reflect the views of ONR.

Table of Contents

Page

List of Tables . . . vii
List of Figures . . . viii
Abstract . . . x
1 Introduction . . . 1
2 Prior Work . . . 3
2.1 Feature Engineering . . . 4
2.2 Spectral Smoothing . . . 7
2.3 Convolutional Based Spectrogram Approaches . . . 9
2.3.1 Event Detection . . . 14
2.3.2 Damage Identification . . . 17
2.4 Similarity Analysis . . . 17
2.5 Clustering . . . 18
2.6 Research Contribution . . . 21
3 Preliminary Findings . . . 22
4 Methodology . . . 26
4.1 Dataset Development . . . 27
4.2 Supervised Actuation Detection . . . 28
4.2.1 Architecture #1 . . . 30
4.2.2 Architecture #2 . . . 30
4.2.3 Architecture #3 . . . 31
4.2.4 Segmentation and Validation . . . 32
4.3 Unsupervised Analysis . . . 33
4.3.1 Feature Engineering . . . 33
4.3.2 Similarity and Clustering Analysis . . . 35
5 Results . . . 38
5.1 Supervised Classification . . . 38
5.2 Unsupervised Clustering . . . 42
5.2.1 Y-axis Similarity Analysis . . . 42

5.2.2 Y-axis Clustering Analysis 44
5.2.3 X-axis Similarity Analysis 52
5.2.4 X-axis Clustering Analysis 53
6 Discussion 61
7 Conclusion 64
A Feature Equations 65
References 66

List of Tables

Table Page

4.1 CNN Architecture # 1 31

4.2 CNN Architecture # 2 32

4.3 CNN Architecture # 3 33

4.4 Unsupervised Features 35

5.1 Validation on Ground Truth Spectrogram (y) 39

5.2 Validation on Ground Truth Spectrogram (x) 39

A.1 Unsupervised Features 65

List of Figures

Figure		Page
1.1	SCARA and Cartesian reference frame used (reprinted from [1] with permission from Cambridge University Press)	2
2.1	Example of spectral edge in spectrogram	10
2.2	Example of convolutional kernel as parameter sharing	12
3.1	Actuation spectrogram examples	23
3.2	Neural Network Architecture Classification Preliminary Attempts	24
3.3	Confusion matrices for shallow/deep network classification	25
4.1	Example of CNN convolutional layers in architecture (#3). Generated using the resource presented in [2]	30
4.2	Frequency Response and Spectral Distribution (No Smoothing)	36
4.3	FFT and associated spectral responses for y-axis subclasses	37
4.4	Spectral distribution comparison between y-axis subclass 1(a) and 1(b)	37
5.1	Validation of Architectures - k-fold Cross Validation	38
5.2	Poor classification around actuation transitions with wideband noise	40
5.3	Wideband noise effects on x-axis classification accuracy	41
5.4	Y-axis cosine similarity	43
5.5	Y-axis principal component scatter plot (2D)	45
5.6	Principal component scatter plot (3D) overlaid with polyhedron clusters	48
5.7	Principal component scatter plot (3D) overlaid with polyhedron clusters (removal of 3(a) visualization)	49
5.8	Y-axis k-means clustering results (3 centroids)	50
5.9	3D visualization of PC space with polyhedron clusters as K-means class assignments	51
5.10	Y-axis k-means clustering results (7 centroids)	52
5.11	Sum of Squared Error (SSE) between Plot: y-axis motions	53
5.12	Silhoutte score: y-axis motions	54
5.13	x-axis cosine similarity	55

5.14 x-axis principal component plot 56
5.15 X-axis k-means clustering results (8 centroids) 57
5.16 3D visualization of PC space with polyhedron clusters as K-means class assignments (X-axis motions) 58
5.17 SSE graph: x-axis sub-classes 59
5.18 Silhouette graph: x-axis sub-classes 60

Abstract

A SEMI-SUPERVISED MACHINE LEARNING APPROACH FOR ACOUSTIC MONITORING OF ROBOTIC MANUFACTURING FACILITIES

Jeffrey Bynum

George Mason University, 2019

Thesis Director: Dr. Jill Nelson

Diagnosing characteristic industrial equipment characteristic behavior non-invasively and in situ is an emerging field of study. An algorithm was developed to acoustically monitor mechanical systems with minimal data labels. The methodology was evaluated using a semiconductor device manufacturing process, consisting of a Selective Compliance Assembly Robot Arm (SCARA) system, via an embedded microphone array. Combined unsupervised and supervised data analysis techniques to identify critical processes for eventual life-cycle tracking, was demonstrated. A spectrogram-based convolutional neural network performed primary robotic motion segmentation with an average accuracy of 85% using ground-truth validation data. Subsequent unsupervised analysis using similarity metrics as well as k-means clustering on engineered features had mixed success in distinguishing secondary robotic actuations. A semi-supervised technique was viable to differentiate characteristics in robotic motions with limited available labeled data. Data visualizations demonstrated potential limitations in engineered feature separability as well as probable error sources. Further refinement is required for better segmentation accuracy as well as identifying features that represent secondary characteristics in manufacturing systems.

Chapter 1: Introduction

Industrial manufacturing systems must maintain fabrication tolerances as well as meet production capacity requirements [3]. These systems demand persistent monitoring and maintenance due to progressive mechanical wear. Conventional approaches traditionally relied on operator oversight and experience for monitoring system health; however, remote monitoring techniques are increasingly applied. Remote techniques allow flexibility for monitoring inaccessible fabrication environments as well as providing continuous production diagnostics. Moreover, remote monitoring techniques can robustly fuse complex data features for better understanding behind normal and damaged system states.

Industrial robots are complex mechanical systems that combine components such as motors, bearings, housings, linear rails, and brakes. These systems typically perform a variety of actuations generalized in 3D Cartesian space. As exampled in robotic manufacturing platforms such as in Figure 1.1, a series of these primary motions describe a process. Actuations for robotic systems can include y-axis (base movement), x-axis (arm extension/retraction), and z-axis (body extension/retraction) motions. Primary motions can be further subdivided into separate, secondary subclasses. Each subclass describes a physically different actuation along the same axis of travel. For example, motion from point A to point B [class label 1(a)] and motion from point B to point A [class label 1(b)] describe two potential subclasses. These subclass actuations are combined sequentially depending on process.

These primary motion classes (x, y, z) contain various engaged mechanical hardware over different, discernible sequences. Audible motion characteristics stemming from differences in system dynamics, could possibly be captured through non-invasive, audio recordings. Deviations in system dynamics coupled with corresponding changes in harmonic content

may indicate progressive component damage. While secondary actuations are similar, each motion class generates distinct acoustic emissions from differing mechanical responses.

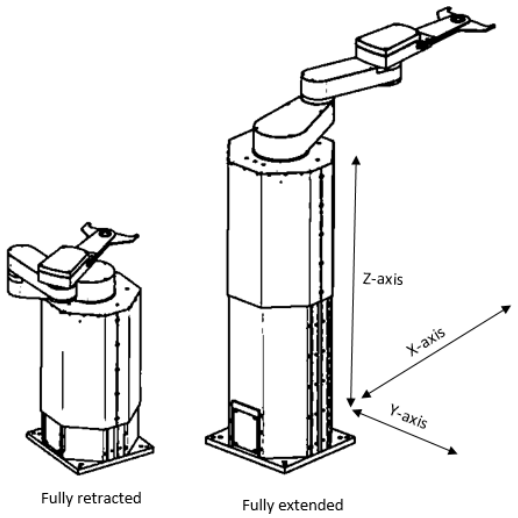


Figure 1.1: SCARA and Cartesian reference frame used (reprinted from [1] with permission from Cambridge University Press)

This work investigates a semi-supervised machine learning approach to isolate and classify acoustic signals corresponding to robotic actuations. The problem consists of two primary elements including segmenting actuation signals from audio recordings, followed by unsupervised similarity analysis of captured segmentations for further characterization. The study addresses unmet research needs by investigating challenges of acoustic health monitoring in a realistic manufacturing settings with limited labeled data. These techniques can be applied to understand system health and eventual life-cycle assessment.

Chapter 2: Prior Work

While monitoring industrial components using auditory characteristics has been studied, numerous practical challenges for monitoring systems in realistic environments remain. Several recent comprehensive literature surveys have outlined various trends in remote health monitoring [3–5]. Several challenges are noted for diagnosing changes in mechanical systems including: robustness to noise artifacts, acoustic event detection, feature extraction methods, and automated damage detection algorithms.

With regards to acoustic monitoring, distinct audio features used to classify events as well as damaged mechanical states are often difficult to determine. Signal features correlating with damage are often weak in amplitude compared to other recorded operations [6] and deviations reflecting damage are not fully understood. While mechanical faults are known to influence system behavior, often reflected in the frequency spectra [6–8], physical phenomena varies with degradation type. Bearing faults, for example, exhibit stress waves influencing high frequency system dynamics [9], which may not be reflected in other mechanical fault behavior.

Expert knowledge is often required to generate features and categorize audible dynamics corresponding to mechanical wear. Moreover, unique features must be separable, distinctive, and representative for audio signature classification [3, 10–13]. There have been a variety of studies on generating such features [3, 10, 11, 14, 15], however these features are typically domain dependent. Damage classification typically involves tuning features corresponding to ground truth and altered states; however, data driven methods are more robust to capturing changing dynamics [9] as well as allowing for practical implementation outside of traditional numerical models [4].

Employing modern machine learning methods [4], including deep neural networks, can

allow for better feature representations as well as relationships between features and underlying characteristics [16–18]. Broadly, machine learning architectures for health monitoring can be divided into a feature extraction and classification stage. Features can be extracted, using time-frequency characteristics [3, 10], or automatically learned through deep autoencoders or deep neural-network architectures [3]. A recent survey outlined features derived from mechanical faults in rotating industrial machinery could be classified using neural network, k-Nearest Neighbors (k-NN), and Support Vector Machine (SVM) methodologies [3].

The remainder of prior work is divided into sections based on primary challenges found in similar literature. Features and their applications to structural health monitoring and audio classification are discussed followed by harmonic data smoothing. Spectrogram features as well as convolutional neural networks are subsequently introduced. Two main subbranches of convolutional neural networks - event detection and damage identification - are discussed. Similarity analysis and data clustering sections are introduced. Potential contributions from this study, addressing unmet research needs, are noted at the end of the chapter.

2.1 Feature Engineering

Unique audio signatures used to classify damaged mechanical states and acoustic events are not easily generalized or understood across various applications [6]. Simplistic techniques to isolate noise (waveform data not correlated with damage), including band-pass filtering, can unintentionally decrease acoustic emission information corresponding to damage [6].

A comprehensive survey for multidisciplinary audio feature extraction and related processing is presented in [14]. Dennis (2014) provided a comprehensive overview between traditional speech recognition literature and environmental sound events including the motivation behind sound as a 2D data type [11]. Relationships between audio features and acoustic event detection as well as environmental event classification were further explored in [11]. Extensive potential parameters exist for audio-based classification and structural

health monitoring domains; these features often rely on prior assumptions and expert intuition [3, 10, 11, 14].

An extensive list of audio features for different domain spaces are presented in [14] and [15]. Features are typically domain dependent [14], [11]; however, similar descriptors exist in health monitoring, speech processing, and sound event classification fields. Several authors including [14] discuss data dimensionality reduction with relevance to audial feature vectors.

Manually extracted features are generalized to temporal, statistical, and spectral content. Specific features includes Mel-frequency cepstral coefficients (MFCCs), zero-crossing rate, auto-correlation coefficients, energy metrics, as well as spectral characteristics [3, 10]. Spectral shapes are commonly used for describing instrumental sound characteristics including spectral skewness and spectral kurtosis as well as amplitude and peak frequency content. Other temporal features are listed including attack, sustain, and decay for structured audio content [14].

Most commonly, damage is characterized in the time-frequency domain through spectral measures captured in Short-Time Fourier Transform (STFT) descriptors [3, 6, 19]. In one study, structural damage was determined through changing patterns within peak harmonic content [19]. Higher-order Fourier moments such as spectral skewness and spectral kurtosis were used in various studies as features in fault classification [4]. Spectral peaks often contain unique descriptor sets for audio signatures [12]. A list of frequency based features used in several studies for fault diagnosis were presented in [4]. Temporal-spectral feature vectors were successfully in conjunction with Deep Neural Networks (DNNs) and SVMs [4].

In one study, frequency based features including spectral kurtosis and cross-correlation have identified bearing wear using k-NN clustering [3]; these features have additionally captured faults related to gear-box degradation as well as gear cracking [3].

Ubhayaratne et al. (2017) discussed progressive wear in sheet metal fabrication using frequency descriptors in audio [8]. Researchers were able to track progressive tooling wear, citing an unmet research need via adapting acoustic emission. Literature studies vary in specific frequency and temporal characteristics with regards to progressive tooling wear;

shifting high or low frequency components that correspond to damage is debated throughout literature. Acoustic signatures from progressive wear potentially occur at harmonic ranges not detectable by humans. The author concluded that time-domain features are not necessarily representative since drastic signal changes are usually coupled with faults, not progressive wear. Peak amplitudes, peak frequency, root-mean-square were used as damage identifying features, projected on Hilbert space. The study suggested frequency changes corresponding with damage were more discernible than features embedded in raw waveform information alone.

Glowacz (2018) classified three damaged motor states using non-invasive acoustic measurements [20]. The author used absolute difference methods for Fast Fourier Transform (FFT) estimates between damaged and undamaged states. Peak frequency components were vectorized and used in an artificial neural network (ANN) classifier. The author noted the non-invasive procedure could classify damage; however, generalizing results to other problems would be difficult since spectrum analysis was directly tied to specific motor dynamics. Guillen et al. (2018) similarly used features derived from STFT responses to identify induction motor faults via current responses [21].

Statistical features including kurtosis and skewness were correlated with bearing faults in one study [22] in addition to crest factor [7]. Skewness, spectral features, and principal components were used to diagnose faults in rotating machinery [3]. Authors noted signal correlation analysis could be used to determine damaged and healthy signal states [19].

Certain features are debated in literature for practicality in noisy environments. While MFCC features are typically used in a variety of sound classification literature, their usefulness decreases outside speech processing research [11]. According to [11], Lower-level descriptors have outperformed MFCCs in sound event classification with high corruptive noise. Moreover, typical methods such as MFCCs, Hidden Markov Models (HMM), Gabor-filters, MPEG-7 descriptors have difficulty classifying unstructured, environmental sound data. The author specifically notes that lower order cepstral coefficients are adversely affected by noisy data.

Bach et al. suggested that other feature sets, such as amplitude modulation, can outperform MFCCs with noise corrupted signals in speech detection [23]. Their work relied on an SVM trained to recognize speech with corrupting background noise. While the study was successful in detecting speech in urban environments, the authors cited the need for generalized features in realistic environments. According to the authors, current data sets are not necessarily representative of practical, varied noise conditions.

Some limitations exist with manual feature extraction. Firstly, harmonic features are often predicated on prior knowledge and generalized assumptions for specific problems. Comprehensive audial descriptors presented in [14] are sometimes dictated by external factors such as audio structural composition (periodicity, stationarity) and environmental noise. Secondly, while statistical and spectral moments, such as crest factor, skewness, and kurtosis, have identified machine degradation, variations in specific parameters are intrinsically tied to non-generalizable systems; induction motor fault dynamics, for example, is fundamentally different than gear-wear damage characteristics.

Automatic feature generation can provide more generalizable features at the cost of practical intuition. According to some authors including [4], manually created feature sets are somewhat problem specific coupling specificity with expert intuition. Deeper networks allow richer, more complex relationships between features [4,24]. Moreover, deeper architectures are often robust with respect to noise [4]. Expert crafted features are less generalized especially in spectral domains [10]. These methods, however, have less physical intuition [3].

2.2 Spectral Smoothing

Corruptive noise was cited as a primary challenge across domains for acoustic signal classification. Damage identification, speech recognition, as well as bio-medical signal processing have used optimal filter design to improve estimation and feature extraction. These studies discuss improvements to spectral content through auto-regressive smoothing.

While a detailed description of smoothing filters and their applications are presented in [25], Savitzky-Golay filters have been used to filter acoustic signals in mechanical damage applications [26,27]. Savitzky-Golay filtering fits a polynomial, with a specified degree, to a data series by minimizing mean-squared error [25]. The smoothing technique retains peak harmonic content as well as high frequency dynamics.

The authors in [28] isolated instances of gear pitting using spectral kurtosis identification. Their research demonstrated an optimal denoising filter, applied to signal residuals, demonstrated improvements to isolating damage instances in high noise and signal interference conditions. The optimal filtering, through Weiner Filtering, allowed resonant frequencies corresponding to non-stationary signals to be maintained compared to lower order filtering methods.

Authors in [29] cited the effects from ARMA and associated auto-regressive techniques on signals corresponding to gear tooth faults. The study demonstrated filtering procedures, based on auto-regressive and minimum entropy deconvolution metrics, could improve signal kurtosis in gear tooth signals. The filtering measures augmented gear tooth fault identification by retaining impulse Characteristic gear spalling in the spectral domain.

In other domains, spectral smoothing can improve characteristic features. One study demonstrated that ARMA techniques can be applied to physical data to better estimate spectral characteristics with noise [30]. The authors noted situations where autoregressive techniques can be applied such as spectral estimates with definitive peak structure. Other studies have applied least-squares denoising procedures on noisy ECG signals using spectral smoothing while retaining relevant peak information [31, 32]. The method has improved spectral content estimation in noisy, semi-periodic structures [32, 33]; however, spectral content can be over-smoothed, or under-smoothed, which can change relative peak width and intensities [33]. Peiyang et al. (2015) outlined regressive techniques employed in EEG signals [34].

Several auto-regressive methods were introduced to improve speech recognition under noisy conditions [12,35]. Smoothing spectral responses can improve formant discovery [12].

An auto-regressive, sliding window approach was introduced to emphasize spectral characteristic peaks [35]. Reducing spectral valleys improved speech representation significantly in their study. Spectral smoothing has increased format identification in noisy speech signals [36].

Martin (2001) described noise influence on Power Spectral Density (PSD) for speech signals [37]. Recursive mean-squared estimation was used to compensate for non-stationary noise allowing for better representation of speech PSD. The method employed a 256-point recursive window with smoothing length of 0.2 seconds. The paper discussed some affects from smoothing spectral content. Frequency peaks are widened altering harmonic structure. PSD smoothing coupled with a novel algorithm were able to capture weak sounds and distinguish PSD of non-stationary noise.

Generalized filter strategies, such as auto-regressive models, could benefit feature extraction in noisy environments. These smoothing procedures are employed across various disciplines to augment and amplify relevant signal characteristics.

2.3 Convolutional Based Spectrogram Approaches

A typical method when applying deep learning methods to acoustic data involves mapping 1D audio waveforms to 2D spectrogram images. Using STFT procedures, overlapping frequency content over short segments of time can estimate temporal-frequency content in audio data. This data representation can capture evolving harmonic content over time - often being exploited with convolutionally stacked auto-encoders or convolutional neural network classification [11].

Dennis (2014) introduces a motivation behind 2D representations of chaotically structured sounds [11]. One dimensional waveform representations tend to capture less informed spectral content than 2D images. Audio context is important – both in capturing events with discrete windowing procedures as well as sound event structure. Fanioudakis and Potamitis (2017) additionally described sound as a multidimensional data type [38]. Acoustic events

can be represented as a 2D matrix using a STFT representation or a 3D representation where colorized pixels describe sound intensity.

Sound events, as defined by [11], encompass sound with definable harmonic content including spectral edges. An example outlining spectral edges is presented in Figure 2.1. Due to varied distributed harmonic content in the frequency domain and diffuse noise, acoustic events demonstrate visual gradients in spectrograms. The mapped data-type allows 2-Dimensional object detection algorithms or other image based learning strategies to capture spectral characteristics.

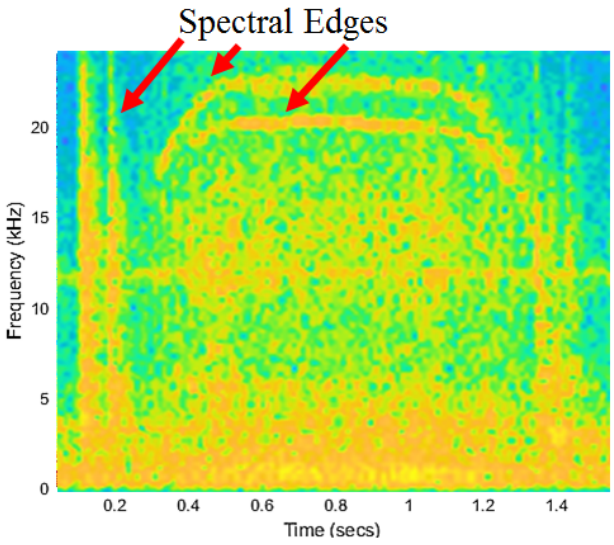


Figure 2.1: Example of spectral edge in spectrogram

Similarities and differences between spectrograms and image content are further noted in [11]. Based on the mixing principal of sound, the highest energy signal is recorded. Diffuse noise can potentially mask spectral information. This fundamentally differs from image data. In images, overlapping objects may obscure each other whereas audio sources are additively mixed. While the author in [11] presents a robust sound event recognition algorithm based on localized spectrograms features, they admit noise may adversely affect

sound event detection. Random noise masking can be easily misclassified as a sound event from similar spectral energies. This may be impossible to avoid in high noise environments with wide-band energies observed in spectrograms. Moreover, audial objects differ from image objects since content is frequency invariant (no rotational component). Similar to images, gradient of pixel transitions in spectrograms are more distinguishable than STFT data alone (analogous to pixels). Typically, spectrogram classification adapting image processing architectures involves normalization. The author in [11] re-scaled spectrograms intensity to [0, 1] range. The author notes varying color scale can better distinguish events; color pixel distributions can provide another potential avenue for spectrogram classification.

Temporal frequency content, described by edges in spectrograms can be captured by complex machine learning architectures such as convolutional neural networks. A convolutional neural network (CNN) can be generally described as a neural network with one or more hidden, convolutional layers [39, 40]. Analogous to neural network classification, neurons are initialized with weights and biases and are updated using back-propagation minimization between an input and target output [40]. However, convolutional layers are added to map higher dimensional features in input data to fully connected neurons for classification, regression, etc. The convolutional kernels are learned feature maps, allowing dimensionality reduction of input data through parameter sharing, see Figure 2.2 [41]. Kernel mapping can capture local, spatially invariant characteristics including edges and textures, relating to global patterns from inputs [39]. Parameter reduction through convolution, stride, and pooling operators are discussed in [39, 42]. Further derivations including back-propagation, stochastic gradient descent, and activation functions are presented in [39].

Relu layers are often added after convolutional layers in order to increase non-linearities between input-output mapping [39]. The added non-linear representation better captures invariant features and characteristic of inputs. ReLu layers, in practice, have outperformed other non-linear mapping functions by retaining gradients otherwise reduced. An optimizing

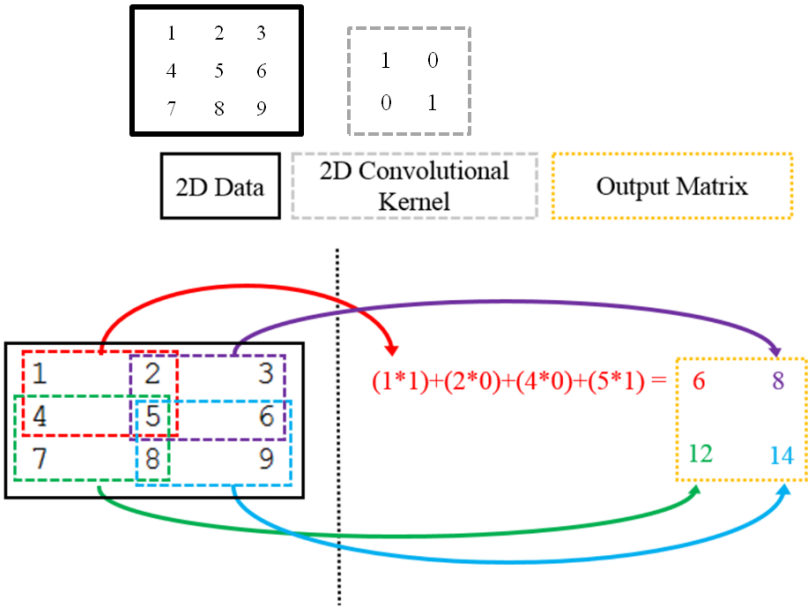


Figure 2.2: Example of convolutional kernel as parameter sharing

method is then used to minimize loss between an input and its target output using back-propagation, typically using stochastic gradient decent with a small subset of training data [39]. This technique estimates gradients for parameter updating rather than unstable effects for gradient updates on single training examples. Parameters are subsequently learned from input-output mapping directly rather than tuned from expert intuition.

CNNs often contain an immense amount of hyper-parameters required for training; however, some studies effectively trained spectrogram based CNNs with minimal hyper-parameters. The authors [9] demonstrated that a CNN could capture distinguishing features, despite noise and limited pre-processing. Lee et al. (2016) summarized that a non-optimal hyper-parameterized network was still valid and could result in high classification accuracy [9]. Hyper-parameter studies are required for optimal architectures; however, these studies are not evidently observed throughout literature. One study introduced particle swarm methods to select architecture and learning parameter [4]. Another study described

CNN architecture considerations for audio classification [10]. Hyper-parameters were chosen in one study based on a simple method of convergence speed [43]. Hyper-parameters are often multivariate and difficult to optimize; [9] lists several studies addressing hyper-parameter selection.

CNNs are a supervised machine learning method, requiring labelled training data. This requirement tends to limit implementation since labeled data describing a variety of systems states often does not exist. Several studies address class imbalances between damaged and healthy states [3, 4, 4, 24]; not many examples of faults are known [4]. Zhang et al (2012) describes data-set scarcity for non-structured environmental sounds as well as the associated cost from procuring labeled data-sets [44]. The authors in [44] subsequently present a method to augment an existing labeled data-set using a semi-supervised machine learning method.

While effective for handling complex and locally related input features, Alías (2016) outlined several limitations for direct implementations of spectrograms and spectrogram-based CNN architectures for classifying audio content [14]. Firstly, harmonic content is generally complex in the frequency axis, often requiring transforms for optimal kernel mapping for spatially linking useful features. Additionally, deeper architectures have less intuitive understanding due to high dimensional feature spaces and associated connections. Moreover, over-fitting with respect to area-specific domains, is a concern. Dropout can provide one method to limit DNN over-fitting concern [43].

Spectrograms have been used extensively in different domains stemming from two key areas in relation to this study: event detection and damage classification. Event detection broadly describes isolating and detecting sound events, including speech, through features localized in spectrograms. CNN-based damage classification literature incorporates differentiating spectrograms of damaged and healthy system states.

2.3.1 Event Detection

Many literature studies cite challenges with capturing events in spectrograms through feature analysis. CNNs, as well as stacked convolutional auto-encoding networks, are commonly employed to distinguish definable events, speech, and environmental sounds from corruptive, non-stationary noise. Acoustic event detection, or classification of spectrogram elements present in non-speech signals, was outlined in [45] and [11]. The authors in [45] outlined how relevant features detected in spectrograms using convolutional architectures could outperform deep non-convolutional architectures.

Binary classification studies between targeted audio events and null events are discussed in [11]; classification tasks in unlabeled spectrograms are non-trivial. Typically, sound events are shorter in duration than the recorded waveform. Windowing spectrograms for potential audio event detection is complicated due to non-optimal window sizes, generalized for a variety of signal events.

The authors [44] noted higher dimensional CNN and image representations of audio data outperforms other classification algorithms with high noise content. The authors also introduced a procedure to divide an arbitrary spectrogram into smaller segmentations for identifying and classing audio events using CNNs.

Analogously, [46] adopted existing architectures, including AlexNet and GoogleNet, for classifying audio events in spectrograms as images. The authors demonstrated limitations behind directly adapting image processing based CNNs. Spectrogram inputs have a rectangular aspect ratio – often significantly different than traditional square CNN inputs. The authors noted and applied an adaption between rectangular and square input spectrogram-images. Cross recurrent plots of the audio segments augmented classification and provided another visualization tool. In summary, Recursive Convolutional Neural Nets (RCNNs) rejected noise stemming from mixed signal sources – outperforming typically used CNNs and RNNs.

Compared to speech and music audio classification, certain environmental sounds are

chaotically structured [11, 44, 46–48] with wider frequency and amplitudes [44]. Environmental sounds are often non-stationary and non-periodic [44, 48]. Complex noise shapes, non-stationary events, and aperiodic structures are present in speech and environmental sound literature [44]. Machine learning methods have analogously been used to classify sounds in these domains [44, 49].

One author defines specific structural differences between speech and environmental signals [11]. Environmental sounds often have a lower signal to noise ratio with uncontrolled corruptive noise and microphone isolation with respect to target audio. Sound events typically contain more distinctive spectrogram visualizations than speech based on impulsive and harmonic content. These advantages can overcome limitations in speech processing literature using image-like content.

CNN processing environmental sounds was discussed in several papers including [50]. These authors similarly adapted image processing architectures for spectrogram-derived information. Ozer (2018) noted the lack of prescribed CNN architectures for environmental sounds – the authors used an initial kernel map of 11x11 for their classification tests [50]. Adapting object recognition CNNs successfully classified environmental audio events [46].

Typical speech processing literature is insufficient to handle more exotic audio structures [47]. Environmental sounds have particular traits, often exhibiting non-repeatable phenomes which limits traditional feature generation [48]. A comprehensive taxonomy in audio features for environmental sounds is presented in [48]. Chachada and Kuo (2014) discussed discrete, frame-based audio classification where features can be extracted from windowed audio segments [48]. Since these sounds are arbitrarily time-varying, optimal windowing is often unfeasible. Gabor filter maps along with feed-forward neural network architectures successfully classified environmental sound sources.

One study demonstrated spectrogram derived features outperformed traditional MFCC mappings under windowed audio data [47]. The authors noted numerous potential features including statistical and spectral domains. Local features, or features in smaller windowed segments, contained more useful information than global spectral features due to noise and

temporal characteristics in environmental sounds. The study concluded that spectrograms retained useful information with minimal pre-processing for environmental sounds. Discussions concerning spectrogram windowing with higher classification accuracy stemming primarily from increased frequency resolution. The authors demonstrated that k-NN clustering could successfully classify environmental sound segments.

Specific concerns and justifications for using convolutional neural networks (CNN) in urban (environmental) sound classification were presented in [51]. The paper introduced a number of potential decisions and parameters required for spectrogram driven CNN as well as training data creation. Piczak, the author, briefly discussed the generation of training data of spectrograms of events; audio data split into overlapping segments could provide adequate information for training. Frequency invariance was adjusted based on kernel dimensionality.

Other domains have addressed similar challenges with isolating audio events in noisy spectrogram signals. Fanioudakis (2017) described challenges when determining bird sounds in larger spectrograms [38]. Deep neural architectures were used coupling spectrogram representations as images and existing image processing architectures (ImageNet, U-net). The authors rescaled and normalized spectrogram windows to adapt prior architectures. Using a user-defined dataset, the authors were able to determine binary event/null event within unlabeled spectrograms. The researchers introduced a novel bounding box validation method where overlapping bounding box areas to ground truth bounding boxes were compared. Using algorithms for localization, the authors managed to achieve roughly 67% accuracy when identifying bird vocalizations in larger spectrograms.

While deep neural network architectures have proven effective in classifying audio content, future challenges are noted in [11]. Classifying sound event literature often lacks practical implementation. Unstructured audio environments containing non-stationary noise, competing audio sources, and unforeseen challenges limit current effectiveness.

2.3.2 Damage Identification

CNN variants have been used to diagnose fault data in mechanical systems [22, 24, 52]. Both in vibration data, stemming from in-contact sensor placement, and acoustic emission sources, the studies demonstrate how system characteristics are determined through deep, convolutional architectures. CNNs with auto-encoding layers detected bearing faults [3, 4].

Lee et al. (2017) described a method of diagnosing semi-conductor process faults [43]. Fault visualization from multivariate sensor data was additionally discussed. The paper cites successful fault identification using a CNN auto encoder to learn complex representations. One study mentioned in [4] described a CNN model to classify four rotating machinery conditions using discrete Fourier transforms from accelerometer signals.

Another study successfully diagnosed rotating machinery faults [24]. The authors described limitations in understanding with deep-learning architectures – presenting new methods for visualizing features. A "t-distributed stochastic neighbor embedding" method was proposed to understand practical implications behind CNN kernels. Due to convolutional layer mapping from inputs to classifiers, these features are not fully understood. The spectral content, inherent in learned CNN kernels, tended to contain different frequency peak information among lower level convolutions.

2.4 Similarity Analysis

Other research domains have addressed challenges in distinguishing audio content between similar sources. Relevant to progressive damage, similarity metrics between gradually decaying segments are more difficult to capture. Rather than rapid changes in system dynamics often arising from impulse phenomena generated from catastrophic faults, temporal-feature relationships are not properly understood. The following studies suggest methods to understand similarity metrics in audio signatures in biologic and music identification fields.

A novel feature set for clustering non-stationary, frequency dependent signals was presented in [53]. The authors describe bird call recordings are often corrupted by wind, other

noise sources, as well as other bird species. The authors present a method to distinguish similar audio signatures with cross-correlating spectrograms. Singular value decomposition (SVD) based features were calculated as a metric for similarity and adequate enough for complex, non-stationary classification. Authors note that spectrogram similarity alone had a decrease in performance compared to raw data signals. Lower dimensionality feature vectors from SVD outperformed traditional features including MFCC.

Other methods introduced to visually identify similarities between different audio waveforms have been introduced [54, 55]. These works characterized audio segments through self-similarity analysis. MFCC features were extracted from partial spectrogram segments from the full-length waveform. Each feature vector from the partial segmentations were compared to each other segmentation – resulting in a cosine similarity heatmap. The authors then used the mapping as both a visual tool and similarity comparison between waveforms.

2.5 Clustering

Unsupervised approaches attempt to group data instances based on inherent patterns. Extracted features in audio content [56] may naturally cluster based on separability in high dimensional sub-spaces. Clustering methods, such as K-means, use distance metrics derived between a centroid, or a central region of data, and instances of data [57]. While the centroid locations are initially randomized, the number of centroids are predetermined.

The K-means clustering algorithm has key fundamental operations: 1) initialize centroid locations; 2) minimize centroid distances to partitioned data (Equation 2.1); 3) update centroid locations; and 4) repeat steps 2-3 until solution converges. The centroid-to-data distance can be defined by various metrics including the cosine similarity - or the angular distance between vector components, see Equation 2.2 [58, 59]. Using the angular metric is sometimes referred to as spherical k-means clustering [60]. Spherical clustering distance metrics are shown to outperform euclidean distances in high dimensional spaces [61]. A

detailed description of the spherical k-means algorithm is presented in [61].

$$\underset{x_c \in \mathcal{X}}{\operatorname{argmin}} D(\overline{x_i}, \overline{x_c}) \tag{2.1}$$

The angular distance, $D(\overline{x_i}, \overline{x_c})$, between the feature vector and the iteratively computed centroid clusters describes one minus the angle between the feature vector $\overline{x_i}$ and the centroid $\overline{x_c}$. The set of all possible centroids is X. Cluster membership is assigned based on minimum cluster distance; the entire process can be replicated which removes possibilities of local minima occurring during minimization [59].

$$D(\overline{x_i}, \overline{x_c}) = 1 - \frac{\overline{x_i} \cdot \overline{x_c}'}{\sqrt{(\overline{x_i} \cdot \overline{x_i}')(\overline{x_c} \cdot \overline{x_c}')}} \tag{2.2}$$

Certain metrics exist for validating data separability when clustering data with varying centroid amounts. A silhouette value $S(i)$, represented in Equation 2.3, describes inherent similarity between observations and their assigned cluster [62]. The variable $a(i)$ is the mean euclidean distance between observations (i) and an arbitrary cluster. The other variable $b(i)$ describes the minimum average euclidean distance between observations and all other clusters. The denominator argument $max(a_i, b_i)$ describes the silhouette score as a normalized similarity ratio. This score describes a metric to describe the relationship between cluster assignments as well as neighboring clusters [62]. A higher silhouette score suggests high similarity, approaching the value one, between observations and cluster assignments. As the silhouette ratio decreases, observations are equally as likely to be classified in other cluster assignments. This metric assumes relative spherical and separable clusters.

$$S(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{2.3}$$

Another metric used to determine optimal centroid (k) amounts involves computing the sum of squared error (SSE) between observation to centroid distances [63, 64]. The SSE

metric, outlined in Equation 2.4, estimates an optimal centroid determination by measuring differences in error between observations (x) in a cluster (k) and the centroid ($\tilde{y}_k$). The variable n describes the total number of centroids. Typically referred to as an 'elbow curve' or the 'elbow method,' the cluster tightness (SSE_k) rate decreases as more centroids are introduced for k-means clustering. A threshold for the SSE rate, indicated by a relative plateau in error reduction while more centroids are added, is noted as possible optimal k-centroid selection.

$$SSE_n = \sum_k \sum_{x \in k} ||x - \tilde{y_k}||^2, \ k = 1, 2, 3..., n \tag{2.4}$$

Bach et al. (2009) demonstrated that clustering algorithms could be applied to audio source separation problems [65]. The authors used a similarity matrix as a metric to define separability in speech sources within spectrograms. Spectral mixing problems can be addressed by clustering algorithms, however, the authors proposed a spectral clustering method can outperform, linear and separable k-means clustering approaches.

Autonomous classification of audio events using neural network architectures was also studied in [56]. While the published results are preliminary, a proposed audio source separation procedure using a combination of a deep network architecture and k-means clustering was employed. The study documents how unsupervised spectrogram features could be extracted from an encoding layer in a deep auto-encoder network and clustered to determine different audio sources. The two-fold approach suggested extracted spectrogram features allowed for separable classification.

There are limitations associated with clustering methods. Clustering procedures often lack physical intuition [3] partially do to high dimensional feature spaces. While generally regarded as an unsupervised classification method, clustering procedures often require priori assumptions for healthy and damaged states as well as labeled validation data.

2.6 Research Contribution

Overall, a general survey of prior work indicated that deep neural networks, particularly CNN variants, are viable and effective for segmenting acoustic signals. Furthermore, studies indicate that extracted features, present in both spectrogram and unmapped signal domains, contain viable information relevant for audio classification; however, the use of such approaches for remote health monitoring, particularly with respect to unsupervised learning approaches, is currently emerging in literature.

In this work, a semi-supervised analysis of robotic actuation signatures from acoustic recordings is presented. A two-fold methodology attempts to segment acoustic characteristics attributed to motion for eventual progressive damage identification and analysis in noisy, industrial environments. Convolutional neural networks are combined with unsupervised similarity analysis to quantify instances of robotic motions embedded within recordings of a manufacturing facility. The method attempts to capture deviations in system behaviour potentially relating to progressive, mechanical degradation. To develop and evaluate this methodology, recordings from a SCARA-series (Selective Compliance Assembly Robot Arm) (Figure 1.1) were leveraged for network training and testing, as well as for assessment of subsequent similarity analysis. These robots are widely used in electronics manufacturing for tasks such as material transitioning [1].

The study is organized into the following chapters: preliminary findings; an outlined methodology including data-set development; testing and validation of the proposed methodology; a discussion of results; and potential motivations for future work.

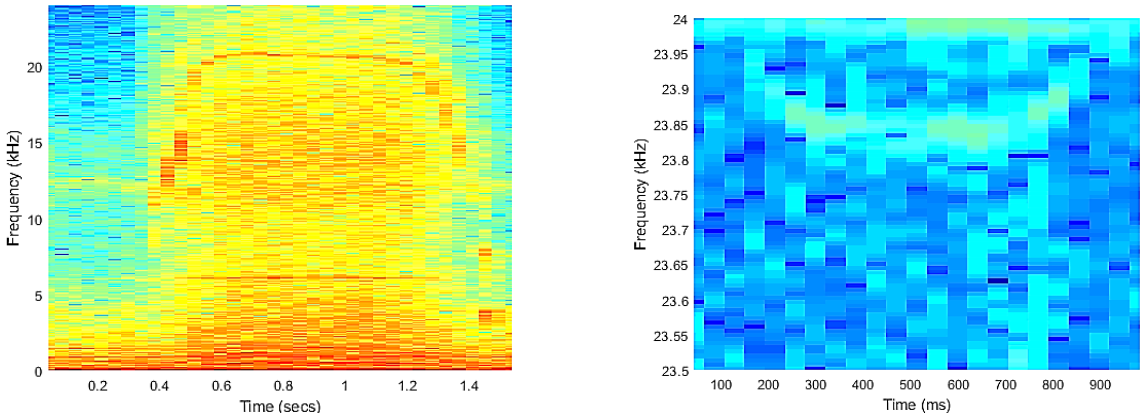
Chapter 3: Preliminary Findings

The behavior of the SCARA-series robot was used as a basis for evaluating a remote monitoring methodology. As previously mentioned, primary SCARA actuations can be generalized as a series of movements in Cartesian coordinates, as depicted in Figure 1.1. Limited acoustic emission and associated operator intuition of z-axis motion acoustics limited the study's scope to discern y- and x- axis actuations. Primary motions are further divided into subclasses based on various actuations along the same axis of travel. For example, base robot motion from location(s) A to B, A to C, and B to A would represent three distinct subclasses for a single primary class.

Visual analysis of primary actuation spectrograms describe uniqueness present in harmonic content. This suggests that each signature may be potentially distinguishable through supervised machine learning, however, occur at different frequency ranges and different variations in power. For instance, y-axis motions are visually observable in a band ranging from 0-20 kHz, see Figure 3.1a, while x-axis motions are observable from 23-24 kHz, see Figure 3.1b.

Analogous to the methodology described in proceeding sections, recordings were made of various a SCARA actuations, in situ, in a physical manufacturing bay. The manufacturing bay contained a single SCARA-type tool. Each motion class had a duration of approximately one second; however, the exact number of secondary actuation classes were not known. As a result, it was impractical to attempt to classify them through a supervised machine learning approach, given the lack of relevant labeled training data. The ability to distinguish these secondary classes through feature-space analysis of segmentations is presented later.

Simplistic techniques to isolate noise and actuation characteristic information through filtering was omitted. Based on possibility of unintentionally decreasing relevant acoustic



(a) Y-axis actuation spectrogram

(b) X-axis actuation spectrogram

Figure 3.1: Actuation spectrogram examples

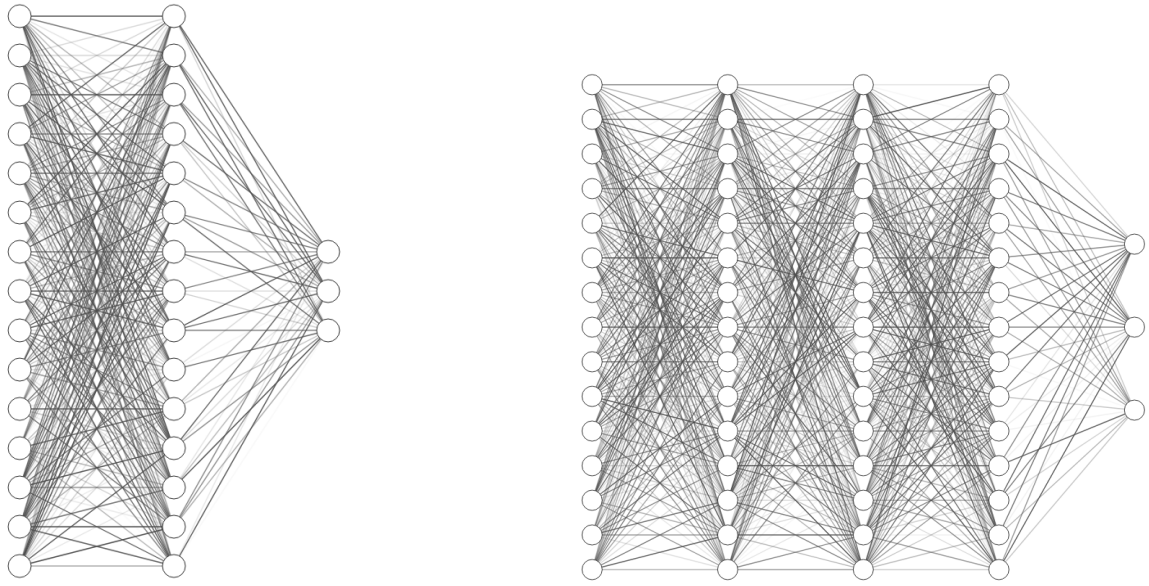
emission [6] and inability to correct for wide-spectrum environmental noise in preliminary experiments, no pre-processing steps were applied to acoustic signal waveforms. Furthermore, extensive signal filtering would bias resulting approaches to the specific data-set rather than a generalized methodology.

Preliminary work was conducted to investigate classification of primary SCARA motions through artificial neural network (ANN) models. Analogous to the procedures outlined in later chapters, segmented training data consisting of primary SCARA motions was labelled and inputted to varying ANN architectures. Two architectures were evaluated: a shallow network consisting of a single hidden layer (Figure 3.2a) and a deeper network comprised of three hidden layers (Figure 3.2b). The duration of primary actuations ($\sim$ 1 second) and the sampling frequency necessary to capture x-axis motions (48 kHz) lead to network inputs of 49153 amplitude values, impractical for such shallow ANN models. The input dimension was subsequently reduced through engineered feature extraction. Analogous to subsequent evaluations, 10% of the training data was withheld for validation.

Training and testing data was created correlating robotic motion with aligned video recordings and spectrogram visualizations of corresponding audio waveforms. Only primary motions were obtainable in scale where identifiable visual features in spectrograms, along

with operator authentication of associated features, were used as the basis of training data. A smaller subset of labeled subclass validation data, analogously created using aligned video and audio waveforms, was used for secondary analysis presented in subsequent sections.

A final training set of 692 audio recordings for each x-axis, y-axis, and noise instances were collected, with a waveform length of 49153 points. A validation set (10% of withheld training data) consisting of 77 segmentations in each class were used. To reduce potential overfitting from class size imbalances, motion classes were capped to the minimum number of segmentations recorded (y-axis motion) and sampled without replacement to reach the identical size. The process is analogously discussed in the subsequent chapters.



(a) Shallow ANN architecture (b) Deeper ANN architecture

Figure 3.2: Neural Network Architecture Classification Preliminary Attempts

A shallow network consisting of a single hidden layer (3.2a) and a deeper network comprised of multiple hidden layers (3.2b) was used for analysis. Fifteen neurons were used in

Predicted \ Actual	N	Y	X
N	80.52%	0%	19.48%
Y	0%	93.51%	6.49%
X	33.77%	0%	66.23%

(a) Shallow architecture confusion matrix (1 hidden layer)

Predicted \ Actual	N	Y	X
N	84.42%	0%	15.58%
Y	0%	94.81%	5.19%
X	25.97%	0%	74.02%

(b) Deeper architecture confusion matrix (3 hidden layers)

Figure 3.3: Confusion matrices for shallow/deep network classification

the initial neural network models which corresponded to an analogous number of input features. A confusion matrix was derived between predicted and classified primary actuation classes, depicted in Figure 3.3.

Lower energy actuations (x-axis) were consistently confused with noise states, see Figure 3.3a. Even with more dense architectures, the result was furthered with only a slight improvement possibly indicating overfitting rather than increased classification accuracy. These tests suggested a more sophisticated architecture was required due to complex relationships present in actuation signals and drastic variations between harmonic content. 2D Convolutional architectures were subsequently adopted as a basis for classification as demonstrated in subsequent sections.

Chapter 4: Methodology

A semi-supervised data analysis approach is developed to extract and classify acoustic signatures from a SCARA system. A dbx RTA-M microphone captured audio process data in a manufacturing bay containing a single SCARA-type robotic tool. Four-minute unlabeled, audio recordings containing SCARA actions spanning various semiconductor manufacturing processes are captured and used for all training data. An eleven-minute audio recording, aligned with corresponding video data, is used as final validation data since ground-truth labels are established based on operator authentication.

A convolutional neural network is designed and trained to perform initial segmentation of primary motion classes from an unlabeled spectrogram. Using normalized image-like spectrograms, see Figure 3.1, y-axis, x-axis, and noise actuation signatures are used for initial segmentation. However, due to limited available labels for subclasses, training data could only be divided into primary groups due to CNN's requiring large data-sets for learning relavant features. Since the actuation signatures share visually similar features in primary actuations, a methodology to create spectrogram-based training data for course segmentation is presented. The visual information present in spectrograms may allow complex, contextual harmonic features to be learned, more robustly against corrupting, non-stationary environmental noise for initial segmentation tests.

Due to the lack of subclass actuation data labels, subsequent unsupervised analysis attempts to group segmentation results into known subclasses from a smaller, ground truth dataset. Temporal, statistical, and spectral features, based on damage and audio classification literature are extracted from each primary segmentation. These features are normalized allowing for a standardized comparison due to different scales present. Principal component analysis subsequently reduces dimensionality for maximum variances inherent in derived feature sets. K-means clustering then separates primary robotic motions into a specified

number of clusters. SSE, silhouette, and visual observations are determined for a clustering validation metric. Cosine similarity analysis analogously attempts to find patterns between segmentations allowing for an alternative visualization of subclass similarity.

4.1 Dataset Development

Rather than direct analysis of the time-series waveforms, audio signals were transformed into spectrograms and treated as pseudo-images. Normalized spectrogram intensities at specific time-frequency locations allowed for an analog to pixel representations in an image. This image-like data format enabled representation of events with a diverse range of spatial features, such as harmonic edges (see crest features in 3.1a).

A training data-set was created using collected audio recordings in conjunction with operator authentication. Fixed signal length spectrograms were labeled as x-axis, y-axis, and idle/noise motions as depicted in Figure 3.1. Indicators for each motion class, based on visual observation, guided training spectrogram derivation. Events containing neither visual indicator are designated as noise.

As previously stated, data-set creation for SCARA acoustic data held certain limitations. While recordings were assumed to span all potential subclasses, subclass imbalances present in each primary motion were possible. Second, visual indicators of lower energy motions (z-axis) were not observable in spectrogram representations and were therefore omitted in the training set. Lastly, ground truth data for motion subclasses was not readily available for fine-grained, CNN training segmentations.

Spectrograms were generated using a Blackman-Harris window of 4096 waveform data points, overlap of 2048 data points, and 8096 Discrete Fourier Transform (DFT) points. Input spectrograms were fixed with 4097 (height) x 25 (width) dimensions, with regards to the resulting binned spectrogram intensities. These dimensions stemmed from understanding of the smallest known event size (1.1 seconds). Due to the fixed length of the spectrogram input and variability between subclass y-axis motion duration, some training examples were unintentionally split into overlapping sections. While introducing redundant

information potentially biases network training by overfitting, several other benefits may be introduced. The variation in training set may outweigh bias introduced in redundant overlapping sections, as mentioned in [51]. Moreover, the CNN will realistically encounter similar, partially obscured spectrograms during segmentation in operation.

The final training set consisted of 1360, 769, 1181 training examples of noise, y-axis, and x-axis motions, respectively. Training example sizes were kept constant. For example, training a 3-class classifier CNN between y-axis, x-axis, and noise used 769 examples as the maximum training set size; the other classes were randomly sampled, without replacement, to match the minimum training set amount. Keeping the training data count consistent further prevented overfitting bias. A smaller dataset containing labeled motion subclasses was used to evaluate the unsupervised similarity aspect of the overall approach. This video aligned data consisted of 78 y-axis motions and 194 x-axis motions. Y-axis actuation subclasses with labels 1(a), 1(b), 2, 3(a), 3(b), 4, and 5 had the following quantities of 19, 10, 10, 21, 7, and 6, respectively. X-axis actuation subclasses with labels 1, 2, 3, 4, 5, 6, 7, and 8 had the following quantities of 49, 49, 19, 19, 21, 21, 8, and 8, respectively.

Spectrogram normalization was first used to limit spectrogram variability. Across 3000+ test spectrograms, minimum and maximum temporal-frequency bins values were calculated and used as boundaries. All training and test spectrogram intensities were subsequently re-scaled from a minimum bound of -14.68 and maximum bound of 5.89 to values between 0 – 255, making the spectrograms a more image-like data type and enabling a wider range of network activation functions. A similar mapping procedure was successfully implemented in [11].

4.2 Supervised Actuation Detection

To the authors' knowledge no established CNN architecture exists for spectrogram training as reaffirmed in previously mentioned studies. Moreover, due to domain-specific requirements, other spectrogram specific architectures were unsuitable for direct application. Due to the high resolution and frequency dependent spacing in training spectrograms, an

empirical study was conducted to evaluate general trends among candidate CNN architectures. A hyperparameter sensitivity study addressing convolutional kernel dimensions was performed; however, an exhaustive parameter search for an optimal architecture was considered outside the scope of the current work and requires future study.

Three architectures were used to understand whether varying convolutional kernel dimensions had measurable impacts on primary spectrogram segmentation. The architectures chosen were primarily based on similar construction including hidden layer dimensions and kernel sizes; however, a comprehensive parameter study for determining optimal CNN architecture sizes were considered outside the scope of the current work and requires future study. Specifically, general trends in symmetric and asymmetric filter sizes for highly rectangular spectrograms were explored. Padding, kernel, and layer dimensions were determined based on reducing dimensionality to a minimized, 1-D, fully connected softmax layer. Each architecture was of comprised of six hidden, convolutional layers. While literature studies vary on optimal network size using CNNs for spectrogram analysis, 4-6 stacked hidden layers were common in spectrogram based classification exampled in [66–71].

Network parameters held constant after empirical evaluation are listed here. The training rate for stochastic gradient descent with momentum (SGDM) was held to 0.001. The maximum epoch number was held at 1000 while minibatch size was kept constant at 64 samples. Max pooling layers were intentionally omitted with all tested architectures. While the down-sampling procedure provided feature translational invariance, the reduction in parameters also decreased the potentially learn-able feature space and degraded performance. These behaviors were discussed in [72–75]. To reduce the computation time of segmentation, a stride corresponding to 5 temporal-frequency bins, or 0.264 seconds, was specified. While a CNN may benefit from over-fitting prevention, such as dropout, these steps were omitted for simplicity; as stated in the future work section, such augmentations require a separate and thorough study.The complete list of CNN parameters is presented in Tables 4.1,4.2, and 4.3. An illustration of an example architecture (Architecture #3) is provided in Figure 4.1.

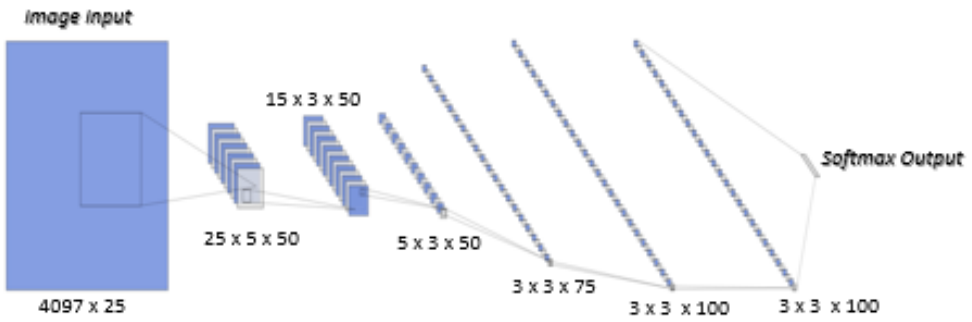


Figure 4.1: Example of CNN convolutional layers in architecture (#3). Generated using the resource presented in [2]

4.2.1 Architecture #1

The first tested architecture was based on CNNs developed for image processing, with square kernels applied to the asymmetric spectrogram inputs [38, 46]. Smaller initial kernel sizes (5x5) were used to understand the effects of rectangular kernel dimensions. While smaller than ImageNet (11x11) and GoogleNet (7x7) first layer kernel dimensions [76], the architecture was generally representative of smaller, local features embedded in the spectrogram's time-spectral relationship.

4.2.2 Architecture #2

For the second architecture, a 9x7 input kernel was chosen due to the minimum spacing between features in an observed "cupping" phenomena present in x-axis motions (Figure 3.1b). This was considered the smallest asymmetric kernel dimension to completely capture the visual signature in training data.

Table 4.1: CNN Architecture # 1

Layer	Type	Dimension	Depth	Stride (h,w)	Padding (h,w)
1	Input	4097x25	-	-	-
2	Conv	5x5	50	[3,1]	[4,2]
3	ReLU	-	-	-	-
4	Conv	5x5	50	[2,1]	[3,1]
5	ReLU	-	-	-	-
6	Conv	5x5	50	[2,2]	[2,1]
7	ReLU	-	-	-	-
8	Conv	5x5	75	[3,2]	[1,1]
9	ReLU	-	-	-	-
10	Conv	3x3	100	[3,2]	[0,0]
11	ReLU	-	-	-	-
12	Conv	3x3	100	[1,1]	[0,0]
13	ReLU	-	-	-	-
14	Fully Connected	-	-	-	-
15	Softmax	-	-	-	-
16	Classification	-	-	-	-

4.2.3 Architecture #3

This architecture demonstrates the highest asymmetry tested. An initial 25x5 kernel dimension with large vertical strides attempted to capture more frequency dependence among temporal-frequency bins. However, the horizontal component of kernel filters should capture some temporal harmonic content.

In order to evaluate effectiveness of each architecture and potential bias to underlying testing and training data, k-fold validation is conducted. Five partitions of data are used, where four groups are used for CNN training and one group is withheld for evaluation. These five groups consist of 153 samples for each class (totalling 459) randomly sampled without replacement from 765 samples of the initial dataset. This process is repeated five times, for each architecture, so that sensitivity to underlying data can be estimated [77]. Averaging correct and incorrect classifications between noise, y-axis, and x-axis motions can better indicate architecture performance for comparison.

Table 4.2: CNN Architecture # 2

Layer	Type	Dimension	Depth	Stride (h,w)	Padding (h,w)
1	Input	4097x25	-	-	-
2	Conv	11x9	50	[2,1]	[3,4]
3	ReLU	-	-	-	-
4	Conv	9x7	50	[2,1]	[1,3]
5	ReLU	-	-	-	-
6	Conv	7x5	50	[2,2]	[1,2]
7	ReLU	-	-	-	-
8	Conv	5x3	75	[2,2]	[0,1]
9	ReLU	-	-	-	-
10	Conv	3x3	100	[2,3]	[1,1]
11	ReLU	-	-	-	-
12	Conv	3x3	100	[2,3]	[0,0]
13	ReLU	-	-	-	-
14	Fully Connected	-	-	-	-
15	Softmax	-	-	-	-
16	Classification	-	-	-	-

4.2.4 Segmentation and Validation

Once trained, the CNN would be tested on withheld training data and finally on the aforementioned ground-truth, labelled spectrogram. To match training data normalization, the new spectrogram data would be normalized similarly to training spectrograms. The highest outputted softmax probability of the softmax network layer provided the basis for classification.

Overlapping segmentations shared between classes would become potentially problematic. To handle class bounding box overlap between x- and y- actuations, any overlapping boundary times were averaged. While a simplified approach, the method allowed for a separation of independent segmentations. Other bounding box methods such as fuzzy classification are possible and are a potential avenue for future work. An empirical classification method based on bounding box area, analogous to the methodology presented in [38], was used for validation. The architecture(s) with the highest empirical classification accuracy

Table 4.3: CNN Architecture # 3

Layer	Type	Dimension	Depth	Stride (h,w)	Padding (h,w)
1	Input	4097x25	-	-	-
2	Conv	25x5	50	[2,1]	[4,2]
3	ReLU	-	-	-	-
4	Conv	15x3	50	[2,1]	[3,1]
5	ReLU	-	-	-	-
6	Conv	5x3	50	[2,2]	[2,1]
7	ReLU	-	-	-	-
8	Conv	3x3	75	[2,2]	[1,1]
9	ReLU	-	-	-	-
10	Conv	3x3	100	[3,2]	[0,0]
11	ReLU	-	-	-	-
12	Conv	3x3	100	[2,1]	[0,0]
13	ReLU	-	-	-	-
14	Fully Connected	-	-	-	-
15	Softmax	-	-	-	-
16	Classification	-	-	-	-

would be used for remaining unsupervised analysis.

4.3 Unsupervised Analysis

4.3.1 Feature Engineering

Once segmented into primary motion classes via CNN, the corresponding audio waveforms are further discriminated through unsupervised analysis via feature extraction. Derived from corresponding audio waveforms with varying lengths, features are calculated from time, statistical, and spectral domains. These features include peak amplitude, average amplitude, mean square, root-mean square, zero-crossing rate, variance, standard deviation, kurtosis, crest factor, skewness, and k-factor. The feature list is presented in Table 4.4 while equations are displayed in Table A.1 in the Appendix.

Frequency data between 4 kHz – 24 kHz was empirically known to contain the most useful discriminating, harmonic information between actuations based on operator intuition;

however, comparing peak frequency content became immediately problematic due to motion complexity and diffuse noise concerns. As shown in Figure 4.2a, relevant peak finding in the presence of wideband noise is almost impossible due to the variation in spectral content between subclass samples. While optimal filtering techniques would require further, comprehensive study, a widely adopted acoustic smoothing technique was implemented [26, 27]. Savitzky-Golay filtering [25], with a 3rd order model and 101 regressive points, was used to filter spectral content. While not considered an optimal filtering strategy, the general implementation of spectral smoothing yielded improvements to understanding and estimating actuation spectral content, as shown in Figure 4.3.

Peak estimates were difficult to directly compare due to spectral distribution even after filtering. As shown in Figure 4.3, even smoothed frequency responses demonstrated visual indicators with differing peak content including peak locations, magnitudes, width, prominence and quantity. Typical peak finding procedures were insufficient to capture seemingly discriminate frequency content through energy signatures. After numerous attempts, peak amplitude(s) features were intentionally omitted. Spectral moments were subsequently used as a metric for evaluation and considered more robust despite corruptive noise. Analogous to the statistical features from the time domain response, kurtosis, crest factor, k-factor, and skewness were applied on the FFT response data between 4 kHz – 24 kHz. These new features described statistical moments in the spectral domain. The procedure demonstrated filtered spectral statistics could retain observable, discriminating information between subclasses, as presented in Figure 4.4. A final concatenated feature set is shown in Table 4.4.

These feature sets were normalized, using max-min normalization, to remove potential bias arising from features with varying scales in subsequent analysis [78,79]. Principal components were computed to find features of maximum variance for clustering and similarity analysis. Principal components were chosen to represent at least 95% of the variability present in the feature set. The equation for calculating the required principal component number is presented in equation 4.1. Sigma σ_i represents a principal component while N

Table 4.4: Unsupervised Features

Temporal & Statistical Features	Spectral (FFT) Features
Kurtosis	Kurtosis
Skewness	Skewness
Crest Factor	Crest Factor
K-Factor	K-Factor
Mean	-
Variance	-
Standard Deviation	-
Mean Square	-
RMS	-
Peak Amplitude	-
Zero-crossing rate	-

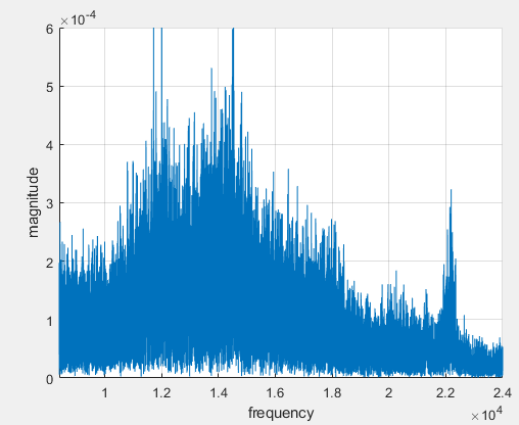
represents a predetermined, satisfactory threshold.

$$N_{threshold} = \frac{\Sigma_{i=1}^{k} \sigma_i}{\Sigma_{i=1}^{n} \sigma_i} \tag{4.1}$$

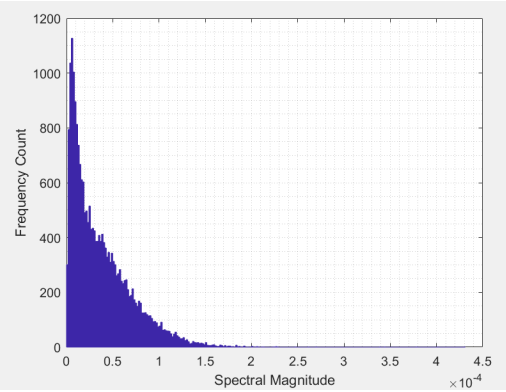
4.3.2 Similarity and Clustering Analysis

After feature extraction, unsupervised methods are evaluated for assessing similarity between segmented subclasses. Cosine similarity, or the inner product between subclass feature spaces, is calculated and compared against known subclasses. Seven y-axis and eight x-axis actuation subclasses were observed in the labelled ground-truth data-set. Calculating similarity metrics between each segmentation is presented in Equation 2. Theta, $\theta_{\overline{\sigma_i},\overline{\sigma_j}}$, describes the angle between feature space vectors from segmentations i and j, respectively. Plotting these cosine-similarities allow for a heat-map visualization between feature vectors from each segmentation.

$$\theta_{\overline{\sigma_i},\overline{\sigma_j}} = arccos(\frac{\overline{\sigma_i} \cdot \overline{\sigma_j}}{||\overline{\sigma_i}||\ ||\overline{\sigma_j}||}) \tag{4.2}$$



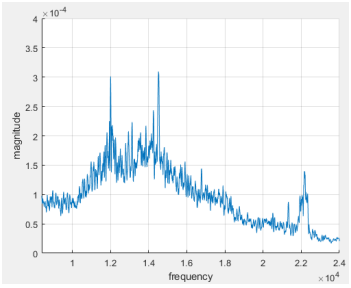
(a) FFT Response (class 1(a))



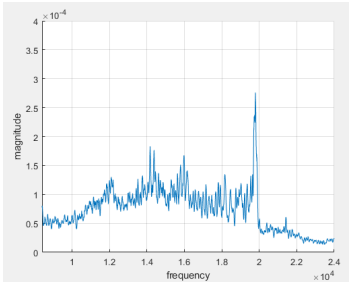
(b) FFT Response - Spectral Distribution (class 1(a))

Figure 4.2: Frequency Response and Spectral Distribution (No Smoothing)

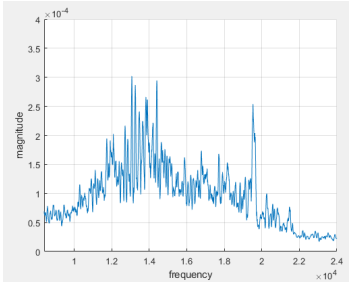
Principal component plots are used as a visualization tool to assess the validity of clustering results. K-means clustering was used to further delineate primary motion segmentations based on the extracted feature set. While other works discuss the k-means algorithm in depth [80–82], the clustering procedure attempts to minimize distances between translatable centroid locations for a set of feature vectors describing actuation subclasses. This distance is computed as an angular distance between actuation feature vectors from Equation 2.2. Minimizing the distances between centroid(s) and features from actuations determines cluster membership to each feature-set. SSE and silhouette graphs assess clustering validity.



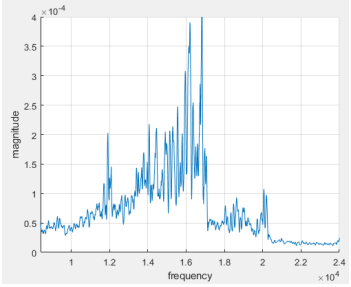
(a) Class 1(a) Filtered Spectral Response



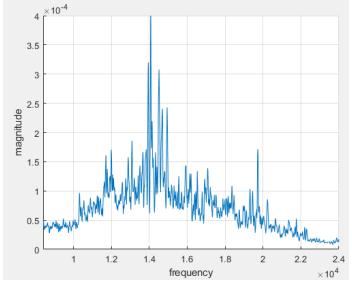
(b) Class 2 Filtered Spectral Response



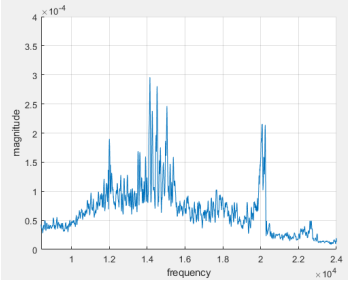
(c) Class 3(a) Filtered Spectral Response



(d) Class 3(b) Filtered Spectral Response

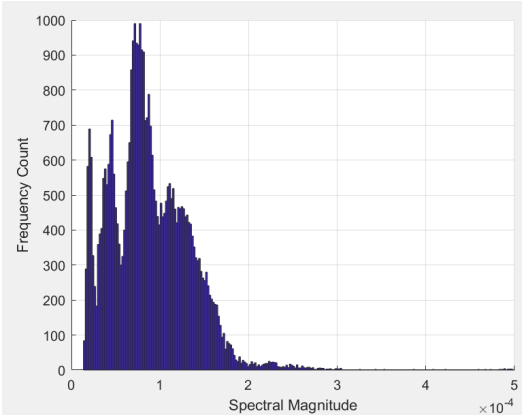


(e) Class 4 Filtered Spectral Response

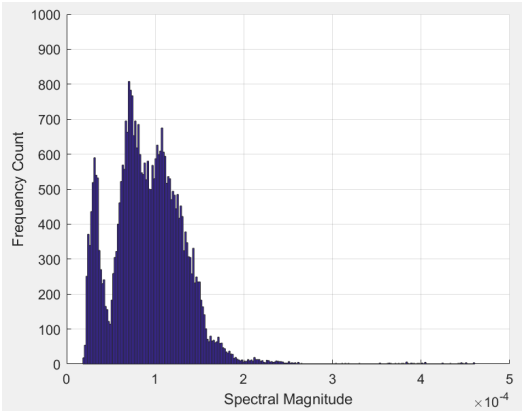


(f) Class 5 Filtered Spectral Response

Figure 4.3: FFT and associated spectral responses for y-axis subclasses



(a) Class 1(a) Spectral Distribution



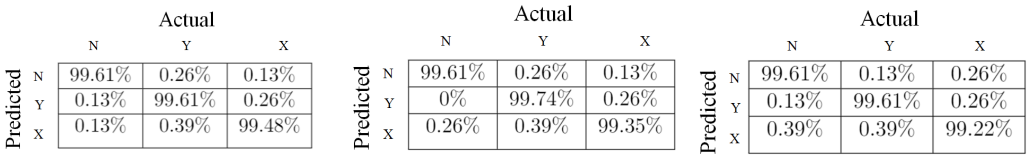
(b) Class 1(b) Spectral Distribution

Figure 4.4: Spectral distribution comparison between y-axis subclass 1(a) and 1(b)

Chapter 5: Results

5.1 Supervised Classification

To test effectiveness of each architecture, k-fold cross validation was performed with five partitions of 153 samples per class. These results were averaged and displayed in Figure 5.1. Notably, architectures had nearly identical performance with high classification between all actuation groups. This result could suggest over-fitting bias from limited feature variability within training examples. While additive noise within training classes provided some temporal-frequency variability, limited spatial variance from spatial edges may have unintentionally biased architecture performance. This result, however, indicates that the architectures were not sensitive to withheld training set groups.



Predicted \ Actual	N	Y	X
N	99.61%	0.26%	0.13%
Y	0.13%	99.61%	0.26%
X	0.13%	0.39%	99.48%

(a) Architecture #1

Predicted \ Actual	N	Y	X
N	99.61%	0.26%	0.13%
Y	0%	99.74%	0.26%
X	0.26%	0.39%	99.35%

(b) Architecture #2

Predicted \ Actual	N	Y	X
N	99.61%	0.13%	0.26%
Y	0.13%	99.61%	0.26%
X	0.39%	0.39%	99.22%

(c) Architecture #3

Figure 5.1: Validation of Architectures - k-fold Cross Validation

Due to the sliding window nature of the spectrogram inputs, a classification accuracy metric based on bounding box area was employed, based on the work in [38]. Overlapping false-positive and false-negative areas were calculated from differences between labeled actuation locations in a validation spectrogram and CNN classification results. Waveform

segments shared by known actuation duration and classification were recorded as overlapping area percentage. Actuation data not captured by classification resulted in a false-negative area percentage. Audio segments incorrectly classified as actuations resulted in a false-positive area percentage. The results from each architecture was presented in Tables 5.1 and 5.2.

Table 5.1: Validation on Ground Truth Spectrogram (y)

Architecture	Overlapping Area	False Positive Area	False Negative Area
1	67.8%	1.4%	6.4%
2	55.3%	2.5%	8.7%
3	87.1%	1.2%	2.6%

Table 5.2: Validation on Ground Truth Spectrogram (x)

Architecture	Overlapping Area	False Positive Area	False Negative Area
1	67.3%	12.0%	12.3%
2	91.4%	20.2%	3.5%
3	84.2%	14.6%	6.3%

Tables 5.1 and 5.2 highlight architecture differences discerning x-axis and y-axis motions. Architecture #3 demonstrated the highest overlapping area percentage in y-axis classification, while architecture #2 had the lowest classification accuracy for y-axis motions. Architecture #2 falsely classified 8.7% percent of spectrogram data as y-axis motions while missing nearly 45% of known actuation duration. Architecture #3 additionally held the lowest false positive and false negative area during y-axis classification. Architecture #1 did not outperform other architectures. Results for x-axis motions differed. While architecture #1 was still clearly inferior, architecture #2 showed the highest accuracy in capturing x-axis actuations. However, the increase in accuracy was accompanied by the highest false

positive rate.

The results suggest that classification accuracy is somewhat dependent on kernel dimensions. Highly asymmetric spectrogram input kernels demonstrated the best compromise in accuracy for y and x-axis classification. The result in Table 5.1 suggests that harmonic complexity in y-axis motions was better captured with skewed kernels with dominating frequency content. Smaller kernel dimensions demonstrated a lower classification accuracy with respect to discriminating known y-axis actuations. However, this trend is not followed with x-axis actuations - harmonic content may be captured with smaller kernel dimensions.

Figure 5.2 describes the difficulty in delineating feature edges with bounding box methods. While the method cleanly identified 87.1% of y-axis motions (architecture #3), y-axis segmentations corrupted by noise artifacts degraded classification accuracy, particularly near the boundaries of actuation transitions otherwise described as between manipulator movement and idle states.

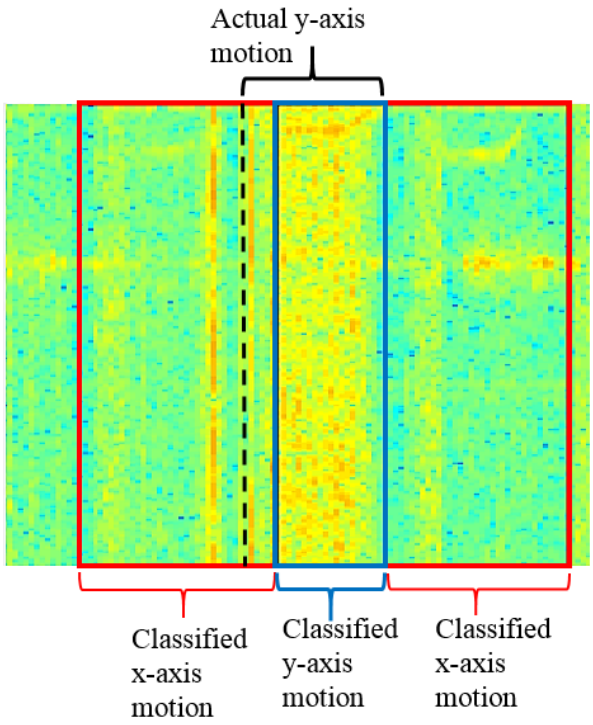


Figure 5.2: Poor classification around actuation transitions with wideband noise

The trained CNN architectures had difficulty distinguishing differences between x-axis motions from y-axis motions, especially during event transitions. This may be due to several phenomena including the similar feature-space stemming from both events. This may have led to segmentation bias during y-x transitions; spectrograms with overlapping x/y transitions were biased towards probability of y-axis motions (architecture #3) or towards x-axis motions (architecture #2). Sharp, wideband noise arising from neighboring mechanical phenomena during production processes, often seen as spikes in spectrograms (see 5.3) were captured in both x- and y-axis segmentations. These artifacts were observed to influence classification including false positive percentages in each segmentation.

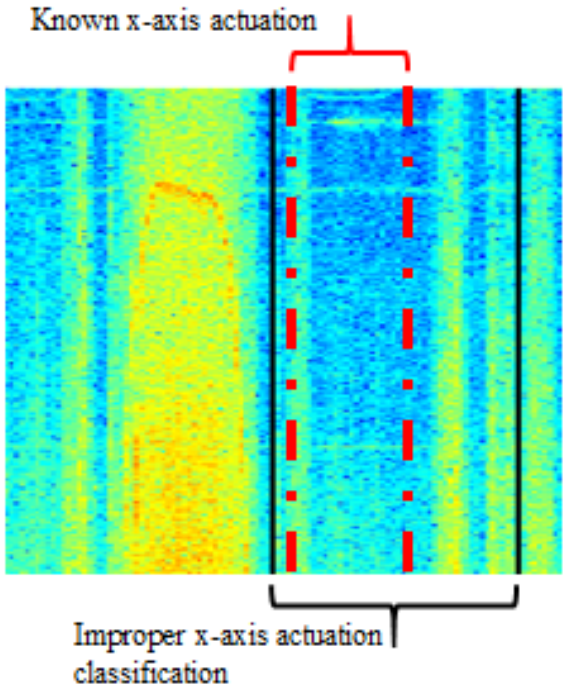


Figure 5.3: Wideband noise effects on x-axis classification accuracy

The results suggest a supervised approach to actuation classification from acoustic signals is feasible, given sufficient training data and accounting for the necessary asymmetry

of the convolutional layers. subclass actuations, comprising each course training set, should additionally contain similar spectrogram features, such as spectral edges, for a generalized parent class. Moreover, these primary class features should be distinctive for multi-class classification, i.e. y-axis motions should be harmonically distinct to x-axis motions as well as noise. These assumptions demonstrated relative segmentation success in course labeled actuation data.

An exhaustive hyper-parameter search to determine optimal convolutional kernels and network topology would increase classification accuracy. Unforeseen noise artifacts and other factors not experienced in the CNN training set may have additionally contributed to an architecture not fully representative of SCARA subclasses. Furthermore, deficiencies in subclass quantities may have unintentionally biased accuracy towards dominant sub-groups in training data.

Due to its more consistent performance, CNN architecture #3 was used for the analysis of the unsupervised methodology. The trade-off in x-axis classification overlap was acceptable given the higher y-axis classification accuracy along with lower false positive and false negative percentages among certain classification tests.

5.2 Unsupervised Clustering

5.2.1 Y-axis Similarity Analysis

Cosine similarity analysis was conducted between each segmentation for y-axis motions, illustrated as a heat-map visualization of similarity in Figure 5.4. A numeric score of zero indicates perfect similarity, exampled by the shaded blue region, describing a segmentation's similarity with itself. As dissimilarity between segmentations increases, the numeric similarity score increases towards an increasing red-shaded region.

An evaluation of this similarity yields several insights. Actuation subclasses 1(a) and 1(b) are mechanically similar and consequently share high similarity scores. These subclasses are additionally dissimilar to nearly all other actuation groups. Actuation subclass

2 is relatively similar to other motions in its category; however, it shares some similarity with subclass 4. Subclass actuation 5 is distinctly dissimilar from the other actuation subclasses. Subclass 3(a) and 3(b) are not clearly distinguished through cosine similarity despite having mechanically similar operation. Notably, several outliers are present in each subclass, potentially due to incorrect segmentations or noise-corrupted results.

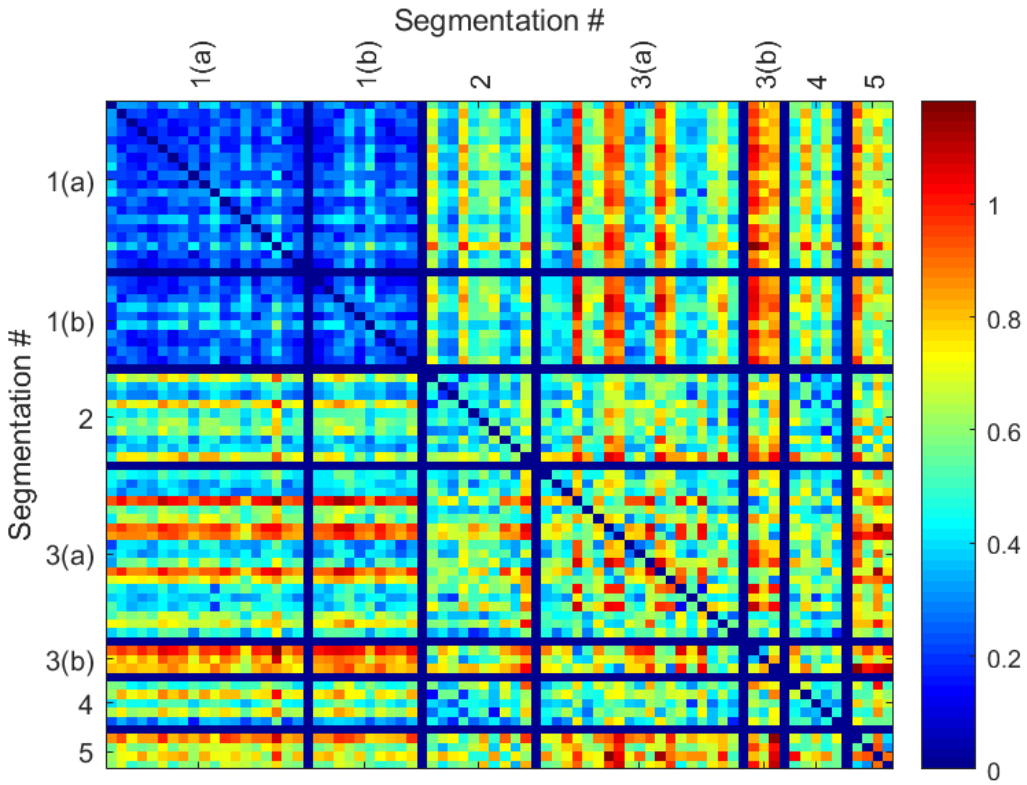


Figure 5.4: Y-axis cosine similarity

Figure 5.4 illustrates several notable trends within SCARA audio data. Firstly, some classes are relatively inseparable, such as groups 1(a) and 1(b), possibly indicating applied featureset may not fully distinguish differences between each class. Slight variations in

manipulator travel corresponding to separate sub-processes, labelled by manufacturing operators, may not have distinguishing characteristics warranting sub-process classification with given features. This result suggests that user-labels corresponding to sub-processes are coupled with feature fidelity (either manual or automatic); less noise with more distinguishing features allows more narrow classification between primary classes.

5.2.2 Y-axis Clustering Analysis

After similarity analysis, k-means clustering was employed to autonomously separate primary classes. Principal component analysis was conducted on normalized segmentation features before applying k-means clustering. Six principal components comprised 96.2% variability in the feature set. The first and second principal components were subsequently plotted with respect to their corresponding subclass labels in Figure 5.5. A secondary visualization adding the 3rd principal component is displayed in Figure 5.6. These results would be additionally compared to the heat-map presented in Figure 5.4.

From initial visual inspection, certain groups were naturally clustered with their respective segmentation labels as described in Figure 5.5. Firstly, actuations 1(a) and 1(b) shared a similar, inherent cluster. Occupying the rightmost portion of the principal component graph, the cluster demonstrated some level of separateness with class 1(a) holding a slightly higher 2nd principal component score than class 1(b). Classes accounting for mechanically similar operation may therefore have a characteristic feature set; while some observable separability was shown, further refinement in features may increase sparsity between subclasses. Actuation 5 was observably separable from other subclasses. Exampled by its relative cluster location, its principal components tended to cluster at bounds from subclasses 1-4. Other subclasses such as 2, 3(a), and 4 share less definable feature spaces primariyl stemming from variablity in class 3(b).

Figures 5.6a and 5.6b show an alternative, 3-Dimensional representation to Figure 5.5. These plots depict a 3D scatter plot, with the 3rd highest principal component, added for spatial depth. Colored polyhedrons are bounded at vertices corresponding to the motion

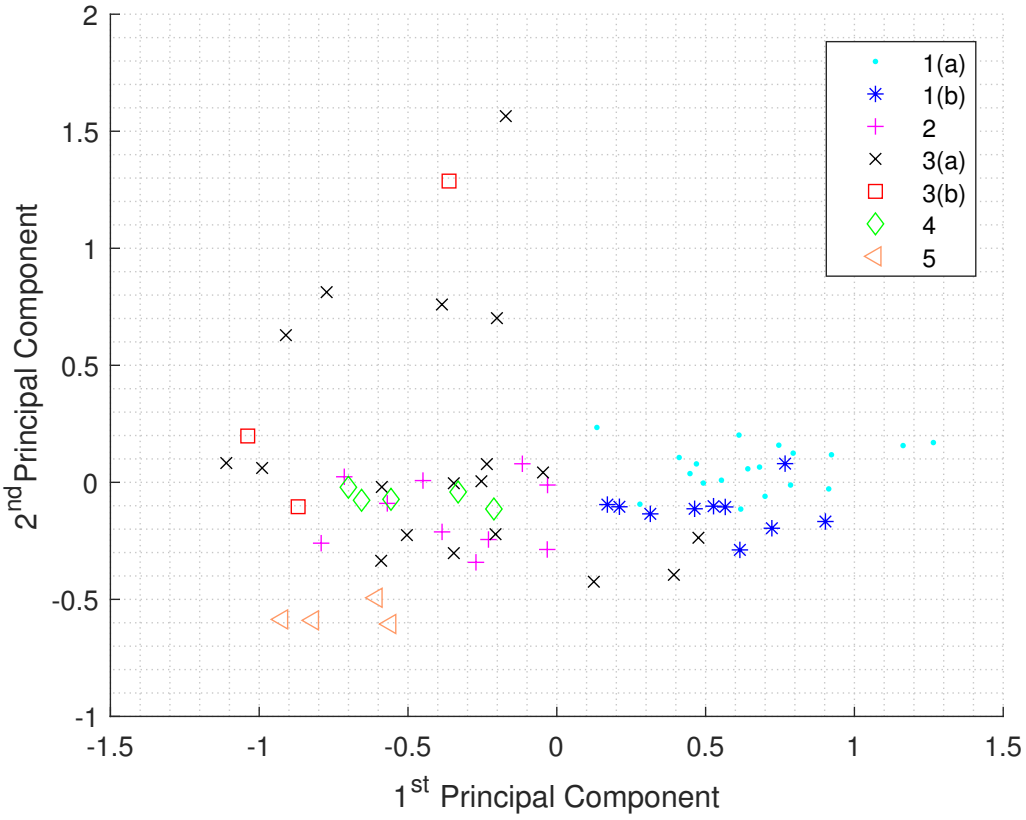


Figure 5.5: Y-axis principal component scatter plot (2D)

sub-classes. Polyhedron face colors additionally match sub-class marker colors pertaining to motion labels. Figures 5.6a and 5.6b are identical, however, offer rotated perspectives.

Subclass 3(a) has the highest observable feature space variance; other sub-class motion relationships may be potentially obscured. To demonstrate this effect, Figures 5.7a and 5.7b describe identical visualizations omitting the 3(a) class polyhedron region. Immediately, sub-class regions are better isolated from other groups. While regions containing 3(a), 1(a)/1(b), 2/4, and 5 are better distinguished without the 3(a) boundary, classes 2 and 4 remain visually coupled.

Figures 5.8, 5.6, and 5.10 illustrate k-means clustering attempts versus known labels

using varied centroid quantities. While perfectly separated clusters were not observed in 3-centroid or 7-centroid k-means attempts, certain trends became evident. Firstly, actuations 1(a) and 1(b) shared a dominant cluster with a few outliers contributed from class 3(a) in 3-centroid clustering, as depicted in Figure 5.8. A second cluster primarily grouped subclass actuations 2-5 while a third cluster captured actuations sparsely populating the highest, 2nd principal component ranges. These actuation feature vectors, clustered in the 3rd cluster, may have contributed to potential outliers from improper segmentation as well as corruptive noise.

A 3D visualization of clustering assignments with three centroids, expanding on Figure 5.8, is portrayed in Figures 5.9a and 5.9b with rotated perspectives. Cluster assignments are outlined by the colored polyhedron regions, while the ground-truth labels for each subclass are indicated by marker color and style. These figures further demonstrate that outliers and variability in groups 3(a) and 3(b) influenced cluster assignment as visually demonstrated by black and red-shaded polyhedrons.

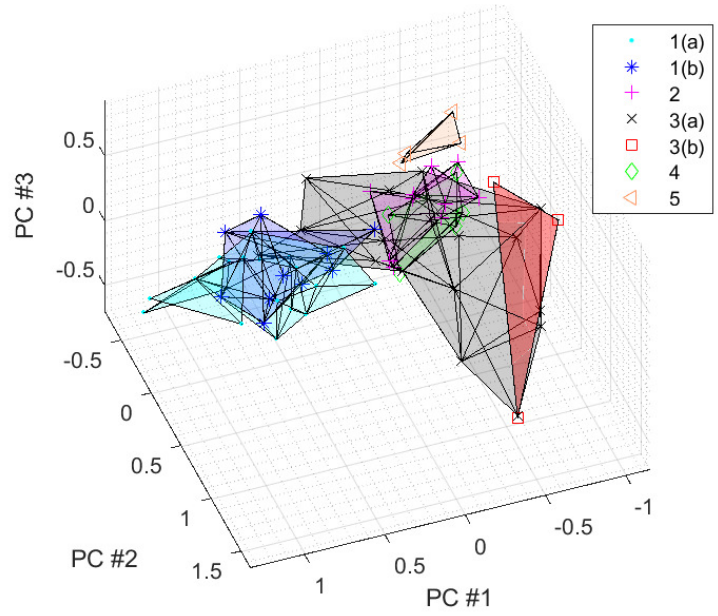
Increasing the centroid quantity to the number of known actuation subclasses (7), did not visually yield improvements, as shown in Figure 5.10. For visual clarity, 3D visualizations were omitted. By introducing more centroids, subclasses became over-segmented rather than identifying more secondary motions; however, some results matched three-centroid k-means results. The majority of subclasses 1(a) and 1(b) remained classified as a single grouping. Moreover, a centroid observably tended to group subclass 5, with additional outliers from other subclasses. Another cluster, noted by the black marker color, visually grouped feature space outliers from other motions. The cluster suggests that these points may have negatively affected clustering accuracy at lower centroid designation due to their spatial differences from other classes.

Clustering validations for y-axis subclasses are presented in SSE and silhouette score graphs, presented in Figures 5.11 and 5.12. The SSE rate observably decreases after four centroids are introduced, as depicted in Figure 5.11. This trend somewhat mimics the highest silhouette scores between two, three, and four centroids, shown in Figure 5.12. The

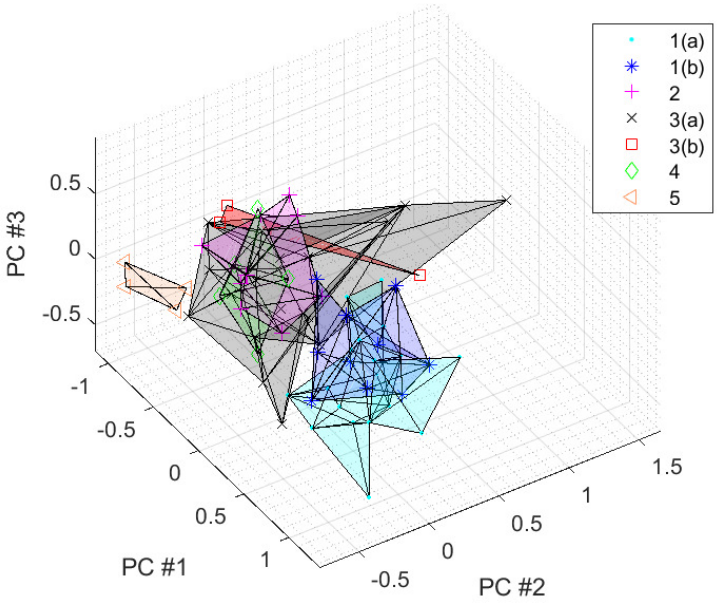
highest silhouette score is 0.6 which corresponds with two centroids. This result suggests that two clusters contain the highest similarity within observations in their respective cluster and could be optimal 'k' designation.

The silhouette score reflects visual observations in principal component plots, such as in Figures 5.9a and 5.9b. Two dominant clusters are seemingly formed: a cluster incorporating subclass 1(a) and 1(b) and a cluster incorporating other subclasses. Potential feature space outliers as well as limited separation between classes may have decreased the silhouette ratio.

Analogous to mixed accuracy in similarity analysis, inseparability in subclass segmentation suggests other contributing factors. Noise artifacts may have contributed to potential feature set outliers. Both PCA and k-means are sensitive to noise, stemming from calculated features, suggesting a more robust algorithm could better reject variations present in segmentation. Sparcity not present in overlapping feature spaces suggests k-means could not adequately delineate subclass separation effectively. Other features as well as fuzzier classification attempts may increase unsupervised, subclass segmentation; however, the method showed promise when identifying certain key actuation sub-classes.

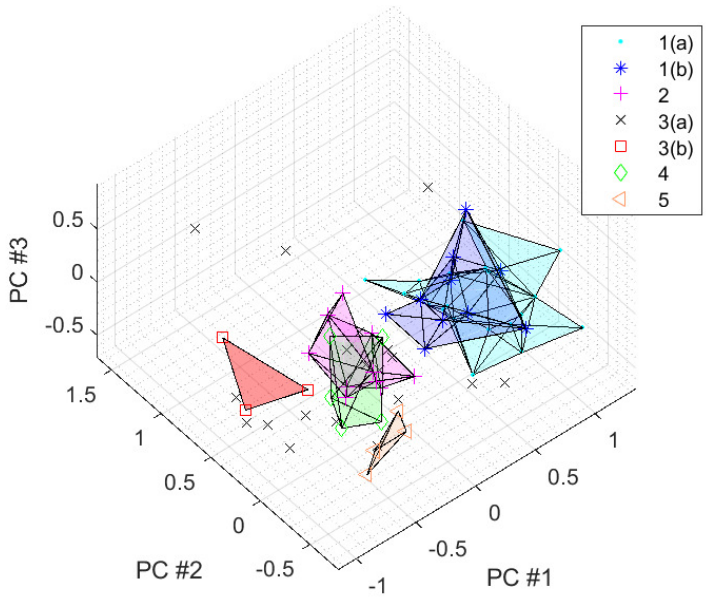


(a) 3D PCA visualization: Perspective #1

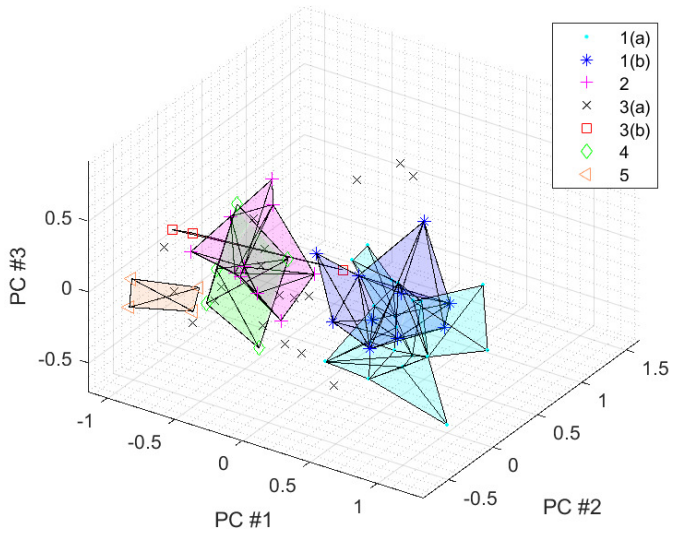


(b) 3D PCA visualization: Perspective #2

Figure 5.6: Principal component scatter plot (3D) overlaid with polyhedron clusters



(a) 3D PCA visualization (removal of 3(a) class): Perspective #1



(b) 3D PCA visualization (removal of 3(a) class): Perspective #2

Figure 5.7: Principal component scatter plot (3D) overlaid with polyhedron clusters (removal of 3(a) visualization)

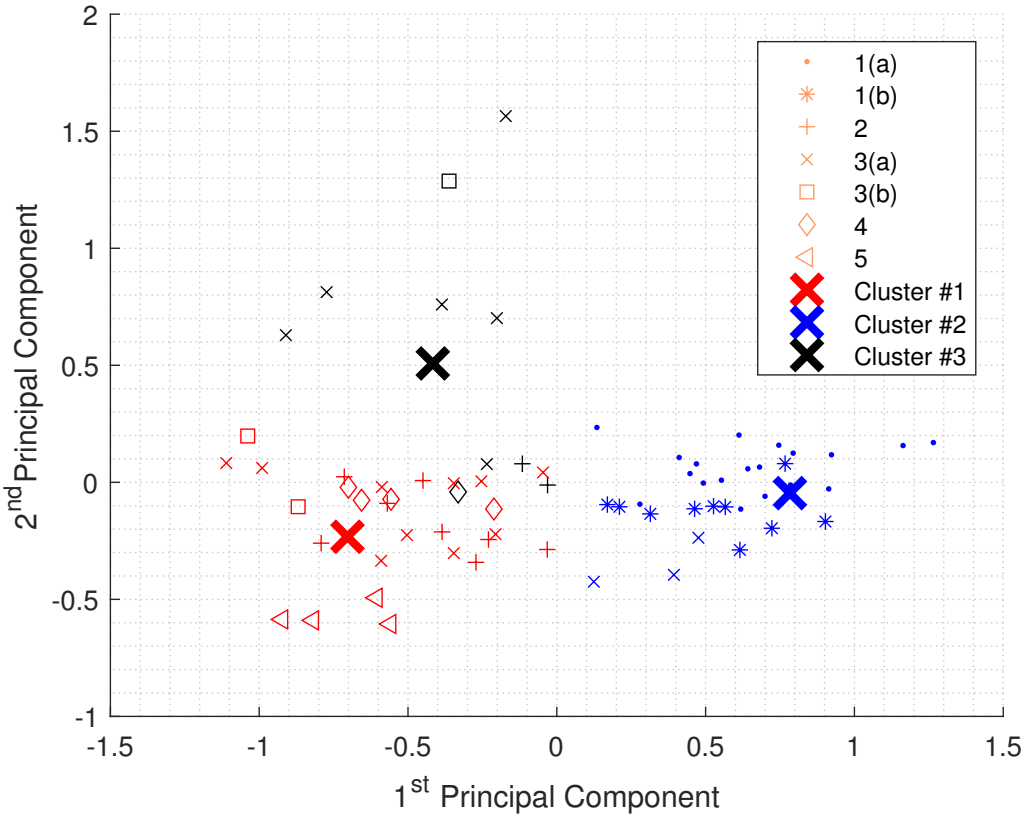
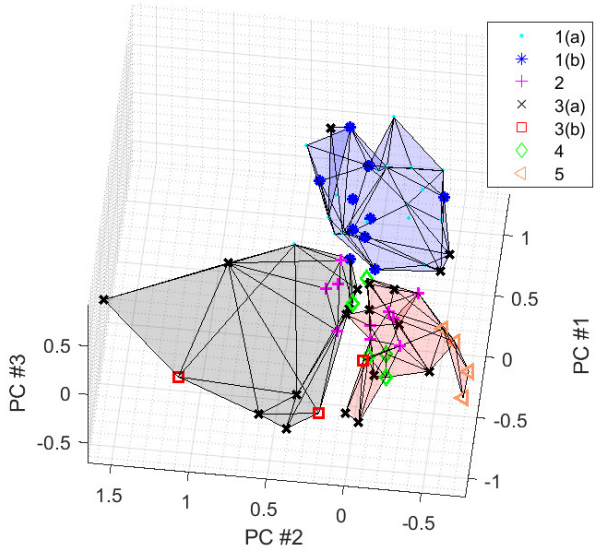
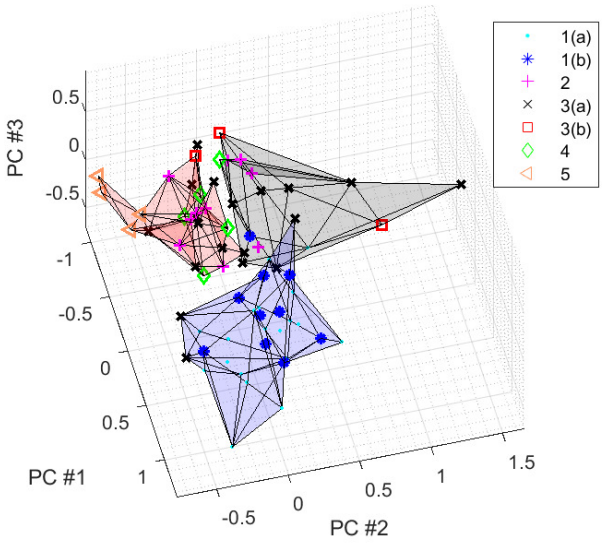


Figure 5.8: Y-axis k-means clustering results (3 centroids)



(a) 3D PCA visualization of K-means clustering (3 centroids): Perspective #1



(b) 3D PCA visualization of K-means clustering (3 centroids): Perspective #2

Figure 5.9: 3D visualization of PC space with polyhedron clusters as K-means class assignments

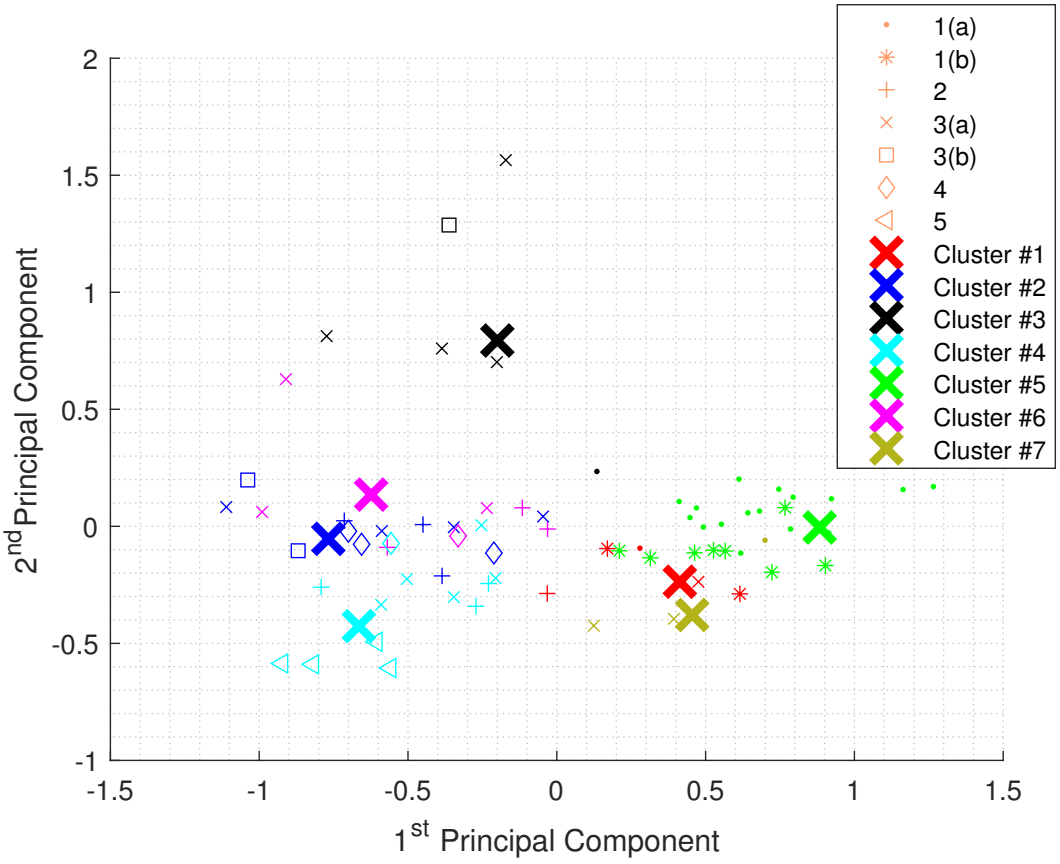


Figure 5.10: Y-axis k-means clustering results (7 centroids)

5.2.3 X-axis Similarity Analysis

X-axis segmentations were subsequently analyzed using cosine similarity analysis, as described in Figure 5.13. Compared to y-axis segmentations, most x-axis segmentations were not as easily separable into the 8 known actuation sub-classes. The majority of blue-shaded regions and high similarity scores suggests that the x-axis motion features do not contain enough discriminating information across neighboring classes. Considering the reduced power in acoustic signals as well as manifesting in a smaller frequency band than y-axis motions, the similarity scores suggest the chosen feature set may not fully represent x-axis motion subclasses for proper delineation. Wideband, high-power corrupting noise

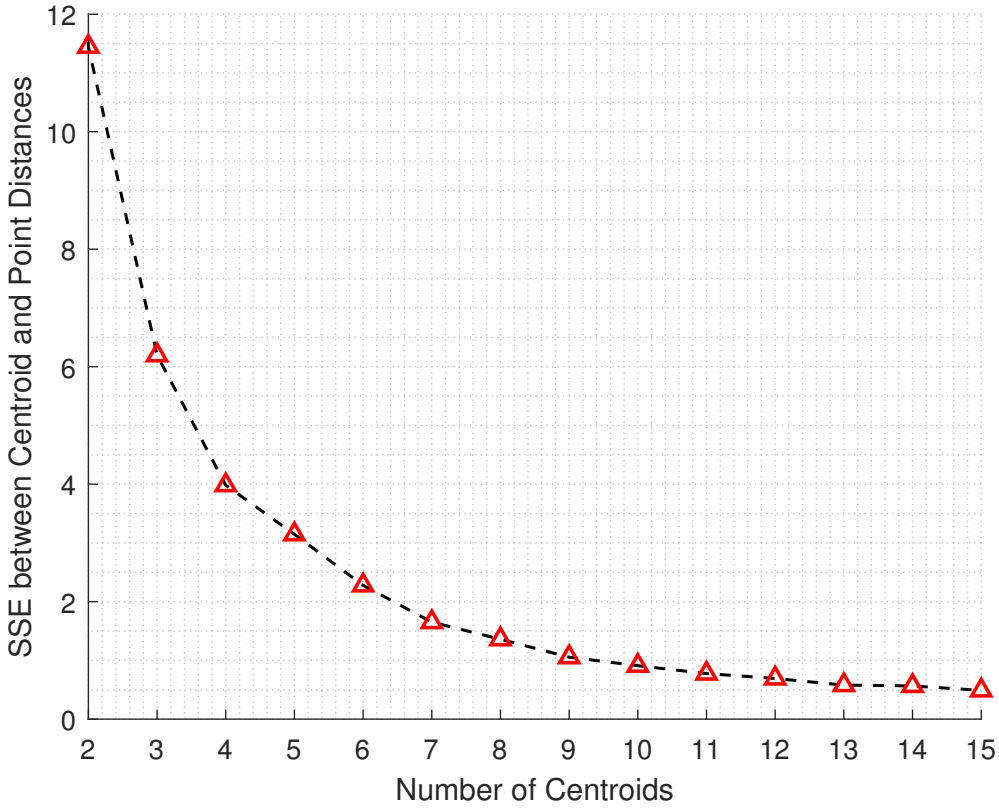


Figure 5.11: Sum of Squared Error (SSE) between Plot: y-axis motions

as well as chosen user-label fidelity may have contributed to dissimilarity across subclass samples. Some outlier segmentations were identified due to their higher dissimilarity scores, particularly in sub-classes 1 and 2.

5.2.4 X-axis Clustering Analysis

Similar to the y-axis clustering, x-axis actuations were grouped using the k-means algorithm with angular distance metrics. The two highest principal components, depicted in Figure 5.14, depict the general inseparability of these subclasses. Overlapping subclass features can be further visualized in Figures 5.16a and 5.16a. Plotting bounded regions of only x-axis

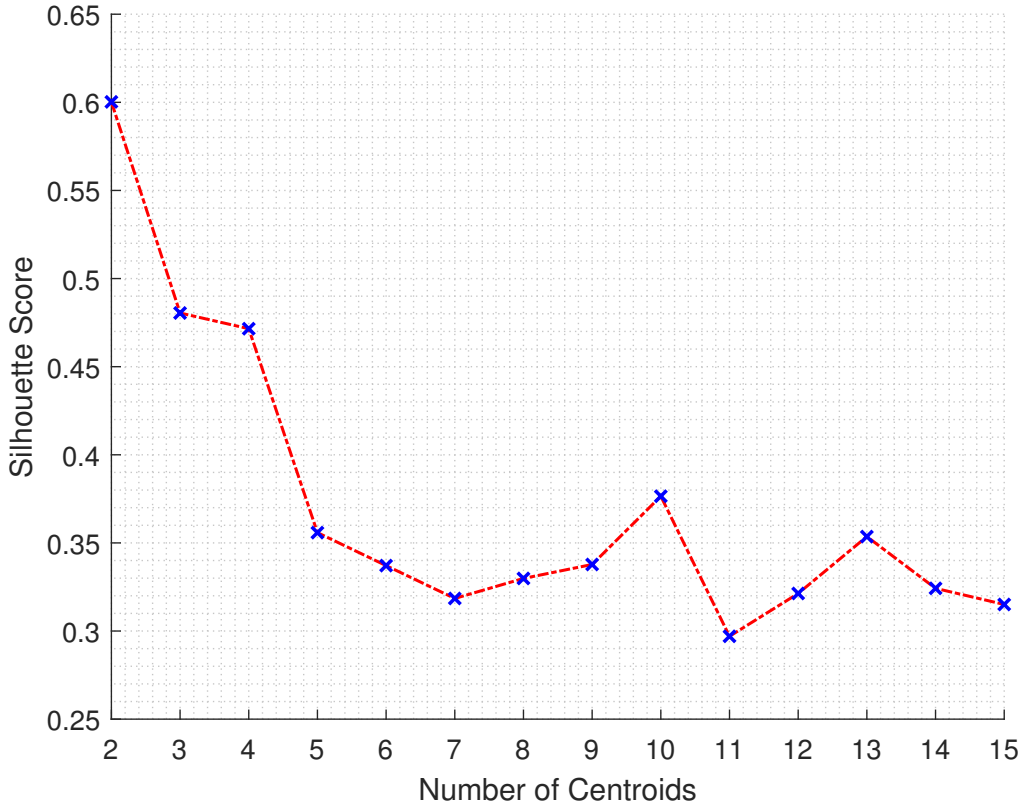


Figure 5.12: Silhoutte score: y-axis motions

motion subclasses 1 and 5, demonstrates separability limitations.

Large overlapping principal component regions between subclasses effectively limited any practical segmentation as demonstrated in 8-centroid segmentation. While Cluster #3 captures possible x-axis segmentation outliers in Figure 5.15, over-segmentation was observed from relative inseparability of extracted features. This result mirrors high cosine similarity scores across all x-axis subclasses.

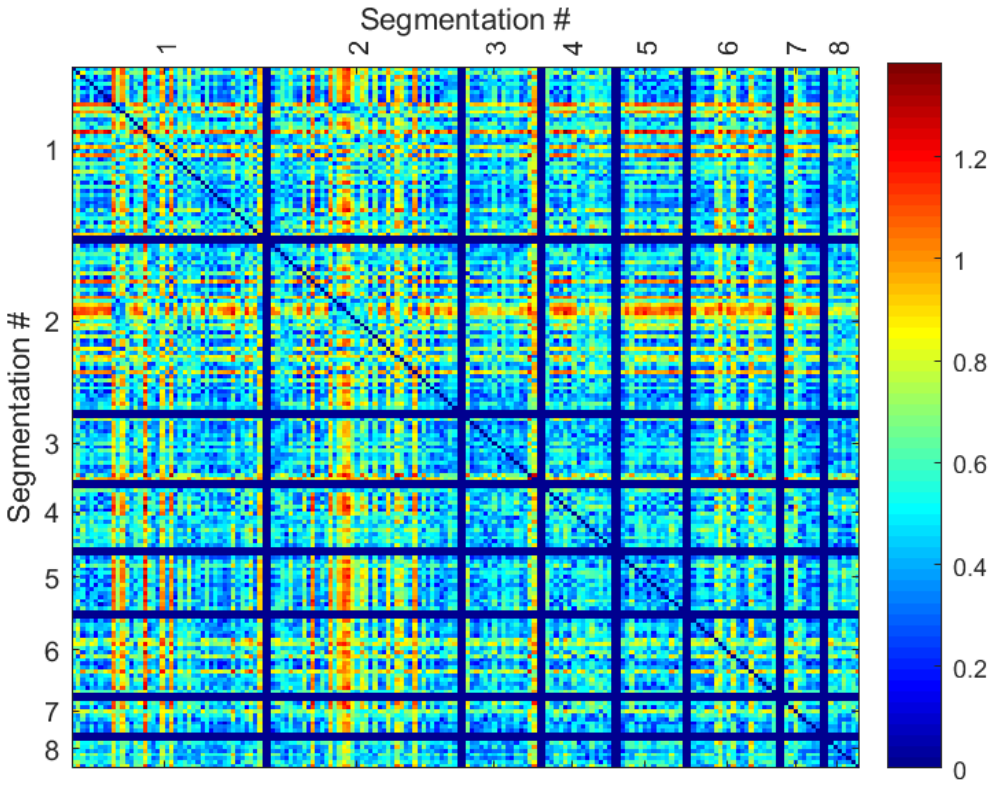


Figure 5.13: x-axis cosine similarity

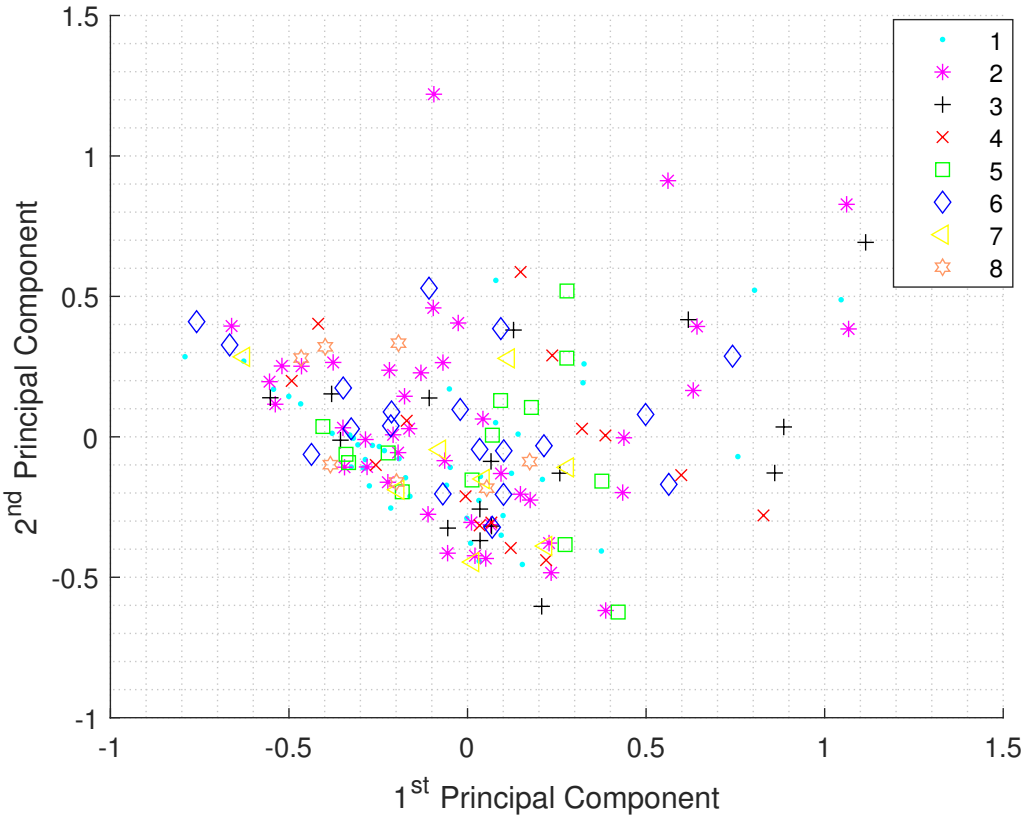


Figure 5.14: x-axis principal component plot

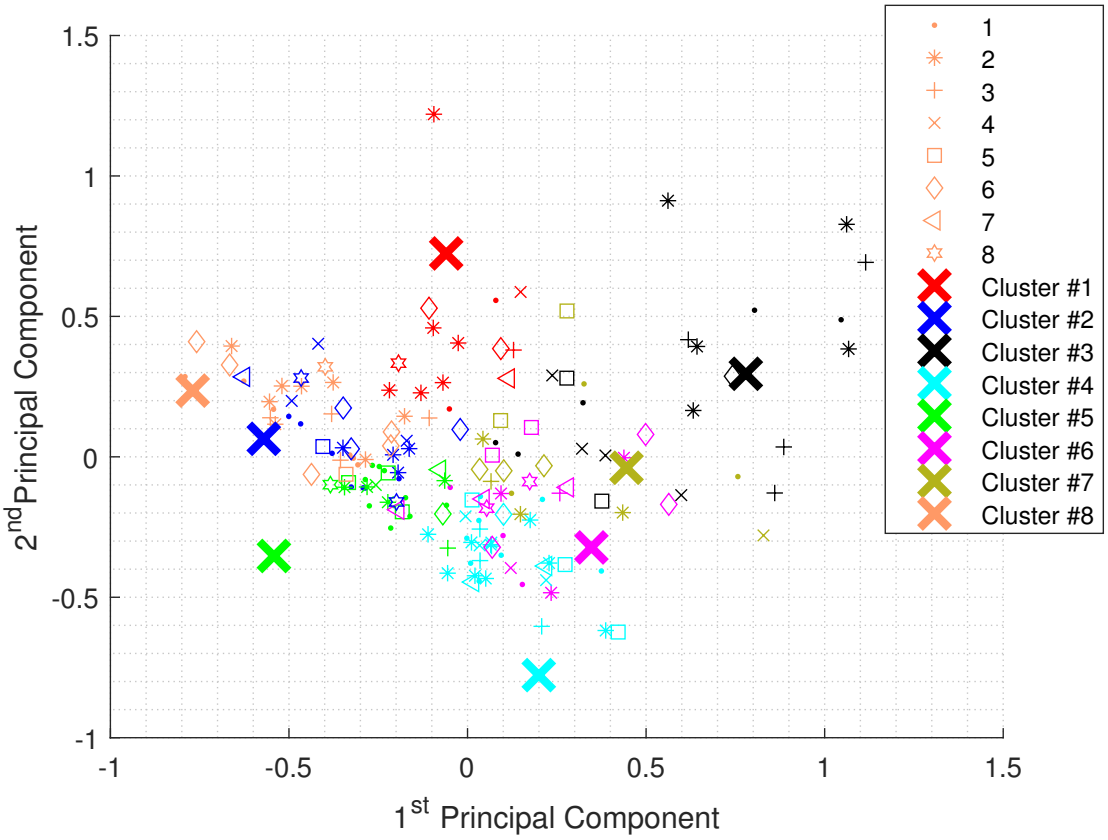
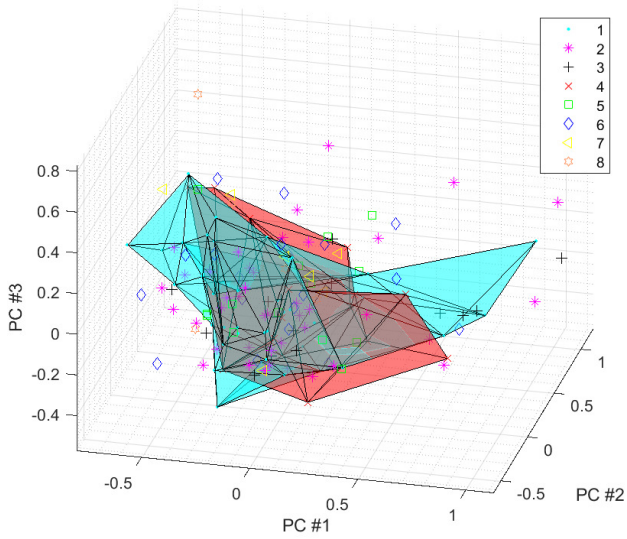
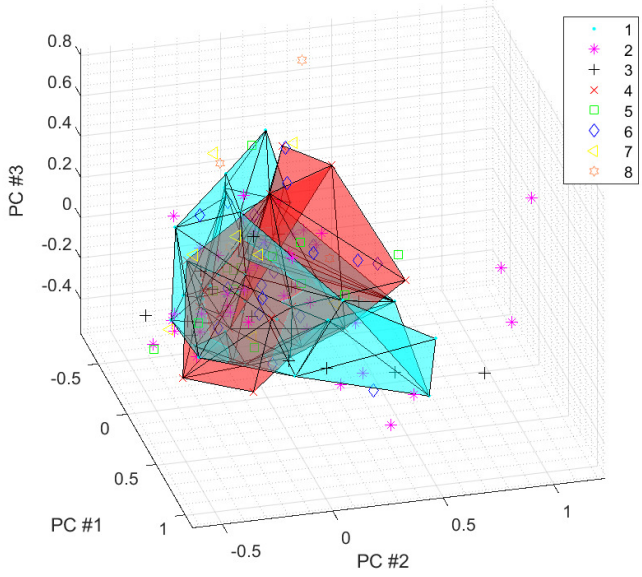


Figure 5.15: X-axis k-means clustering results (8 centroids)



(a) 3D PCA visualization of K-means clustering (3 centroids): Perspective #1



(b) 3D PCA visualization of K-means clustering (3 centroids): Perspective #2

Figure 5.16: 3D visualization of PC space with polyhedron clusters as K-means class assignments (X-axis motions)

SSE and silhouette scores for x-axis clustering, displayed in Figures 5.17 and 5.18, suggest three to four centroids could be an optimal selection for 'k.' The SSE rate observably decreases after four clusters while the silhouette ratio peaks at three centroids. These metrics, however, may take into account the variability of the motion subclasses which could be grouping outlier regions, see Figure 3.1b, rather than actual motions.

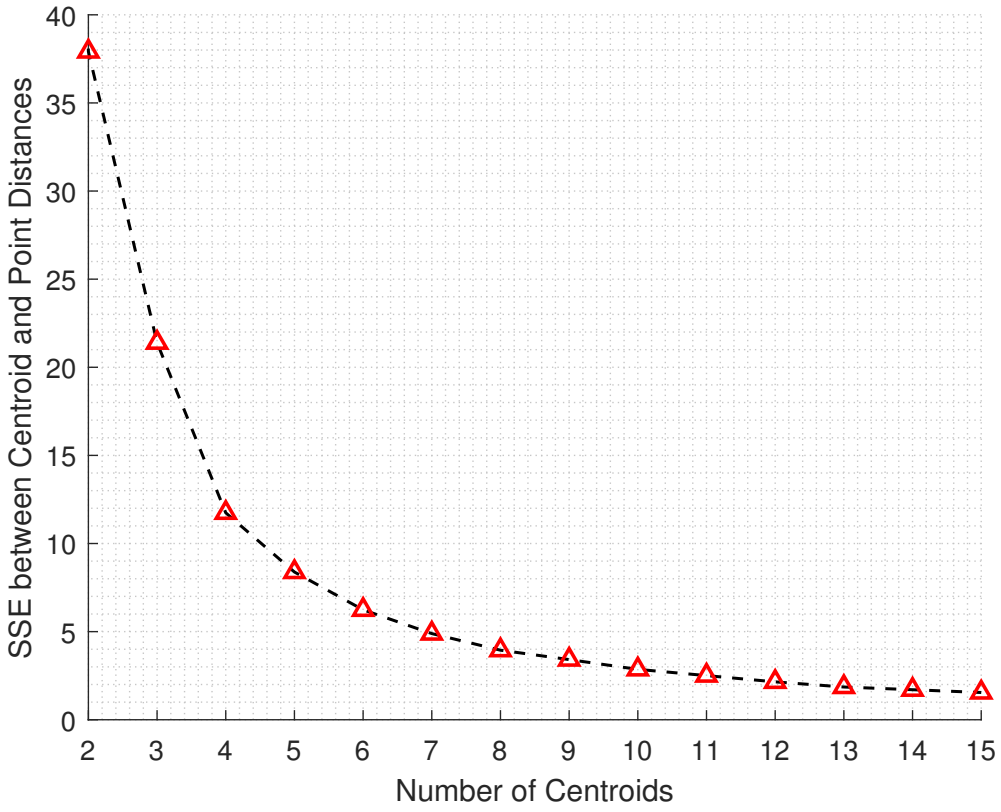


Figure 5.17: SSE graph: x-axis sub-classes

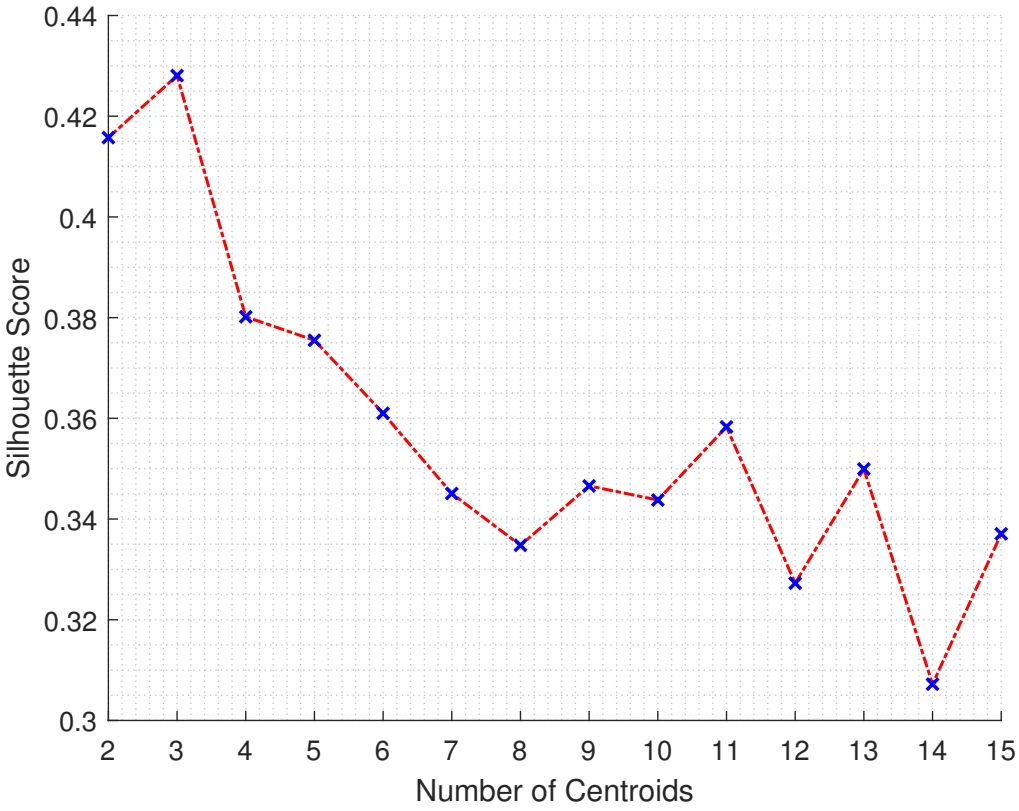


Figure 5.18: Silhouette graph: x-axis sub-classes

Chapter 6: Discussion

Several potential error sources were noted which may have negatively affected results. Each portion of the methodology had specific potential improvements.

This study presents three potential CNN architectures; however, relationships between convolutionally derived features and actuation spectrograms requires a comprehensive, hyper-parameter study. While potentially sub-optimal, the architectures introduced, demonstrated feasibility. A comprehensive exploration of asymmetric convolutional kernels is needed for spectrogram-based CNN architectures.

Sub-optimal neural network architectures may have contributed to reduced classification accuracy. Furthermore, the relatively limited training set may have unintentionally contributed to segmentation error from overfitting using data-intensive methods such as convolutional neural networks. The analyzed methodology assumed that subclass quantities were constant within each coarse, parent class training set; however, unbalanced subclasses used for training parent classes may have biased segmentations.

The proposed semi-supervised methodology is also dependent on the bounding box approach to handling overlapping segmentations, which could potentially be improved. Segmentation transitions were partially obscured by nose in some examples and limited classification accuracy in validation spectrogram examples. While y-axis and x-axis motions occurred independently, each actuation had a short transition window representing manipulator acceleration and deceleration. Spectrograms augmented with significant noise at transition stages, as visualized in spectrograms, tended to have lower classification accuracy. The accuracy, stemming from calculated softmax probabilities, became biased towards y-axis motions. For example, actuations sometimes misclassified actuation transitions in spectrograms obscured by short-time, wideband noise. This result suggests visual features

in noise, including high energy vertical spectrogram bands, may improperly bias classification to a primary class. More robust bounding algorithms, such as fuzzy classification methods, may address this concern.

The method assumes actuations do not occur simultaneously which limits direct generalizing for concurrent actuation identification problems. CNN based spectrogram semantic segmentation may address this augmented requirement, however, suffers similar limitations when developing a characteristic training set with limited known harmonic elements from each actuation.

Noise inherent to the dataset proved to be a consistent challenge with CNN processing methods. Better segmentations may be possible with a cleaner training and testing data; however, the study demonstrated that the proposed method robustly rejected a significant amount of noise artifacts in several subclass segmentation instances. Moreover, omitting preprocessing helps generalize this method to other, practical implementations for manufacturing environments.

Both 2D and 3D visualizations demonstrate potential limitations with the chosen feature set. Separation in the segmentation feature-space ultimately determined k-means clustering accuracy. The engineered features chosen were demonstrated in similar applications; however, their use may require additional work for domain specific implementation. Subclass motions demonstrated visually similar principal component spaces. These actuations, specifically subclasses 1(a) and 1(b) or subclasses 3(a) and 3(b), shared similar mechanical operation and suggested the chosen engineered feature set could not perform direct separation. This result was compounded with clusters and associated centroids pertaining to subclasses 2-4; actuations with mechanically different operation, including actuations 3(a) and 4, were classified the same cluster.

Non-stationary noise present in the experimental manufacturing environment tended to corrupt spectral features significantly, including peak definition and overall energy. Optimal filtering may present one potential improvement; smoothing spectral responses yielded an empirical improvement for harmonic features such as spectral moments. Spectral features

such as peak content required significant a priori knowledge for understanding prominence, width, magnitude, and quantity inherent in each subclass example. Spectral moments, therefore, were considered as a generalized alternative for describing harmonic information. Wavelet decomposition coefficients could serve as another alternative to filtered spectral content. Convolutional autoencoding networks present another method over manual feature generation.

A more robust clustering algorithm or non-linear dimensionality reduction technique might remove some dependency on feature-space separateness. Feature-space outliers possibly corresponding improper segmentations, or high-energy noise artifacts, conceivably reduced k-means accuracy further. Due to the wide variance of subclass 3(a), other mechanical processes may have been unknowingly and sympathetically captured by acoustic recordings during movements contributing more noise.

Moreover, user-defined labels stemming from operator intuition may be too subtle for separation with the proposed feature set and limited validation data knowledge. Suggested in comparision between x-axis actuation subclasses, fidelity in subclass clustering is coupled to characteristics and separability of underlying features.

Chapter 7: Conclusion

A remote monitoring methodology, based on semi-supervised data analysis techniques and acoustic data, was demonstrated. A supervised convolutional neural network architecture was developed to isolate primary actuation instances from an arbitrary length spectrogram. Using ground-truth, labeled spectrogram data, 87.1% of y-axis and 84.2% of x-axis motions were successfully segmented. Due to the lack of labeled training data, overfitting bias may have lowered classification accuracy. Asymmetric convolutional kernels yielded higher relative accuracy, however, requires further exploration.

An unsupervised analysis, based on cosine similarity and k-means clustering with an angular distance metric, subsequently distinguished primary y-axis actuations into higher fidelity subclasses. X-axis motions subclass segmentations were not observably separable based on lower power signals coupled with the generalized feature set used. Features, primarily comprised of temporal, statistical, and spectral content, demonstrated certain subclass segmentation from primary motions; however, were not observably separable for all subclasses in both cosine similarity and clustering visualizations. Despite limitations, including challenges with mechanically similar actuation subclasses, and noisy segmentation classification, the proposed methodology demonstrated feasibility in practical mechanical environments.

Future developments, including automatic feature generation through convolutional autoencoding networks, may increase characteristic feature separation allowing for more distinctive, inherent subclass clusters. Non-linear data reduction methods may provide another alternative to separating characteristic features from classes. Moreover, more robust clustering algorithms could compensate for mechanically similar subclasses as well as overlapping principal component spaces.

Appendix A: Feature Equations

Table A.1: Unsupervised Features

Parameter	Equation
Mean	$\frac{1}{N-1}\Sigma_{i=1}^{N} x_i$
Variance	$\frac{1}{N-1}\Sigma_{i=1}^{N} \lvert x_i - \bar{x}\rvert^2$
Standard Deviation	$\sqrt{\frac{\Sigma_{i=1}^{N}(x_i-\bar{x})^2}{N-1}}$
Mean Square	$\frac{1}{N}(x_1^2 + x_2^2 + ... + x_n^2)$
RMS	$\sqrt{\frac{1}{N}(x_1^2 + x_2^2 + ... + x_n^2)}$
Peak Amplitude	$max(\lvert x\rvert)$
Kurtosis	$\frac{\frac{1}{N}\Sigma_{i=1}^{N}(x_i-\bar{x})^4}{(\frac{1}{N}\Sigma_{i=1}^{N}(x_i-\bar{x})^2)^2}$
Skewness	$\frac{\frac{1}{n}\Sigma_{i=1}^{n}(x_i-\bar{x})^3}{(\sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(x_i-\bar{x})^2})^3}$
Crest Factor	$\frac{max(\lvert x\rvert)}{\sqrt{\frac{1}{N}(x_1^2+x_2^2+...+x_n^2)}}$
K-Factor	$max(\lvert x\rvert) * \sqrt{\frac{1}{N}(x_1^2 + x_2^2 + ... + x_n^2)}$
Zero-crossing Rate	$\frac{1}{N}\Sigma_{i=1}^{n-1}\lvert(y_{i+1} - y_i)\rvert; y \in \{0,1\}$where $y_i = (x_i > 0)$
Spectral Response (Y)	$Y(k) = 2\lvert\frac{1}{L}\Sigma_{j=1}^{n} x(j)(e^{(-2\pi i)/n})^{(j-1)(k-1)}\rvert$
Spectral Kurtosis	$\frac{\frac{1}{n}\Sigma_{i=1}^{n}(Y_i-\bar{Y})^4}{(\frac{1}{n}\Sigma_{i=1}^{n}(Y_i-\bar{Y})^2)^2}$
Spectral Skewness	$\frac{\frac{1}{n}\Sigma_{i=1}^{n}(Y_i-\bar{Y})^3}{(\sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(Y_i-\bar{Y})^2})^3}$
Spectral Crest Factor	$\frac{max(\lvert Y\rvert)}{\sqrt{\frac{1}{n}(Y_1^2+Y_2^2+...+Y_n^2)}}$
Spectral K-Factor	$max(\lvert Y\rvert) * \sqrt{\frac{1}{n}(Y_1^2 + Y_2^2 + ... + Y_n^2)}$

References

[1] K. Mathia. *Robotics for Electronics Manufacturing*. Cambridge University Press, Cambridge, United Kingdom, 2010.

[2] Alex Lenail. Publication-ready NN-architecture schematics, 2019. https://github.com/zfrenchee/NN-SVG.

[3] Ruonan Liu, Boyuan Yang, Enrico Zio, and Xuefeng Chen. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, 108:33–57, 2018.

[4] Rui Zhao, Ruqiang Yan, Zhenghua Chen, Kezhi Mao, Peng Wang, and Robert Gao. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115:213–237, 2018.

[5] Samir Khan and Takehisa Yairi. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107:241–265, 2018.

[6] Ruqiang Yan and Robert X. Gao. Multi-scale enveloping spectrogram for vibration analysis in bearing defect diagnosis . *Tribology International*, 42(2):293–302, 2009.

[7] Hussein Al-Bugharbee and Irina Trendafilova. A fault diagnosis methodology for rolling element bearings based on advanced signal pretreatment and autoregressive modelling. *Journal of Sound and Vibration*, 369:246–265, 2016.

[8] Indivarie Ubhayaratne, Michael P. Pereira, Yong Xiang, and Bernard F. Rolfe. Audio signal analysis for tool wear monitoring in sheet metal stamping. *Mechanical Systems and Signal Processing*, 85:809–826, 2017.

[9] D. Lee, V. Siu, R. Cruz, and C. Yetman. Convolutional Neural Net and Bearing Fault Analysis. *In Proceedings of the International Conference on Data Mining (WorldComp)*, page 194, 2016.

[10] Yu Wu, Hua Mao, and Zhang Yi. Audio classification using attention-augmented convolutional neural network. *Knowledge-Based Systems*, 161:90–100, 2018.

[11] J. Dennis. *Sound Event Recognition in Unstructured Environments Using Spectrogram Image Processing: Ph.D. Thesis*. School of Computer Engineering, Nanyang Technological University, Singapore, 2014.

[12] S. A. Zahorian and Z. B. Nossair. A partitioned neural network approach for vowel classification using smoothed time/frequency features. *IEEE Transactions on Speech and Audio Processing*, 7(4):414–425, 1999.

[13] Wentao Mao, Wushi Feng, and Xihui Liang. A novel deep output kernel learning method for bearing fault structural diagnosis. *Mechanical Systems and Signal Processing*, 117:293 – 318, 2019.

[14] F. Alías abd J. C. Socoró and X. Sevillano. A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds. *Applied Sciences*, 6(5):143, 2016.

[15] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. *CUIDADO I.S.T. Project Report*, pages 1–25, 2004.

[16] Chris. Aldrich. *Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods*. Advances in Computer Vision and Pattern Recognition. Springer London, London, 2013.

[17] Hoon Sohn and Charles R Farrar. Damage diagnosis using time series analysis of vibration signals. *Smart Materials and Structures*, 10(3):446–451, jun 2001.

[18] Yaguo Lei, Feng Jia, Jing Lin, Saibo Xing, and Steven X Ding. An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data. *IEEE Transactions on Industrial Electronics*, 63(5):3137,3147, 2016-05.

[19] L. Qiao, A. Esmaeily, and H. G. Melhem. Signal pattern recognition for damage diagnosis in structures. . *Computer-Aided Civil and Infrastructure Engineering*, 27(9):699–710, 2012.

[20] Adam Glowacz. Acoustic based fault diagnosis of three-phase induction motor. *Applied Acoustics*, 137, 2018.

[21] Jesus R. Rivera-Guillen, J.J. De Santiago-Perez, Juan P. Amezquita-Sanchez, Martin Valtierra-Rodriguez, and Rene J. Romero-Troncoso. Enhanced FFT-based method for incipient broken rotor bar detection in induction motors during the startup transient. *Measurement*, 124:277–285, 2018.

[22] Z. Chen, C. Li, and R. Sanchez. Gearbox fault identification and classification with convolutional neural networks. *Shock and Vibration*, pages 1–10, 2015.

[23] J.H. Bach, J. Anemüller, and B. Kollmeier. Robust speech detection in real acoustic backgrounds with perceptually motivated features. *Speech Communication*, 53:690–706, 2011.

[24] F. Jia, Y. Lei, N. Lu, and S. Xing. Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. *Mechanical Systems and Signal Processing*, 110:349–367, 2018.

[25] Sophocles J. Orfanidis. *Introduction to Signal Processing*. Pearson Education, Inc, 1996–2009.

[26] Dingcheng Zhang, Mani Entezami, Edward Stewart, Clive Roberts, and Dejie Yu. Adaptive fault feature extraction from wayside acoustic signals from train bearings. *Journal of Sound and Vibration*, 425:221 – 238, 2018.

[27] Mario A. De Oliveira, Nelcileno V. S. Araujo, Rodolfo N. Da Silva, Tony I. Da Silva, and Jayantha Epaarachchi. Use of savitzky–golay filter for performances improvement of shm systems based on neural networks and distributed pzt sensors. *Sensors*, 18(1), 2018.

[28] F. Combet and L. Gelman. Optimal filtering of gear signals for early damage detection based on the spectral kurtosis. *Mechanical Systems and Signal Processing*, 23(3):652 – 668, 2009.

[29] H. Endo and R.B. Randall. Enhancement of autoregressive model based gear tooth fault detection technique by the use of minimum entropy deconvolution filter. *Mechanical Systems and Signal Processing*, 21(2):906 – 919, 2007.

[30] J.M. Spyers-Ashby, P.G. Bain, and S.J. Roberts. A comparison of fast fourier transform (FFT) and autoregressive (AR) spectral estimation techniques for the analysis of tremor data. *Journal of Neuroscience Methods*, 83(1):35–43, 1998.

[31] A. Ghaffari, M. R. Homaeinezhad, M. Atarod, and R. Rahmani. Detecting and quantifying T-wave alternans using the correlation method and comparison with the FFT-based method . *In Computers in Cardiology*, pages 761–764, 2008.

[32] F. Scholkmann, J. Boss, and M. Wolf. An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals . *Algorithms*, 5(4):588–603, 2012.

[33] E. L. Kosarev and E. Pantos. Optimal smoothing of 'noisy' data by fast Fourier transform . *Journal of Physics E: Scientific Instruments*, 16(6):537, 1983.

[34] Peiyang Li, Xurui Wang, Fali Li, Rui Zhang, Teng Ma, Yueheng Peng, Xu Lei, Yin Tian, Daqing Guo, Tiejun Liu, Dezhong Yao, and Peng Xu. Autoregressive model in the Lp norm space for EEG analysis. *Journal of Neuroscience Methods*, 240:170–178, 2015.

[35] Z. B. Nossair, P. L. Silsbee, and S. A. Zahorian. Signal modeling enhancements for automatic speech recognition. *In Acoustics, Speech, and Signal Processing: ICASSP-95 (IEEE)*, 1:824–827, 1995.

[36] D. T. Chappell and J. H. Hansen. A comparison of spectral smoothing methods for segment concatenation based speech synthesis . *Speech Communication*, 36(3-4):343–373, 2002.

[37] R. Martin. Noise power spectral density estimation based on optimal smoothing and minimum statistics. *IEEE Transactions on speech and audio processing*, 9(5):504–512, 2001.

[38] L. Fanioudakis and I. Potamitis. Deep Networks tag the location of bird vocalisations on audio spectrograms. *arXiv preprint*, 2017.

[39] Jianxin Wu. Introduction to convolutional neural networks. Technical report, National Key Lab for Novel Software Technology,Nanjing University, China, 2017.

[40] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *ArXiv e-prints*, 11 2015.

[41] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis.

[42] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *ACM Multimedia*, 2015.

[43] K. B. Lee, S. Cheon, and C. O. Kim. A Convolutional Neural Network for Fault Classification and Diagnosis in Semiconductor Manufacturing Processes. *IEEE Transactions on Semiconductor Manufacturing*, 30(2):135–142, 2017.

[44] H. Zhang, I. McLoughlin, and Y. Song. Robust sound event recognition using convolutional neural networks. *In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference*, pages 559–563, 2015.

[45] M. Espi, M. Fujimoto, K. Kinoshita, and T. Nakatani. Exploiting spectro-temporal locality in deep learning based acoustic event detection. . *EURASIP Journal on Audio, Speech, and Music Processing*, (1)26, 2015.

[46] V. Boddapati, A. Petef, J. Rasmusson, and L. Lundberg. Classifying environmental sounds using image recognition networks. *Procedia Computer Science*, 112:2048–2056, 2017.

[47] Peerapol Khunarsal, Chidchanok Lursinsap, and Thanapant Raicharoen. Very short time environmental sound classification based on spectrogram pattern matching . *Information Sciences*, 243:57–74, 2013.

[48] S. Chachada and C. C. J. Kuo. Environmental sound recognition: A survey. *APSIPA Transactions on Signal and Information Processing*, 3, 2014.

[49] I. McLoughlin, H. Zhang, Z. Xie, Y. Song, and W. Xiao. Robust Sound Event Classification Using Deep Neural Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):540–552, 2015.

[50] Ilyas Ozer, Zeynep Ozer, and Oguz Findik. Noise robust sound event classification with convolutional neural network . *Neurocomputing*, 272:505–512, 2018.

[51] K. J. Piczak. Environmental sound classification with convolutional neural networks. *IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)* , (1)26:1–6, 2015.

[52] S. Li, G. Liu, X. Tang, J. Lu, and J. Hu. An ensemble deep convolutional neural network model with improved DS evidence fusion for bearing fault diagnosis. *Sensors*, 17(8):1729, 2017.

[53] M. Sandsten, M. Große Ruse, M. Jönsson, and J. EURASIP. Robust feature representation for classification of bird song syllables. *Adv. Signal Process.*, 68, 2016.

[54] M. L. Cooper and J. Foote. Robust Sound Event Classification Using Deep Neural Networks. *Automatic Music Summarization via Similarity Analysis. In ISMIR.*, 2002.

[55] M. Cooper and J. Foote. Summarizing popular music via structural similarity analysis. *In Applications of Signal Processing to Audio and Acoustics: IEEE Workshop.*, pages 127–130, 2003.

[56] G. Jang, H. G. Kim, and Y. H. Oh. Audio Source Separation Using a Deep Autoencoder . *arXiv preprint*, (1)26, 2014.

[57] Chris Piech. K means (cs211 lecture notes). *http://stanford.edu/ cpiech/cs221/handouts/kmeans.html*, 2013.

[58] Dibya Jyoti Bora and Anil Kumar Gupta. Effect of different distance measures on the performance of k-means algorithm: An experimental study in matlab. *CoRR*, abs/1405.7471, 2014.

[59] MathWorks Inc. k-means clustering. *https://www.mathworks.com/help/stats /kmeans.html# buefs04-Options*, 2019.

[60] Kurt Hornik, Ingo Feinerer, Martin Kober, and Christian Buchta. Spherical k-means clustering. *Journal of Statistical Software*, 50(10):1–22, 2012.

[61] Shi Zhong. Efficient online spherical k-means clustering. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 5, pages 3180–3185 vol. 5, July 2005.

[62] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

[63] Milad Afzalan and Farrokh Jazizadeh. An automated spectral clustering for multi-scale data. *Neurocomputing*, 347:94 – 108, 2019.

[64] A. M. Rajee and F. Sagayaraj Francis. A study on outlier distance and sse with multidimensional datasets in k-means clustering. In *2013 Fifth International Conference on Advanced Computing (ICoAC)*, pages 33–36, Dec 2013.

[65] F. R. Bach and M. I. M. I. Jordan. Spectral clustering for speech separation. *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods*, 369:221–253, 2009.

[66] N. Takahashi, M. Gygli, and L. Van Gool. Aenet: Learning deep audio features for video analysis. *IEEE Transactions on Multimedia*, 20(3):513–524, March 2018.

[67] C. Wang, A. Santoso, S. Mathulaprangsan, C. Chiang, C. Wu, and J. Wang. Recognition and retrieval of sound events using sparse coding convolutional neural network. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 589–594, July 2017.

[68] R. Monteiro, C. Bastos-Filho, M. Cerrada, D. Cabrera, and R. Sánchez. Convolutional neural networks using fourier transform spectrogram to classify the severity of gear tooth breakage. In *2018 International Conference on Sensing,Diagnostics, Prognostics, and Control (SDPC)*, pages 490–496, Aug 2018.

[69] R. K. Nayyar, S. Nair, O. Patil, R. Pawar, and A. Lolage. Content-based auto-tagging of audios using deep learning. In *2017 International Conference on Big Data, IoT and Data Science (BID)*, pages 30–36, Dec 2017.

[70] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey, and P. Tiwari. Sound classification using convolutional neural network and tensor deep stacking network. *IEEE Access*, 7:7717–7727, 2019.

[71] Z. Ren, K. Qian, Y. Wang, Z. Zhang, V. Pandit, A. Baird, and B. Schuller. Deep scalogram representations for acoustic scene classification. *IEEE/CAA Journal of Automatica Sinica*, 5(3):662–669, May 2018.

[72] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3856–3866. Curran Associates, Inc., 2017.

[73] J. Zhou, W. Xu, and R. Chellali. Analysing the effects of pooling combinations on invariance to position and deformation in convolutional neural networks. In *2017 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pages 226–230, Oct 2017.

[74] Dingjun Yu, Hanli Wang, Peiqiu Chen, and Zhihua Wei. Mixed pooling for convolutional neural networks. pages 364–375, 10 2014.

[75] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net, 2014.

[76] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27 – 48, 2016. Recent Developments on Deep Big Vision.

[77] U. Rajendra Acharya, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, and Hojjat Adeli. Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals. *Computers in Biology and Medicine*, 100:270,278, 2018-09-01.

[78] A. Coates and A. Ng. *Learning Feature Representations with K-Means.* Springer, Berlin, Heidelberg, 2012.

[79] Junjie Wu, Hui Xiong, and Jian Chen. Adapting the right measures for k-means clustering. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data. ACM.*, 2009.

[80] Seunghee Park, Jong-Jae Lee, Chung-Bang Yun, and Daniel J. Inman. Electromechanical impedance-based wireless structural health monitoring using pca-data compression and k-means clustering algorithms. *Journal of Intelligent Material Systems and Structures*, 19(4):509–520, 2008.

[81] S. Banerjee, A. Choudhary, and S. Pal. Empirical evaluation of k-means, bisecting k-means, fuzzy c-means and genetic k-means clustering algorithms. In *2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, pages 168–172, Dec 2015.

[82] Mehrisadat Makki Alamdari, Thierry Rakotoarivelo, and Nguyen Lu Dang Khoa. A spectral-based clustering for structural health monitoring of the sydney harbour bridge. *Mechanical Systems and Signal Processing*, 87:384 – 400, 2017.

Biography

Jeff Bynum is the first recipient of a Bachelor's of Science Degree in Mechanical Engineering from George Mason University in 2016. After contributing to various projects and publications, as both an Undergraduate and Graduate Research Assistant under mentorship of Dr. David Lattanzi, he went on to receive his Master's of Science in Electrical Engineering with a concentration in Control Systems and Robotics in 2019. He wishes to pursue future multidisciplinary research endeavors that intertwine manufacturing, art, robotics, instrumentation, and machine learning disciplines.