Rapid Remaining-Useful-Life Prediction of Li-ion Batteries Using Image-Based Machine Learning

by

Jonathan Couture

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Applied Science in Automotive Engineering

Department of Automotive and Mechatronics Engineering
School of Graduate and Postdoctoral Studies
The Faculty of Engineering and Applied Science
Ontario Tech University
Oshawa, Ontario, Canada

August 2022

THESIS EXAMINATION INFORMATION

Submitted by: **Jonathan Couture**

Maser of Applied Science in Automotive Engineering

Thesis title: Rapid Remaining-Useful-Life Prediction of Li-ion Batteries Using Image-Based Machine Learning

An oral defense of this thesis took place on August 08, 2022, in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Martin Agelin-Chaab
Research Supervisor	Dr. Xianke Lin
Examining Committee Member	Dr. Meaghan Charest-Finn
Thesis Examiner	Dr. Mathew Harker

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

With the increased integration of lithium-ion batteries in our everyday lives, accurate and reliable battery management systems have become an imperative aspect of the well-being of our everyday electronics. This thesis proposes the use of novel machine learning methodologies to predict the remaining-useful-life (RUL) of lithium-ion batteries reliably, accurately, and swiftly. Firstly, a method that prides itself on being publicly available, and which can be easily implemented alongside existing methodology, is proposed to increase the prediction accuracy of the conventional health indicator methodology by 6.72% by using images of data curves as inputs. Subsequently, a more in-depth machine learning model is presented which managed to considerably outperform the current literature in terms of speed, accuracy, and reliability, achieving an RUL prediction accuracy of 90.85%. These proposed methodologies have a wide range of applications, from fault diagnostics, state-of-charge, and state-of-health prediction, to other, more complex, regression applications.

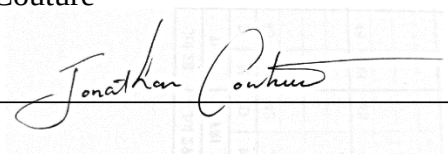
Keywords: Health prognostics prediction, lithium-ion battery, remaining-useful-life estimation, machine learning, transfer learning, capsule network

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize the University of Ontario Institute of Technology to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Jonathan Couture



ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor, Dr. Xianke Lin, for his support and guidance throughout this research process, my colleagues, which were always willing to lend a helping hand, and my wife, who supported me throughout.

STATEMENT OF CONTRIBUTIONS

The work described within this thesis has been published as:

J. Couture and X. Lin, "Novel Image-Based Rapid RUL Prediction for Li-Ion Batteries Using a Capsule Network and Transfer Learning," in *IEEE Transactions on Transportation Electrification*, doi: 10.1109/TTE.2022.3173918.[1]

J. Couture and X. Lin, “Image- and Health-Based Transfer Learning Hybridization for Battery RUL Prediction”, in *Engineering Applications of Artificial Intelligence*, doi: 10.1016/j.engappai.2022.105120.[2]

TABLE OF CONTENTS

LIST OF TABLES

Chapter 4

Chapter 5

LIST OF FIGURES

Chapter 1

Chapter 3

Chapter 4

LIST OF ABBREVIATIONS

ANN	Artificial neural network
BMS	Battery management system
CNN	Convolutional neural network
DCNN	Dilated convolutional neural network
ECM	Equivalent circuit model
EIS	Electrochemical impedance spectroscopy
EV	Electric vehicle
FCNN	Fully-connected neural network
GAN	Generative adversarial networks
GPR	Gaussian process regression
HI	Health indicator
LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MLP	Multilayer perceptrons
MLR	Multivariable linear regression
MSE	Mean-squared error
NB	Naïve Bayes
ReLU	Rectified linear unit
RMSE	Root-mean-squared error
RNN	Recurrent neural network

RUL	Remaining-useful-life
RVM	Relevance vector machine
SEI	Solid electrolyte interface
SGD	Stochastic gradient descent
SVM	Support vector machine
SVR	Support-vector regression
TLPH	Transfer learning parallel hybrid

Chapter 1. Introduction

1.1. Background and motivation

Since the invention of automobiles in the late 19th century, their sales have only been expanding, and today's day and age is no exception. The internal combustion engine allowed citizens to move around faster than ever and facilitated trade between societies. However, these vehicles had an unforeseen effect on our environment, being the release of greenhouse emissions as a byproduct of fuel combustion. According to the Center for Climate and Energy Solutions, the use of fossil fuels to power the modern transportation infrastructure constitutes just over 15% of all global carbon emissions[3]. Since the discovery of the negative impacts of the internal combustion engine, the race has been on to find suitable alternatives for the now very well-established transportation industry.

While not being a new technology, the electric vehicle has recently been widely adopted as the frontrunner to replace the fossil fuel-driven industry. Although still not a perfect solution, electric vehicles (EVs) offer a promising alternative to internal combustion engines. Due to the way that modern electricity still makes use of non-renewable energy and the need for mining rare-earth materials, there is a rapid need for more green energy sources to supply the increasing EV demand. This motivated many governments around the world to create transportation electrification incentives in recent years, which created an exploding demand for Evs. For their well-functioning, these require intricate algorithms that need to be able to measure the state-of-charge of the battery packs, its state-of-health, and among other factors, the degradation of the batteries themselves. This allows the driver to be aware of these prognostics and also helps onboard safety

measures function appropriately. Also, when taking into consideration the fact that changing an electric vehicle's battery pack can cost the consumer up to $20,000 USD[4], the importance of the well functioning of this component is clear.

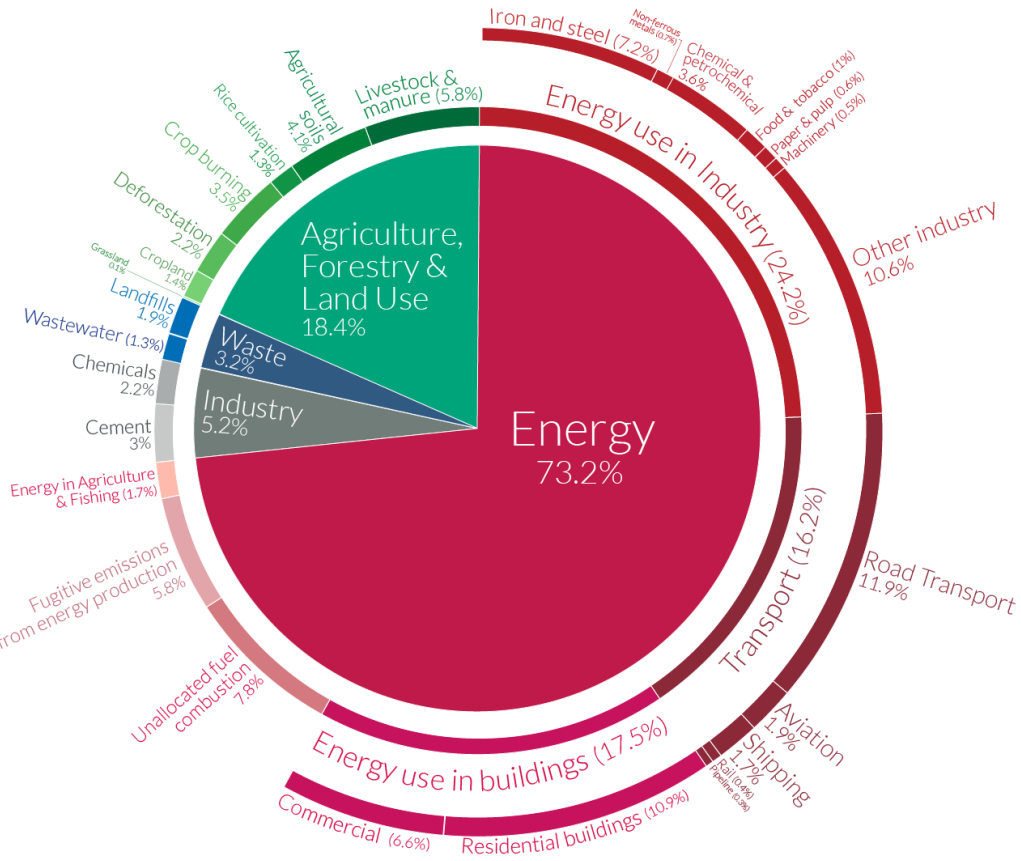


Figure 1. Global greenhouse gas emissions by sector[5]

The remaining-useful-life of a battery pack refers to the remaining number of combined charging and discharging cycles before the battery reaches a state-of-health of 80%, or in other words before its energy capacity becomes 80% of its initial value due to degradation. Various factors come into play when considering the degradation of a lithium-ion battery's capacity, making it infeasible to model accurately. In this thesis, the term accurate will be used to describe a methodology that has a relatively low prediction error. To name a few, extreme temperatures, solid electrolyte interface (SEI), loss of lithium

inventory, and lithium plating all contribute to the overall capacity fade, but each to a varying degree. These phenomenons all combine and make their prediction a very challenging task. It is for this reason that so much effort has been poured into efficient prediction methodologies within the past years.

Also, recently, an application has been derived for used electric vehicle battery cells where the batteries are removed and disassembled from their original packs, are then classified according to their state of degradation, and reassembled to be used for stationary storage applications. This process, commonly referred to as battery second life, presents huge improvements over the overall carbon emissions of lithium-ion cells. When taking into account the manufacturing of these batteries, CO2 emissions reduction of over 70% could be achieved by using retired EV batteries instead of new ones[6]. The most common consumers of large-scale energy storage are renewable energy farms, which store the generated electricity to discharge during peak power demand.

This is made possible by the relatively low requirements of stationary energy storage when compared to the dynamic drive cycles of vehicles. However, current methodologies for second-life classification only take into account the state-of-health of the cells and not their degradation rate, which ends up producing inconsistent battery packs, which in turn hurts the longevity of the refurbished batteries[7]. This classification process also needs to be rapid due to the sheer amount of cells needed to be tested. Taking a Tesla Model S for example, the battery pack holds upwards of 7000 li-ion cells. Pairing the existing methodology with RUL prediction might be the key to longer-lasting energy storage implementations, which would in turn help further reduce lithium-ion batteries' carbon footprint on the environment.

Owing to the importance of these battery health prognostics, scientists around the globe have been committed to increasing the accuracy and speed of their prediction algorithms to decrease the likelihood of battery faults. However, this has proven to be a daunting task due to the non-linearity of the internal degradation mechanisms, compelling researchers to further push existing methodologies. Initially, the literature consisted mainly of model-based prediction approaches, where a battery pack was modeled and simulated under given scenarios and the results were projected onto the physical battery pack. This approach proved to lack adjustability and required a large amount of in-depth knowledge to develop. Even so, this technique is still used to this day due to its low computational complexity. In recent years, data-driven approaches have been consistently taking over the simulations by using physical data gathered from battery cycling. This concept of letting computers learn by themselves has astonished researchers for years with their ability to learn complex non-linear relationships between variables. Still, no one method is without fault, with machine learning methodologies generally requiring large amounts of data and extensive computational hardware.

These drawbacks have motivated researchers in all fields to try and perfect these machine learning approaches, one of its popular implementation being the neural network. Inspired by the human brain, a mathematical neuron was proposed in 1943[8] that could take numerical values as inputs, transform these, and give out a different numerical output. It was discovered that, not unlike the human brain, sequentially laying a large number of these neurons could allow the network to distinguish non-linear relationships between the inputs and the output and learn to recognize these. From this, the first fully-connected neural network was developed.

Since then, many advancements have been made to the original architecture, and due to its wide array of applications and uncomplicated data gathering, image recognition networks have had significant improvements in recent years. So much so, that of the writing of this thesis, all available pre-trained neural networks on PyTorch, a popular machine learning library, are convolution-based image recognition networks. These pre-trained networks allow users to have a starting point for their training if used on a different but related dataset. This can be extremely useful since most modern networks can take up to 2-3 weeks to train their model on the ImageNet dataset, which contains 1.2 million images across 1000 categories[9]. This transfer is made possible by the CNNs architecture, which consists firstly of a number of convolutional layers, followed by fully-connected layers. These primary convolutions are used to extract useful information from the shown image by making use of kernels, or filters. Then, this extracted information is fed through the fully-connected neurons which act as classifiers for the network. These networks are taught through backpropagation to remember which given set of extracted features represents which desired output. Owing to the clear division in tasks from these two main components, it is possible to separate the convolutions from the fully-connected layers, and then repurpose the feature extraction capabilities of a trained network on a different, but related task. This allows users to only have to train the final layers instead of the entire architecture, which in turn greatly reduces the computational burden of the task, and helps prevent overfitting.

Since they are no numerical input-based models available for transfer, this thesis was motivated to determine whether it was possible to make use of the image recognition networks in other applications, such as battery prognostic prediction. This would allow

developers to save a significant amount of preprocessing labor in terms of architecture development and data gathering and processing. Overall, these networks also reduce the knowledge of the field required by shifting the responsibility to the machine learning model to determine the usefulness of the data. This theory would also have various other applications in many other areas of research where the numerical inputs can be transformed into images. To test this theory, this thesis adopts an image-based machine learning approach to achieve high accuracy and speed for batteries' remaining-useful-life (RUL) prediction.

1.2. Objectives and assumptions

The objective of this study is to provide a proof of concept as to the efficiency of using images for regression applications. The use of images as inputs for regression applications has a wide range of applicability due to all publicly available transferrable networks being image-based. This allows researchers and even consumers to access the expertise and advancements within the field of computer vision and also the resources of large technology companies that outweigh general consumers' hardware. This research will also seek to achieve a rapid RUL prediction methodology with high accuracy for use in battery management systems (BMS) and second-life classification applications. An accurate prediction algorithm plays an important role in the longevity of li-ion batteries by providing higher quality information to the onboard computer. This will also allow future researchers to draw more accurate conclusions regarding the efficiency of their developed charging and discharging protocol and drive cycle simulations where the battery degradation is sought to be limited.

The detailed objectives of this research include:

(1) To achieve a high li-ion battery RUL prediction accuracy to provide a proof of concept that images can be used for regression tasks. This in turn will have a wide range of applicability regarding the use of computer vision-based methodologies for regression applications.

(2) To provide a novel rapid and accurate RUL prediction methodology to add to the existing literature an efficient methodology with real-life applications. Many developed models are simply too computationally expensive or require too much data to be deployed in online applications. This thesis seeks to overcome both of these factors by using machine learning models that do not require that its entire architecture be trained, which in turn limits the amount of data needed and greatly reduces the computational burden of the developed methodology.

(3) To propose a method that would be easily implementable and would still have significant benefits by using publicly available neural networks. This gives the proposed methodology a higher potential for real-life deployment, as more complex methodologies require an increased level of expert knowledge.

(4) To reduce the data preprocessing required to limit the amount of human-induced bias in the datasets. This objective came to light as modern RUL prediction methodologies require a large amount of data processing to obtain health indicators to use as inputs.

The following assumptions are made throughout the thesis:

(1) It is assumed that the dataset used contains sufficient variance to provide a proof of concept for the learning capabilities of the developed neural networks

(2) That the dataset does not contain a significant amount of faulty data that would skew the results of this study compared to real-life applications of the models.

1.3. Contributions

To the best of the author's knowledge, the main contributions of this research are listed below:

(1) The use of image-based inputs is introduced for the first time for li-ion batteries' RUL prediction. The proposed machine learning models use generated images as their inputs to obtain an accurate RUL prediction, validating the efficiency of the proposed methodologies.

(2) This research presents the first implementation of a capsule network for use in regression applications. This provides a proof-of-concept as to additional applications and efficiency of this lesser-used neural network architecture.

(3) A comparison of various transfer learning models is made for the use of data curves as inputs which can be used in the future as a starting point to determine which model best suits a given application.

(4) Two novel machine learning architectures are proposed which managed to outperform the currently available literature in terms of accuracy, speed, and reliability. Reliability, in this context, addressing to the robustness of the developed methodologies. These are comprised of the proposed transfer learning parallel

hybrid model, which can be used in parallel with an existing health indicator-based methodology, and a transfer learning-based capsule network architecture, which provides a more accurate prediction at the cost of an increased computational burden.

1.4. Outline

This thesis is organized as follows:

- **Chapter 1 Introduction** presents the motivation of this research alongside its contributions.
- **Chapter 2 Literature Review** will review the existing literature and its current methodology regarding the prediction of li-ion batteries' RUL, as well as highlight some research gaps.
- **Chapter 3 Methodology** will introduce the concepts and the developed methodologies used within this thesis.
- **Chapter 4 Model development** will describe the recommended architecture and hyperparameter selection for the various neural network implementations developed within this study.
- **Chapter 5 Results** will be divided into various sections, where the results from this research will be presented and compared with state-of-the-art implementations from the literature.
- **Chapter 6 Conclusion and Future Works** will serve to conclude this thesis and suggest future works to be done within the field.

Chapter 2. Literature Review

2.1. Introduction

As previously mentioned, many researchers around the globe are motivated to find rapid and accurate RUL prediction algorithms, and so a plethora of methodologies were developed over the years. This chapter will review the existing literature regarding the RUL prediction strategies for lithium-ion batteries. In the end, some research gaps will also be identified and explained.

Firstly, a summary of the internal mechanisms affecting li-ion battery degradation will be presented, followed by some of the most popular prediction models available. These models will be divided into three distinct sections, the model-based approaches, then the data-driven approaches, and finally, hybrid methodologies. Lastly, the research gaps are presented and explained to justify the development of this thesis' proposed methodology.

2.2. Battery degradation

It is a well-known phenomenon that li-ion batteries' capacity degrades over time and with use. Many factors play an important role in the capacity fade of a battery, mainly chemical and mechanical factors, such as solid electrolyte interphase (SEI) formation, binder decomposition, and loss of lithium, among others[10]. These effects have varying implications for the overall capacity fade of the battery, making it a very challenging task to try and predict it due to the non-linearity of the factors' relationship. For this reason, there are many published studies on the accurate prediction of the battery's remaining useful life (RUL) within the literature.

These studies can be summarized into three main methodologies, these being model-based, data-driven, and hybrid configurations of these two[10,11]. These model-based methods are usually not as robust as the data-driven methods since they don't have a way to adapt themselves to the variability of each battery pack and the inconsistencies from the batteries themselves[12] without the help of other models. For these reasons, more and more research is shifting to the implementation of data-driven approaches, such as Support Vector Machines (SVM), Gaussian Process Regression (GPR), and neural networks, which excel in recognizing the non-linear correlation between inputted variables but which generally come at the cost of extensive computational and data requirements.

2.3. Model-based approaches

This methodology often consists of electrothermal battery models that were manually developed by a scientist, which requires a large amount of labor and knowledge. However, their main advantage comes from their low computational requirements and close-loop capabilities, making them favorable for BMS applications[13]. A popular method for improving the long-term prediction capabilities of these models is to implement a filtering algorithm to dynamically update the model's parameters[14].

As one of the first implementations of a modeling approach to predict the RUL of li-ion battery cells, Ramadass *et al.*[15] proposed a mathematical model used to predict the capacity fade. In a later publication, a combined semi-empirical and pseudo-2-dimensional model was then presented, again, by Ramadass *et al.*[16]. A generalized first principle model was expanded to simulate the aging behavior of lithium-ion batteries by Ning *et al.*[17]. These models still had significant limitations, mainly that they were restricted to the cell type, but

also to the charging and discharging protocols causing an inductive bias into the models' performance metrics since the temperature didn't play as important a factor as it should have. This is due to the fact that a higher electrical current flowing through the batteries will cause a higher temperature gradient. Because of this, an electrochemical-thermal model was developed by Li *et al.*[18] to address the cells' behavior under different temperature conditions. Then, to better represent the fact that SEI growth, being one of the primary components of capacity fade[19], upon the battery electrodes is heterogenous in nature, Methekar *et al.*[20] came up with a kinetic Monte Carlo simulation. Since then, multiple approaches have been presented to represent this heterogenous property, such as molecular dynamics[21] and density functional theory[22]. Limited by the computational complexity, these methods were later improved by Röder *et al.*[23] to reduce this drawback.

Equivalent circuit models (ECMs) also play an important role within the model-based RUL prediction literature. These implementations simulate the combination of circuit elements that would produce similar electrical behavior as a battery cell. These representations are able to describe the dynamic response of a battery through state-space mathematical modeling. Many publications have presented internal resistance growth models alongside filtering algorithms to update the parameters[14,24–26]. These models helped much future research develop ECMs that could simulate this impedance growth and use it as a parameter for the RUL prediction such as Saha *et al.*[27] and Li *et al.*[28]. To improve the filter's optimization, a dynamic feedback mechanism was developed[29], and in another article, a regularized auxiliary particle filtering approach was proposed[30] which improved the robustness of the method. However, some of these methodologies were using the internal resistance as an input parameter to the model, which would require the user to

conduct electrochemical impedance spectroscopy (EIS) in real-time which can be very time-consuming and not a viable strategy for electric vehicles' implementations. Because of this, modern ECM models need to estimate the modeling parameters using the limited knowledge of a BMS, such as the current, voltage, and temperature values. This is why Guha *et al.*[31] developed a fractional-order ECM to estimate the EIS for RUL prediction applications.

2.4. Data-driven approaches

This section will cover some of the advancements made within the battery remaining useful life prediction literature where data-driven approaches were implemented. These methodologies use monitoring data from battery cycling and develop algorithms that try to determine the relationship between the input features and their effect on the outputted RUL prediction. These models can be described mathematically with linear components, however, due to the large complexity and depth of these models, these methods can also be used to distinguish non-linear relations between variables. A variety of health indicators (His) are usually extracted from the raw cycling data and used as inputs to the models, typically from current, voltage, temperature, capacity, and cycling numbers[32].

Ng. *et al.*[33] developed a naive Bayes (NB) model for the prediction of li-ion remaining useful life using the NASA Ames battery cycle-life test data[34]. Wang *et al.*[35] implemented a support vector regression (SVR) based on extracted health indicators from temperature, current, and voltage values. An SVR was also modeled by Nuhic *et al.*[36] and Tao *et al.*[37] based on real-world drive cycle data. To accelerate the online training, an

incremental learning algorithm was proposed by Zhou *et al.*[38] based on the NASA dataset. Novel health indicators were proposed by Zhao *et al.*[39] to use as inputs to their SVR based on the charge and discharge voltage difference.

As an improvement over the SVR, the relevance vector machine (RVM) was proposed to increase the sparsity of the model and to provide probabilistic predictions[32]. Once again, a novel HI was proposed, this time based on the mean voltage falloff, to use with an RVM[40]. A work by Zhao *et al.*[41] proposed the use of a deep belief network to first filter and extract useful information from the data to then use as inputs to its RVM using the CALCE battery datasets. Another popular data-driven model within the RUL prediction literature is the Gaussian process regression (GPR). In a longstanding study, Goebel *et al.*[14] used a GPR to predict the end of life of various 18650 cells and also tested their method with a particle filter-based approach. They found that the PF, although computationally more complex, was able to achieve a higher prediction accuracy.

Finally, the emerging methodology that is neural networks has also had significant advancements in the past few years for battery management system implementations, mainly due to their various breakthroughs in regression and classification tasks within other fields. In an early publication, Parthiban *et al.*[42] developed a very simple artificial neural network, using only a handful of neurons, which managed to achieve satisfactory results. Still, Hai *et al.*[43] argued that the SVM was still able to provide better accuracy and a lower computational burden than the artificial neural network (ANN), especially when predicting the long-term performance of the batteries. Remaining in the vein of simple feedforward networks, Wu *et al.*[44] also presented an ANN for RUL prediction using voltage data. From this, a recurrent neural network (RNN) was developed[45] to take into consideration historical

cycling data alongside the latest. Due to the method's drawback of being unable to predict the RUL other than in a short-term scenario, a long short-term memory (LSTM) network was implemented[46]. Lately, more and more complex machine learning implementations have been developed, such as the dilated convolutional neural network (DCNN) by Hong *et al.*[47], which used this architecture to filter the raw cycling data to extract features which were then fed into fully connected layers to obtain an RUL prediction. Also, Pajovic *et al.*[48] used a regularized logistic regression to first classify the battery cells into two groups based on whether or not they had passed their respective capacity knee-point. Then, a multivariable linear regression (MLR) and multilayer perceptrons (MLP) were used to provide an RUL prediction. These combined machine learning architectures allow us to join the advantages of each individual model architecture and reduce their respective drawbacks.

2.5. Hybrid approaches

This section will cover methodologies proposed in the literature that made use of both model-based and data-driven methods. These approaches can help combat the drawbacks of each methodology, such as the model-based approaches' lack of flexibility and their need for expert knowledge, and the data-driven methods which require large amounts of data and have overall lesser robustness.

One of the main purposes of data-driven approaches within model-based methodologies is to update the model's parameters. To this end, He *et al.*[49] made use of the Dempster-Shafer theory and the Bayesian Monte Carlo method to initialize the particle filter's parameters and to update them, respectively. This allowed the reduction of the

convergence time and benefited the RUL prediction accuracy. To avoid sample impoverishment, which is a known issue of traditional PF algorithms, the SVR models were combined with particle filter[50,51]. Also to maintain the diversity of the filter's particles, Zhang *et al.*[52] devised a Markov Chain Monte Carlo method.

Another use for these data-driven algorithms is to predict upcoming measurements that can be used as inputs to the model, which should allow it to increase its accuracy when predicting future trends. With this in mind, Zheng *et al.*[53] and Liu *et al.*[54] each derived similar methodologies by using an RVM and an AR model, respectively, to realize a multi-step prediction of the battery capacity that is then used as input to the filtering algorithm.

2.6. Existing research gaps

From these publications, a few research gaps can be observed. Mainly, the need for accurate and reliable data has yet to have been completely solved. With the newer developments within artificial intelligence networks, such as the generative adversarial networks (GAN), it would be interesting to see whether useful data could be generated, otherwise, datasets consisting of real-life cycling data should be prioritized for future studies. Also, the computational complexity of the proposed methods is oftentimes an overseen drawback of the overall methodologies. For these methods to be useful within online applications, they need to be able to be functional on limited hardware and be easily implemented in a variety of applications.

Many novel publications usually present the battery prognostic prediction simply using a different architecture, or different health indicators to obtain more favorable results. This thesis instead proposes to completely change the current health indicator-based

methodology to replace it with, or add to it, image-based processing. Furthermore, considering that the data-driven methodologies are usually lacking behind the overall advancements within the machine learning field, it becomes imperative that novel data-driven battery research make use of its most recent advancements. This would allow battery studies to take advantage of these improvements and quickly implement these in the various cell prognostics that require prediction. It is for this reason that this thesis uses images within its methodologies since the vast majority of novel neural networks are image-based. Taking advantage of this very wide field of work is an asset for the battery and other regression-based areas of research.

Additionally, although complex theoretical methodologies are necessary for the advancement and research of any field, there is a general lack of implementable methodologies. There are currently real-life issues at hand that need to be addressed using more than only theoretical applications. The ease of access and ease of implementation is an often oversought, yet imperative, aspect of ingenuity. Some modern models have such high requirements that the proposed methodologies are simply out of reach for the majority of consumers, whether it be in terms of financial resources, computational hardware, or even expert knowledge, it should remain a priority to develop accessible methodologies.

2.7. Summary

In this chapter, a literature review was presented regarding the RUL prediction, mainly for lithium-ion batteries. Different methodologies were observed, and research gaps were then discussed. It was noticed that most methodologies could be summarized as being either model-based or data-driven, with a few modern publications combining these two

into hybrid approaches to battle the disadvantages of each method. Overall, the model-based procedure requires a significant amount of expert knowledge to develop and is generally less flexible than the data-driven methods. On the other hand, machine learning's ability to recognize the non-linear correlation between a given set of input variables and an output generally manages to outperform traditional model-based approaches in terms of accuracy. However, the quality of the dataset plays a vital role in the robustness of a developed data-driven methodology. It was also noticed that there is no standardization of the accuracy metrics presented within these studies. This in turn hinders future researchers' abilities to compare their work with existing literature and doesn't always give the readers a full scope of the models' performance. It should be made a priority to present consistent performance metrics throughout the field. This would also help prevent the abuse of certain metrics, which will be discussed in further detail in Chapter 4, where certain segments of the data are omitted to superficially increase the accuracy of the proposed methodology.

Chapter 3. Methodology

This chapter will cover some of the basics of machine learning implementations and will also present the methodology employed within this thesis. Firstly, neural networks are discussed, followed by the dataset used throughout this thesis. Afterward, the inputs of both developed networks are shown and then the models' architecture is presented.

The contents of this chapter were published in J. Couture & X. Lin[1, 2].

3.1. Neural networks

Neural networks are a type of machine learning methodology that was originally proposed by McCullough and Pitts in 1943[8] when they developed the first computational model of a neuron. However, their model had significant drawbacks, one the most significant one being only allowing Boolean inputs and outputs[55,56]. This is why in 1958, Frank Rosenblatt invented the perceptron, which has been widely adopted as the backbone of modern artificial intelligence technology[57]. These perceptrons can be configured in a single layer, which would only allow the network to learn linearly separable patterns, or they can also be arranged as a multilayer network, which can learn more complex relationships between the variables[58].

A multilayer perceptron network is also commonly known as a feed-forward, or fully-connected, neural network. The core components of these networks are the neurons, their biases, and the weights. Since these were inspired by human biology, the neurons of the network can be compared to the neurons of the brain, the weights represent the

interconnections between the various neurons, and the biases are used as a threshold value for the product of the value entering the neuron and the weight.

These were designed to simulate a human-like training ability, where repetition makes us remember and then allows us to react more adequately when encountering a given scenario once again. These pattern recognition abilities are learned by the backpropagation that occurs during the training of the networks, where the gradients of the error function are calculated and sent back through the network to update its various weights and biases[59].

Their uses have an enormously wide range of applications due to the plethora of novel architectures being developed in recent years, but most of these cases can be divided into two distinct sets, regression and classification tasks[60]. In the context of machine learning, regression is used to make numerical predictions from inputted data by learning the relationship between the given features and the desired outputs, whilst classification consists of classifying the given input into a predetermined set of classes, or groups. Although this thesis employs neural networks for a regression task, being the prediction of a battery's RUL, the transferred networks used were initially developed and trained to be used for image classification tasks.

3.1.1. Feed-forward neural network

The simplest form of a nonlinear neural network is one consisting only of fully-connected perceptrons. These kinds of networks are computationally efficient due to the simple calculations required, but this can vary greatly depending on the width and the depth of the network in question. A simple representation of a fully-connected neural network (FCNN) can be seen in Figure 2.

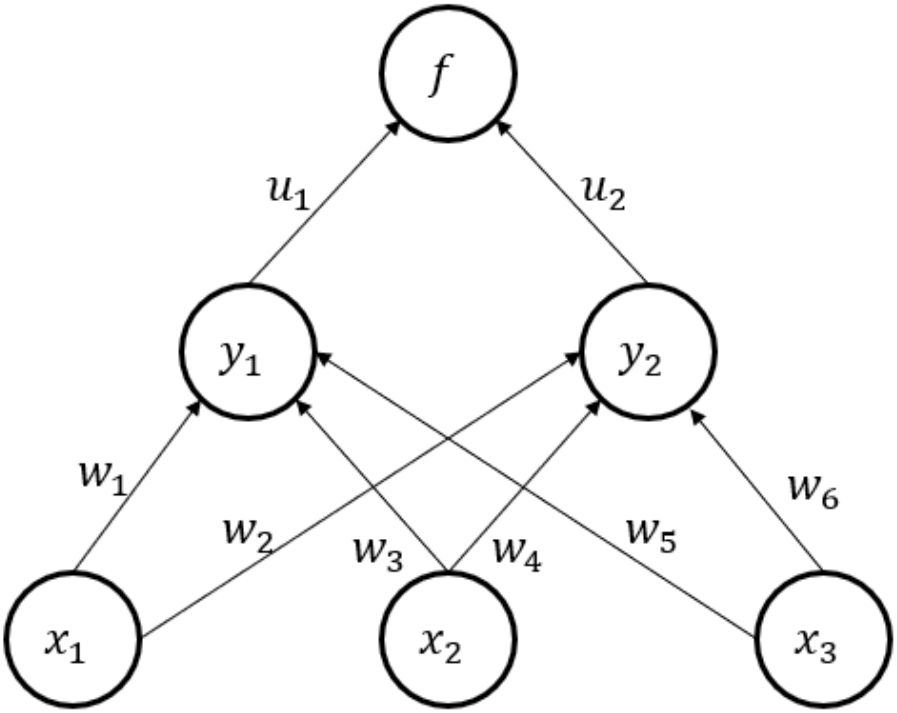


Figure 2. Example of a single hidden layer fully-connected neural network

In the network from the above figure, a series of inputs x are given to the perceptrons y_n, and the output f is obtained. Each neuron's input is multiplied by the weight of the connection between the neurons and then summed with its bias. The following equation is used to compute the outputted value of a neuron.

$$y_n = \overrightarrow{x_n} \cdot \overrightarrow{w_n} + b_n \quad (1)$$

where y_n is the n^{th} neuron's computed value, $\overrightarrow{x_n}$ is the vector of inputs, $\overrightarrow{w_n}$ is the vector of weights, and b_n is the n^{th} neuron's bias.

When a neural network begins its training, all of its weights and biases are randomly generated. The inputs are fed through the input layer and pass through the network, being multiplied and combined, increasing the complexity as it goes, until they eventually reach the output layer. Once there, a loss function is applied to compare the output received from the network and the desired output.

3.1.2. Backpropagation

Once the loss function has been applied to the output, backpropagation is then used to update the weights and biases of the network so that the prediction can more closely resemble the desired outcome. Backpropagation was invented in the 1970s but was only appreciated after Rumelhart *et al.*[61] published their paper "Learning Representations by Back-Propagating Errors" in 1986. To update the weights and biases of the network, gradient descent is used, which is an iterative optimization algorithm employed to minimize a given loss function[62]. In order to utilize backpropagation, three components are necessary[59]:

1) A feed-forward neural network, whose parameters are collectively denoted by Ω. The primary parameters of interest are, as mentioned previously, the weights $w_{i,j}^{k}$, which is the connection between the nodes i and j of layers k and $k+1$, and the biases, b_i^k, of node i in layer k.
2) A loss function, $E(X, \Omega)$, defining the error between the desired outputs $\vec{y}$ and the calculated outputs $\vec{f}$.
3) The dataset comprising of pairs of inputs and outputs $(\vec{x}, \vec{y})$.

With these components available, the update of the parameters can be presented by Equation 2.

$$\Omega^{t+1} = \Omega^{t} - \alpha \frac{\delta E(X, \Omega^{t})}{\delta \Omega} \tag{2}$$

where Ω^t denotes the parameters of the network at iteration t in gradient descent, and α is the learning rate.

This equation allows the network to find local optima within its search space, giving the impression of it learning by itself. This innate ability of machine learning given by backpropagation is what made possible the modern era of artificial intelligence.

3.1.3. Convolutional neural networks

Following the success of the original perceptron and the addition of backpropagation, the first convolutional neural network (CNN) was proposed in 1989 by LeCun *et al.*[63] that could recognize hand-written digits. These networks are composed of multiple layers of neurons, with the first few generally consisting of convolutional layers. With these, kernels, being matrices of weight values, are applied to a section of an image input, where the dot multiplication values are calculated and summed up to obtain a new matrix, called the feature map. Once the kernel has been applied to the entirety of the image, the feature map is completed and passed through the remainder of the network.

The kernels' dimensions can be altered to fit the needs of different inputs and applications. The smaller the matrix used, the smaller the reduction of parameters from the input image to the feature map. However, this has the benefit of allowing the use of more convolutions before the feature map becomes too small and the network starts having difficulty generalizing. This makes it difficult to standardize the development of neural network architectures since every application will require its own set of inputs and outputs, and no two tasks are the same.

The learning characteristic of convolutional layers makes them ideal for computer vision since the network can learn an optimal set of filters to apply to the dataset by learning which features are more useful. Although these can have applications within different scenarios, this thesis employs convolutions solely for their image filtering abilities.

It is common practice to subsequently apply multiple convolutional layers to the same image in order to increase the hierarchical decomposition of the input. This allows the network to learn more complex representations of the given features, such as starting from lines and curves to distinguish shapes, to being able of recognizing the combination of shapes that make up, for example, a vehicle, a train, or a boat. This concept of feature hierarchy was first presented as recently as 2012, in a paper written by Krizhevsky *et al.*[64], and kickstarted the modern era of deep learning.

3.1.4. Capsule networks

The breakthrough of the convolutional neural network cascaded an abundance of machine learning techniques and networks, and the ingenious capsule network was introduced to the public in 2017 in a paper written by Sabour *et al.*[65]. This new advancement sought to tackle an issue that was occurring with the use of traditional convolutional neural networks which was the loss of spatial information. It was quickly noticed from the early days of CNNs that datasets were needed to be able to identify certain features under different scenarios.

A few workarounds were developed to address these issues, such as pooling layers, which compress sections of a feature map into a single value, and data transformation, which are preprocessing techniques designed to inflate a dataset and make the network more robust to real-world data. Nevertheless, these methods are heuristic by nature and don't always work to the advantage of the network. This is why Sabour *et al.* developed the Capsule Network, which instead seeks to capture the parameters and the position of the features on the image. The usage of these capsules makes it more intuitive for the network to learn the relationship between the features, such as their position in relation to each

other. This innovation comes from the use of a routing algorithm that calculates coupling coefficients during the feed-forward process of the neural network. These coefficients are a measure of how likely a lower level feature will activate a given higher hierarchical feature.

Instead of making use of a traditional non-linear activation function, these networks instead make use of a squashing function which can be seen in Equation 3. This function is applied so that the output vector of a capsule is of length between 0 and 1, similarly to the SoftMax function.

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{3}$$

where v_j is the output vector of the capsule j and s_j is its total inputs.

The capsule layers' outputs are different from traditional CNNs due to them holding the instantiation parameters of the features, and not just scalar values. That being said, capsule networks still greatly benefit from the filtering abilities of convolutional layers, but the traditional use of pooling layers is discarded.

3.1.5. Transfer learning

Transfer learning techniques date back as far as 1976, in a publication by Bozinovski *et al.*[66] In their paper, they propose to mimic humanlike learning strategies, where we use information gained from previously solved tasks and apply the gathered knowledge to related applications[67]. This concept consists simply of transferring the entirety of the weights and biases of a previously trained neural network into a new one, where it can then be used as a starting point for training on a different dataset. This

significantly lowers the odds of overfitting the network to the dataset and also has the benefit of kickstarting the training process, making it less likely to get stuck in local minimums.

Regression applications have had limited usage of these transfer learning techniques due to most publicly available networks having been trained for classification applications. However, it is possible to completely reinitialize and retrain the final fully-connected layers and adapt a network to a different task. This is possible since the initial convolutional layers are mainly used for filtering data and feature extraction, whilst the fully-connected layers are used to triage these features and come up with a prediction. This allows us to import a trained image-recognition network and re-train the final few fully-connected layers to obtain RUL predictions.

3.2. Proposed methodology

This study firstly seeks to determine whether images of data curves can be made a viable option for battery prognostics prediction applications. Due to the high variance in electric vehicle battery usage, this study wanted to develop a rapid RUL prediction methodology to limit the amount of data required to obtain an accurate prediction. For this, 4 different datasets are developed, consisting of 10, 5, 3, and 1 consecutive cycle of data to obtain the optimal number of cycles required. A cycle in this case represents a complete battery charging and discharging cycle. The data is then separated into two distinct categories, the training, and testing datasets, holding 80% and 20% of the total dataset, respectively. The selected cells are kept consistent throughout the study to inhibit any data

leakage. This prevents the neural networks to have any knowledge of the testing dataset, allowing for a more accurate assessment of their performance.

A baseline neural network is firstly developed to validate the health indicator inputs that will then be combined with image inputs. This fully-connected neural network also has the task to quantify the benefits of image-based machine learning by comparing its accuracy with the other proposed methods. A correlation analysis is performed on various popular health indicators and the top performers are selected for this study.

A transfer learning architecture will then be proposed which makes use of pre-trained and publicly available convolutional neural networks. This architecture will help us determine whether image-based machine learning is a viable option for the prediction of batteries' RUL. This should at the same time provide a proof of concept that has a wide array of applications within other regression fields that make use of data curves.

A hybridized model of the health indicators and the transferred networks is then proposed. This methodology will serve to determine the direct improvements to be made by the simple addition of data curve images to the list of inputs. This implementation has little additional computational burden or preprocessing labor involved to the networks being already trained. These two methodologies are hybridized in parallel, with the convolutional outputs being combined with the health indicators just before the fully-connected layers that act as classifiers to the overall network.

This proposed methodology, being very successful on its own, warranted the development of a novel neural network architecture that would be solely trained on the data curves of this study. Due to the more intuitive learning of the capsule networks, this

architecture was adopted and revised to better match the applications of this study. Also, since these CapsNets have a better representation of the spatial information available within the images, this thesis sought to also determine whether or not these networks would be able to recognize and attribute numerical values to numbers on an image. A minor change to the input images is applied from the transferred networks to the capsule network.

Finally, once again due to the success of the previously mentioned methodologies, a transfer learning-based capsule network is developed with the help of the 4 generated datasets and imported and re-trained on the individual datasets. This should theoretically allow the feature extraction portion of the network to improve itself due to the larger available dataset, which in turn kickstarts the training process for the individual datasets and reduces considerably the computational requirements of the training process for online scenarios. These proposed methodologies managed to outperform many state-of-the-art RUL prediction methods, whilst requiring significantly less preprocessing labor and knowledge in the field.

3.3. MIT/Stanford battery dataset

Throughout this thesis, the MIT/Stanford battery dataset[68] was used to develop the machine learning algorithms. This dataset consists of 124 LFP battery cells that were discharged to their end-of-life, representing a 20% capacity degradation. Multiple charging protocols were used to observe their effect on the batteries' cycle life. It should be noted that within the scope of this thesis, having consistent cycling procedures would have been more beneficial to the accuracy metrics as it would have given the network fewer variations

to learn within the data. However, this would have negatively impacted the robustness of the developed method within real-life scenarios.

As previously mentioned, this dataset is comprised of the cycle life data from 124 commercial LFP/graphite cells from A123 Systems, model APR18650M1A[68]. These batteries have a nominal capacity of 1.1Ah and were cycled in a temperature-controlled environment at 30°C. Although their charging protocols varied, their discharging was consistent throughout the entire experiment with a current draw of 4.4A to 2.0V.

Another great aspect of this dataset, besides it being the largest collection of battery cycle life data of its kind[69], is that many parameters were being recorded and logged. Such parameters were, not limited to, the temperature, voltage, and current throughout the entire cycling process, and the internal resistance of the cells were also obtained at each cycle. This allows greater flexibility for battery research, as more thorough conclusions can be obtained from a parameter's impact on the lifetime of the cells. Although such an analysis falls outside the scope of this study, it would be interesting to try and compare different inputs for a standardized neural network, or even if hardware-permitting, to compress the entirety of the data within a single input and observe where the network learns its more valuable information.

3.3.1. Data preprocessing

The cycle life of a battery is the number of cycles it takes from the start of use of a cell to its EOL, representing a 20% capacity degradation. Although the dataset contains batteries with cycle lives ranging from 148 to 1935 cycles, as can be seen in Figure 3, only the cells within the 400 to 1175 range were included in an attempt to balance the variance. Also, several batteries were removed from the 400-500

cycle life range due to having significantly more cells than in any other range. These processes excluded 30 cells from this study.

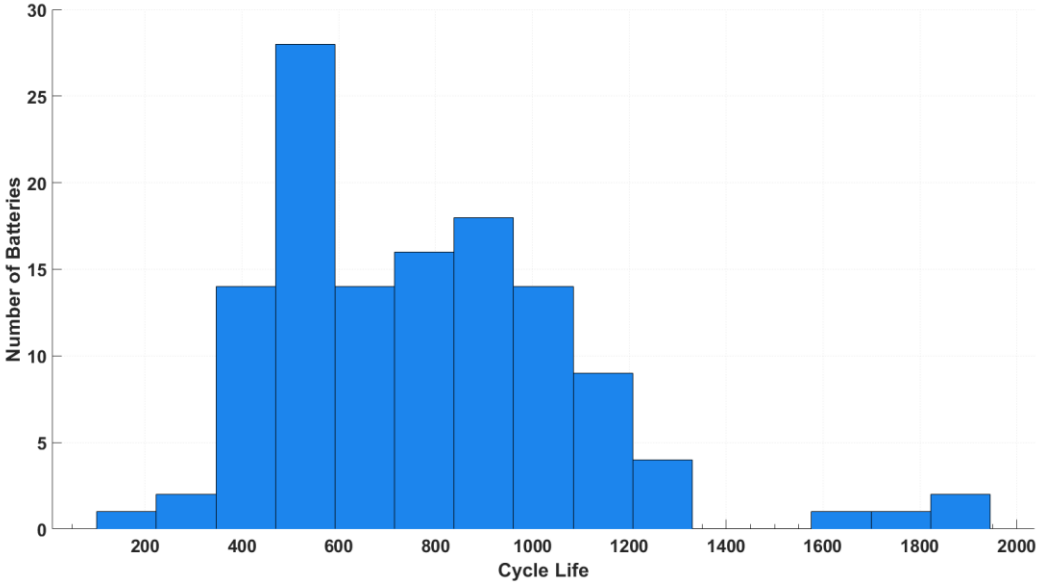


Figure 3. Cycle life distribution of the entire MIT/Stanford LFP batteries dataset[68]

Balancing the variance within the training data is a necessary step in the development of robust machine learning models since otherwise, the data-driven model would be more susceptible to batteries within the range that it is mostly being trained on. In the case of this study, the neural network would perform better on batteries within the 400-500 cycle life range but this would have negative effects on cells outside of this range. It Is even possible that keeping the distribution as it is would have yielded increased performance results, but these improvements would have been superficial and would have impacted the robustness of the trained network. The following figure presents the distribution of the cycle lives of the batteries used within this study.

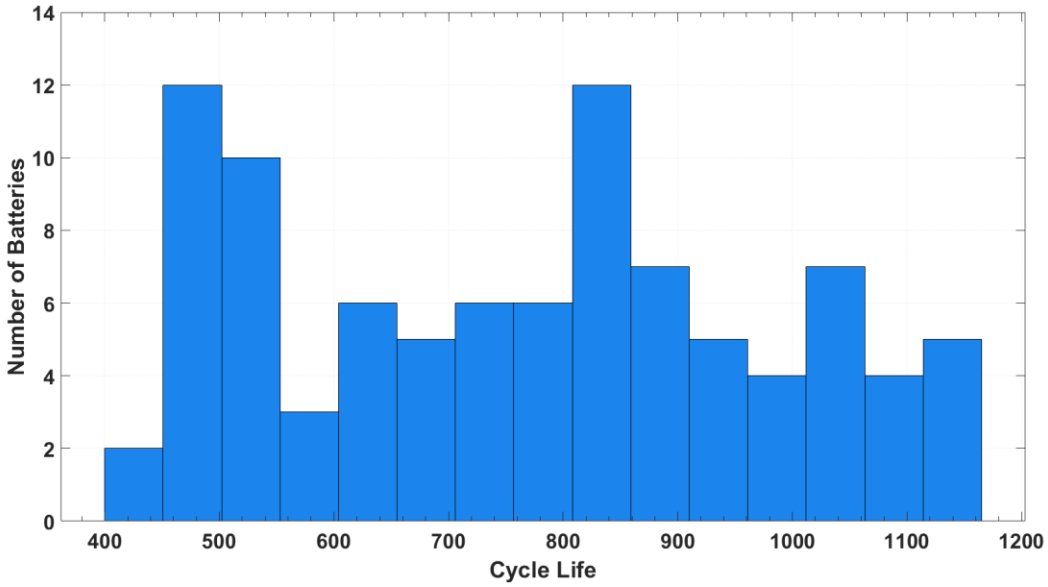


Figure 4. Cycle life distribution of the studied cells

3.4. Correlation analysis

Due to the previously mentioned large number of logged battery cycling parameters available within the MIT/Stanford dataset, it was necessary to cut down some of the possible inputs due to hardware limitations. For this task, a correlation analysis was performed on the possible parameters to determine the optimal set of inputs for the training of the networks. It should be noted that the correlation analysis performed within this thesis is not part of the proposed methodology. It is herein performed so that future research will not require to perform their own, and thus, will not be taken into account when comparing the overall processing requirements of the proposed methodology.

3.4.1. Pearson's correlation coefficient

To bring to term the correlation analysis of this study, the Pearson's correlation coefficient was used. This coefficient has been applied to a plethora of studies in the past years due to its ability to determine the linear correlation between a pair of variables. In

the case of this study, each possible input variable is compared with the RUL of the battery. The equation used to determine this correlation can be seen in Equation 4.

$$\rho(A,B) = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{A_i - \mu_A}{\sigma_A}\right)\left(\frac{B_i - \mu_B}{\sigma_B}\right) \quad (4)$$

where μ and σ represent the mean and standard deviation, respectively of the variables A and B.

The maximal values of the Pearson's correlation coefficient are -1 and 1, representing, respectively, a perfect negative and positive correlation between the two studied variables. A correlation coefficient of 0 would be indicative of absolutely no linear correlation between the pair of studied parameters.

3.4.2. Correlation results

Using the Pearson's correlation coefficient on both the RUL of each cell and the studied variable, this study was able to identify the optimal inputs that should be converted into images for the training and validation of the proposed methodology. Many health indicators were identified and compared, some of which have been proposed in previous studies, such as $dQdV$, $QdLin$, and $TdLin$, representing the capacity of the battery in terms of the voltage, the linearly interpolated discharging capacity, and the linearly interpolated temperature during discharge, respectively[68]. The $volt_t$ parameter was inspired by a publication by Hong *et al.*[47], and represents the time taken to reach 3.3 V from 3.15 V during charging.

Summary statistics, such as the mean, minimum, and maximum values, were also calculated to increase the number of possible health indicators. All of these statistical

numbers were transformed into single scalar values used to capture their change between 5 consecutive cycles. Additionally, to prevent any kind of bias introduced within the testing battery selection, each battery's correlation results were averaged and combined into a single correlation value. Figure 5 presents the results of this correlation analysis.

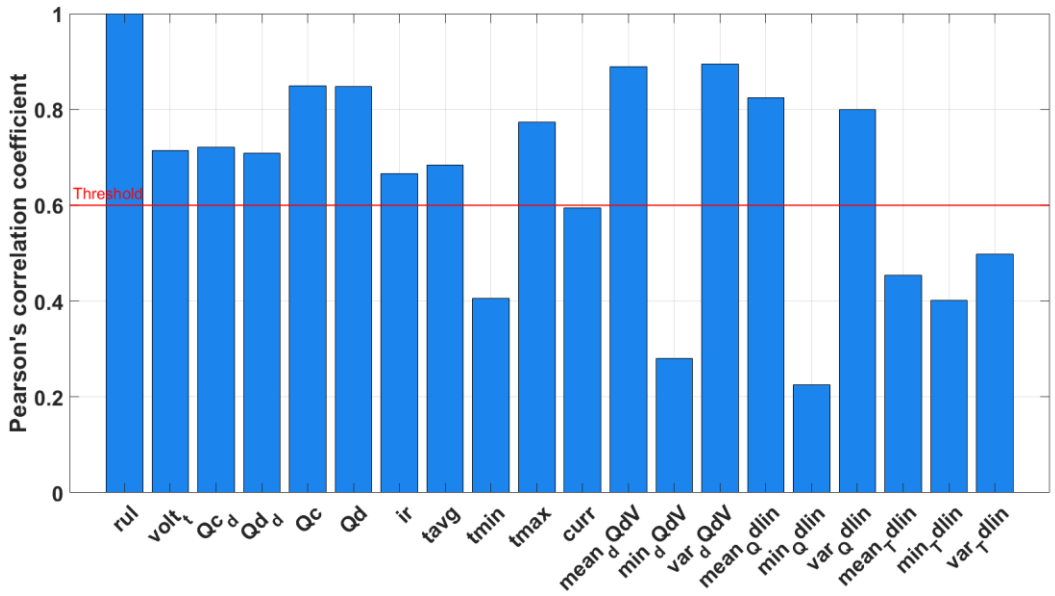


Figure 5. Pearson's correlation coefficient for the compared health indicators

A threshold value of 0.6 was selected to more easily distinguish the parameters having low- and high-correlation values. It can be clearly seen that multiple health indicators managed to achieve a satisfactory correlation coefficient, but since these were averaged over the entire selection of batteries, it is imperative to also consider the consistency of this correlation from cell to cell. This is because a high correlation for a few cells but only a mediocre correlation between other cells would still result in a satisfactory average of correlation. However, once the neural networks would come face-to-face with one of the lesser correlating batteries, the models would consequently perform poorly on these, which in turn would affect the overall robustness of the proposed methodologies. For this, the standard deviation is used to determine whether the correlation varied

considerably between cells, or if the values obtained in the above figure were consistent. The standard deviation equation can be seen in Equation 5 where x_i is the i^{th} value of the observed variable x, μ is the mean value of the entire x-sample, and N is the population's size.

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \tag{5}$$

The calculated standard deviation results are presented in Figure 6. For this purpose, a low standard deviation value is preferable, since this would indicate that the correlation coefficient values obtained previously are consistent from cell to cell. This consistency is very important for data-driven prediction methods since a single outlying battery within the dataset would be enough to hurt the performance of the network by having it update its weights and biases away from a minimum.

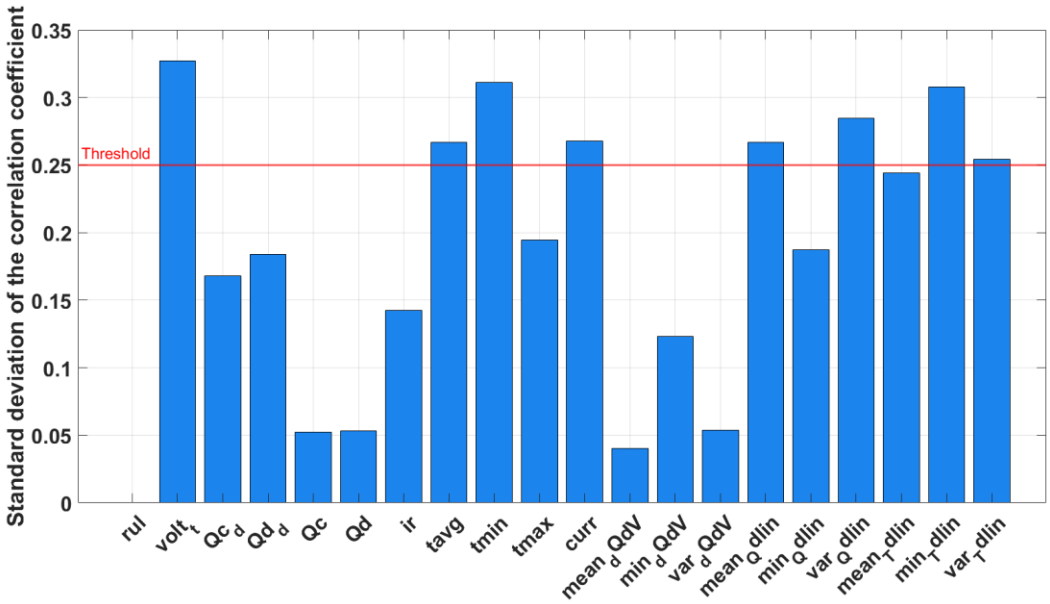


Figure 6. Standard deviation results of the correlation coefficient throughout the studied dataset.

From this bar plot, we can conclude that the temperature data have inconsistent correlation throughout the studied cells, this is true as well for the $volt_t$, and a few of the

statistical summaries of $QdLin$ and $TdLin$. On the other hand, the charging and discharging capacities, Qc and Qd, along with $dQdV$ have the lowest deviation between cell-to-cell correlation.

3.4.3. Inputs to the fully-connected neural network

Firstly, a fully-connected neural network will be developed to validate the health indicators chosen for this study. This network is one of the most basic architectures available and will also serve as a baseline to quantify the improvements possible by the addition of image-based machine learning. Since this neural network takes in numerical 2-dimensional data, it is important to preprocess the raw cycling data to make it compatible with the feed-forward process. This is why during the correlation analysis; all of the health indicators were gathered by compressing entire sequences of raw cycling data into summary values. In other words, this means that the 4 generated datasets, being the data from 10, 5, 3, and 1 consecutive cycles need to be compacted into a singular value.

A total of 12 health indicators were chosen as inputs to the baseline fully-connected neural network. As previously mentioned, these indicators are not time-dependent, and so do not require the need to be sequentially fed through the network. The chosen inputs for this study are: the average current over the final cycle of data shown, $curr$, this will ideally translate to the charging protocol used on the battery, since there is a large variety of these protocols within the dataset which all have their effect on the battery's degradation; Qc_d and Qd_d, in Ah, represent the change in charging and discharging capacity, respectfully, over the shown cycles of data; Qd is the discharging capacity value, in Ah, of the final cycle shown. The charging capacity has been discarded due to its high similarities with Qd which would have been redundant information; IR, which is the internal resistance value

at the final cycle shown; T_{max} and T_{avg} are, respectfully, the maximum and average temperature values over the final shown cycle. We opted to include both of these metrics since there was a significant number of outliers within the temperature values which could easily throw off the network with a single high maximal temperature value, ideally, the network will learn to disregard these outliers with the help of the average temperature during the cycle; As mentioned previously, $dQdV$ and $QdLin$, reflect the capacity of the battery in terms of the voltage and the linearly interpolated discharging capacity, respectfully. We opted to include the mean and standard deviation values of both of these health indicators due to their high correlation with the RUL; We also opted to include the $volt_t$ even though its correlation had a high standard deviation due to its success in the previously mentioned study[47]. For the testing of the 1 cycle dataset, only 10 health indicators are used as inputs since both Qd_d and Qc_d are removed due to having no known degradation between consecutive cycles.

3.4.4. Image inputs for transfer learning

To achieve the desired simplicity of the proposed methodology, limited data preprocessing is required to obtain the health indicators that will be used as inputs. For this reason, only commonly gathered data are used for the image-based networks, being the current applied, the voltage curves, and the capacity according to time. The Matlab programming software was used to generate all of the images used in this thesis.

Since this is a different neural network architecture that will use images instead of numerical values as inputs, the approach of showing consecutive cycles of data must be altered. This thesis proposes to use two different colors to indicate the first and the final curve of the sequence. To allow the network to visualize both of these colors, the generated

images contained 3 distinct matrix dimensions containing the RGB data. The data from the cycles bounded by these are discarded in an effort to prevent any significant overlap between the curves shown in the image. Evidently, the 1 cycle dataset will consequently only have a single curve shown on the image. An example of the information included within the transfer learning networks' inputs can be seen in Figure 7.

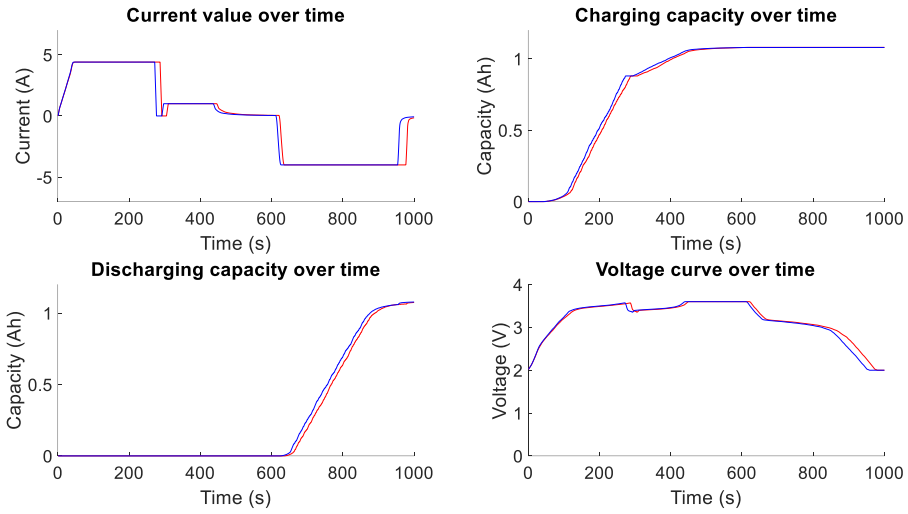


Figure 7. Visual representation of the degradation occurring within the image's input curves

Since a neural network would have no need for axes and titles, these are not included within the input images themselves. It should be noted that for the generation of the datasets, the axes values are fixed in place so that the curves move on the image with degradation. This is important so that the neural network can learn the baseline position of a healthy cell and this is what allows it to predict their RUL. An example of a fully-formatted input image can be seen in Figure 8. These input images have a picture resolution of 256x256 pixels to accommodate the publicly available networks' convolutional architecture. Due to the convolutional layer'' ability to reduce the feature map at each layer, the inputs need to be of a minimum dimension. These images were generated by simply plotting the curves on an appropriate image dimension and saving this image for later use.

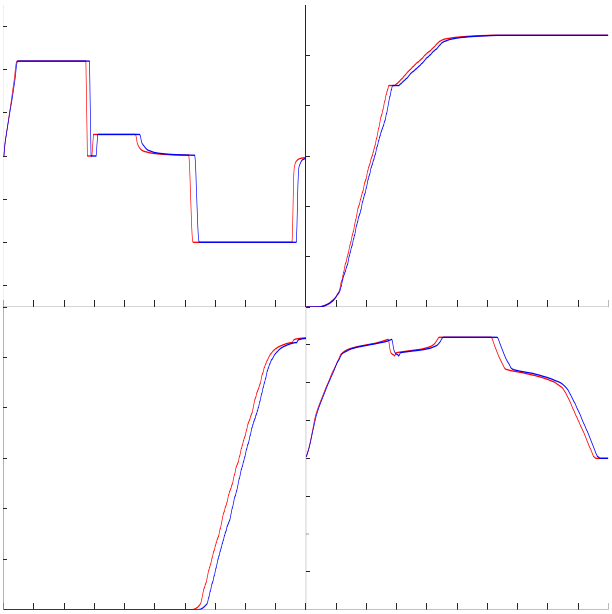


Figure 8. Image input example for the pre-trained transferred neural networks

3.4.5. Image inputs for the capsule network

Of course, the inputs for the capsule network will be of a similar fashion as those for the pre-trained networks, although this study wanted to take full advantage of a capsule network's ability to learn and draw conclusions from the features' spatial relationships. For this reason, numbers are added to the image to observe whether or not the CapsNet would be able to assign values to the numbers. This may seem like an easy task for a neural network as CNNs have already been proven to be able to recognize numbers, but it is important to remember that to assign a value to a sequence of digits, the order of these is of the utmost importance. For example, a CNN would be able to learn to recognize the numbers 3, 4, and 5, although it wouldn't necessarily know that it is seeing 345 or 453, which of course impacts the value of the number. Having a more intuitive spatial recognition ability, capsule networks should be well equipped to establish that the position of the digits is of great value to the overall prediction. For these reasons, the voltage curves are discarded from the input images in favor of the internal resistance and the charging

capacity of the final cycle of the sequence. An example of the inputs used for the capsule network is presented in Figure 9. The internal resistance and capacity values were normalized between 0 and 5, and 0 and 1, respectively. This allows the network to have a narrower number of values to learn.

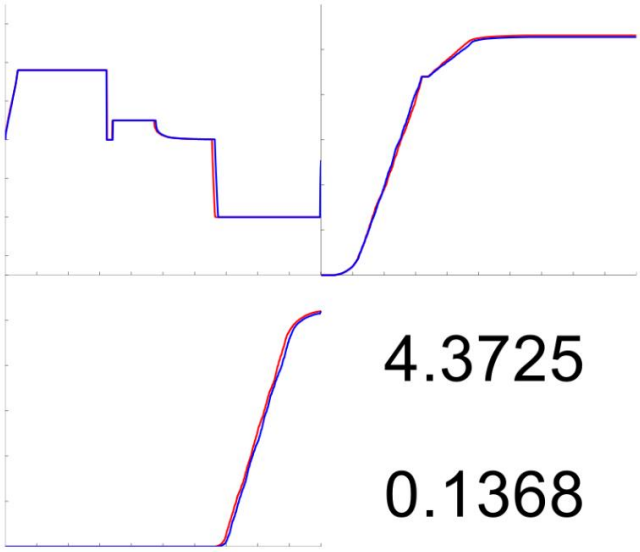


Figure 9. Image input example for the capsule network

Chapter 4. Model development

The contents of this chapter were published in J. Couture & X. Lin[1, 2].

4.1. Hyperparameters selection

The selection of hyperparameters within a neural network is crucial to the good functioning of the model. For this reason, this section will briefly cover the main parameters that were used within the training of these proposed methodologies.

4.1.1. Activation function

An activation function is a formula applied to a given layer in a neural network that defines the output of its nodes given a set of inputs. Various activation functions are made available through popular machine learning libraries which allow the easy implementation of these, such as the Sigmoid, Hyperbolic Tangent, SoftMax, and the Rectified Linear Unit (ReLU) activation functions. Each of these functions has its own applications and for this study, the ReLU activation was used throughout the networks presented in this thesis.

The ReLU activation function can be simply defined as follows:

$$f(x) = \max(0, x) \tag{6}$$

where x is the input to the node, and $f(x)$ is its output.

This function returns the input value but brings all negative inputs to zero. Some of the advantages of using ReLU are its computational simplicity, requiring only a max function, and its ability to output a true zero, an advantage that many of the other activation functions do not have. The ReLU's ability to obtain a sparse representation is sought after since it can accelerate the network's learning. Also, its ability to be a piecewise linear

function allows it to improve the model's ability to be optimized and to almost avoid the problem of vanishing gradients completely.

However, ReLU units can be fragile during training and can "die". This is caused when a large gradient flows through a ReLU neuron, causing it to update in such a way that the neuron would never again activate. There are a few ways to fix this common issue, such as implementing a Leaky ReLU activation instead, which outputs a small positive gradient for negative inputs ($y = 0.01x \; if \; x < 0$) instead of returning the zero value, or simply to lower the learning rate of the network to limit the neurons' ability to update.

4.1.2. Optimizer

Optimizers are mathematical functions used once again within neural networks that are dependent on a model's learnable parameters, such as the weights, biases, and learning rate, which help reduce the overall loss. The two most widely used optimizers are the Stochastic Gradient Descent (SGD) and the Adam optimizer. Adam, which stands for adaptive moment estimation, works by storing the first and second moments of the gradient to adapt the learning rate for each weight and bias of the neural network. Being an adaptive learning method, it computes the individual learning rates for different parameters. The mathematical representation of Adam can be seen in Equations 7 and 8.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{7}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{8}$$

where m and v are the moving averages, g is the gradient on the current batch, and β_1 and β_2 are constants defined by the user. Within the scope of this thesis, these constants were given the values of 0.9 and 0.999 respectively.

Once the moving averages have been computed, the optimizer starts working on updating the model's parameters, to do so, it performs the following operation on each weight:

$$w_t = w_{t-1} - \eta \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}} + \epsilon} \tag{9}$$

where w is the studied weight, η is the step size, and ϵ is a constant.

Adam is used in this study since it combines the advantages of Adagrad, which works well with sparse gradients, and RMSprop, which works well in online settings. It also has the convenience of being computationally efficient and requiring little memory space when compared to other optimizers.

4.1.3. Loss function

Loss functions are used to compute the loss gradients, which are then backpropagated through the network to update the weights and biases of the neurons. This loss is ultimately the function that the network will seek to minimize. This result can be defined as the quantitative measure of the deviation between the predicted output and the anticipated result. As most hyperparameters used for machine learning, many options are available, such as the Cross-Entropy, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), among many others. It is also possible to define a custom loss function, such as the one proposed by Sabour *et al.*[65] for their first implementation of the CapsNet.

This study proposed the use of the MSE loss, presented in Equation 10, which is commonly used for neural network applications due to it optimizing the robustness and

accuracy of the trained network. It does so by calculating the average square error between the estimated and the actual value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{10}$$

where n is the population's size, y_i is the estimated value, and $\hat{y}_i$ is the actual value.

This metric penalizes large errors more harshly than the MAE due to the squaring of the error, making the network more robust in the estimation of different data ranges and thus seeks to limit the reaction of the network to outlying data.

4.2. Models' architecture

4.2.1. FCNN architecture

Firstly, to act as a baseline model to compare the benefits of adding images to the inputs of traditional RUL prediction methodology, the fully-connected neural network is developed with the goal to be comparable to the existing literature in terms of performance. To do so, the architecture shown in Figure 10 was developed and Table I presents a summary of its hyperparameters.

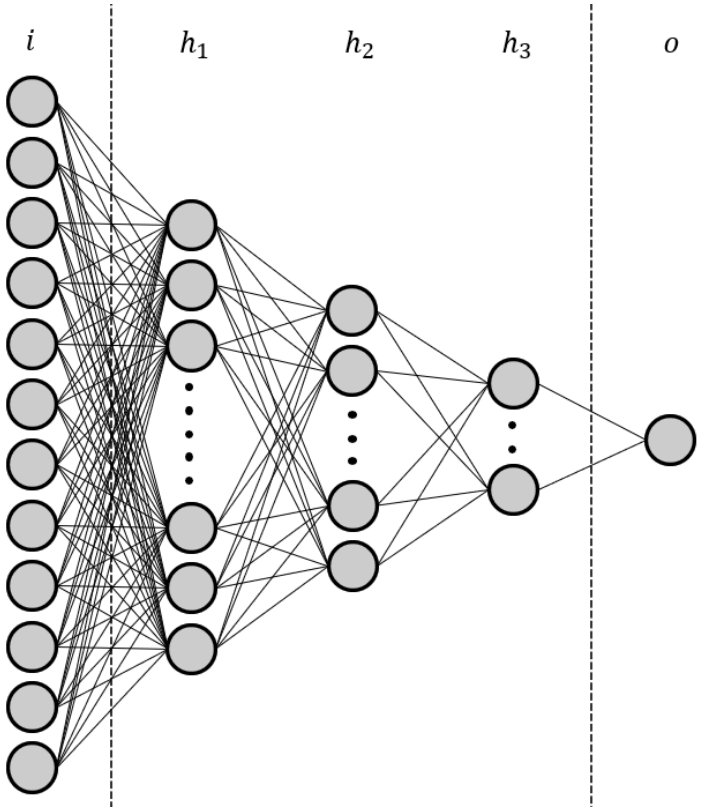


Figure 10. Proposed fully-connected neural network architecture

Table I. Fully-connected neural network's hyperparameters

Layer	$\boldsymbol{i}$	$\boldsymbol{h_1}$	$\boldsymbol{h_2}$	$\boldsymbol{h_3}$	$\boldsymbol{o}$
Number of neurons	12	2048	512	128	1

4.2.2. Transfer learning architecture

Following the development of the FCNN, the transfer learning architecture needed to be standardized to allow the comparison of the various available neural networks. The proposed architecture is presented in Figure 11.

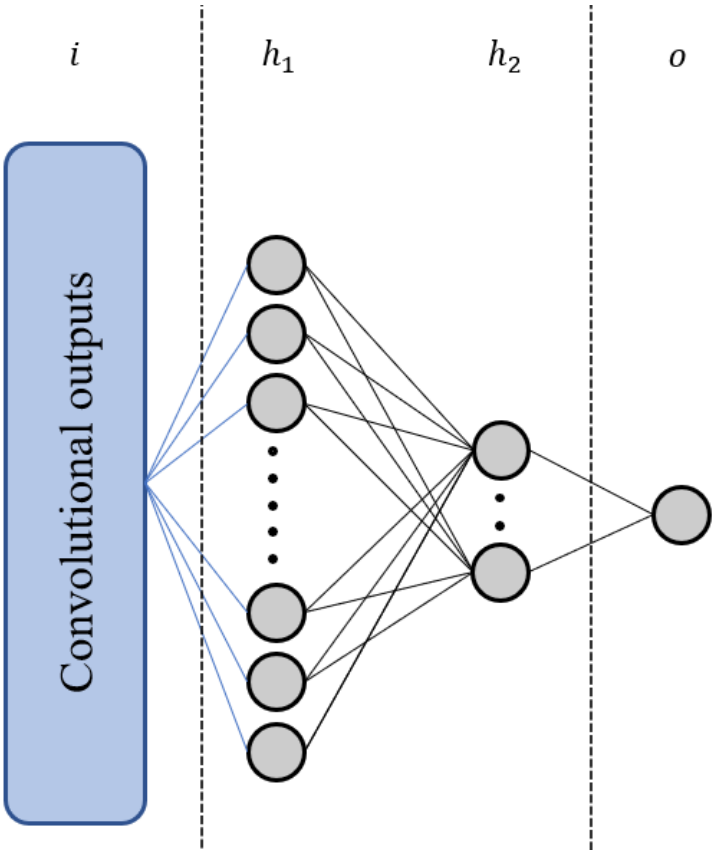


Figure 11. Proposed transfer learning architecture

Table II. Transfer learning network's hyperparameters

Layer	$\boldsymbol{i}$	$\boldsymbol{h_1}$	$\boldsymbol{h_2}$	$\boldsymbol{o}$
Number of neurons	Varying	512	128	1

The convolutional outputs of these transferrable networks all differ from each other. This is due to the fact that different convolutional and pooling dimensions are implemented within each network which causes the feature map to alter in size. In an attempt to counteract these discrepancies, the classifying section of the network was replaced with only 2 fully-connected layers to limit its complexity. This should translate to more robust testing of the feature extraction ability of the networks.

4.2.3. Transfer learning parallel hybrid

This section briefly overviews the proposed architecture of the transfer learning parallel hybrid (TLPH) methodology, which consists of combining the conventional health-indicator-based approach with these new recommended image-based inputs. To do so, the health indicators described earlier are combined with the convolutional outputs of the transferred networks and are then fed through the fully-connected layers together to obtain an RUL prediction output. The architecture can be seen in Figure 12.

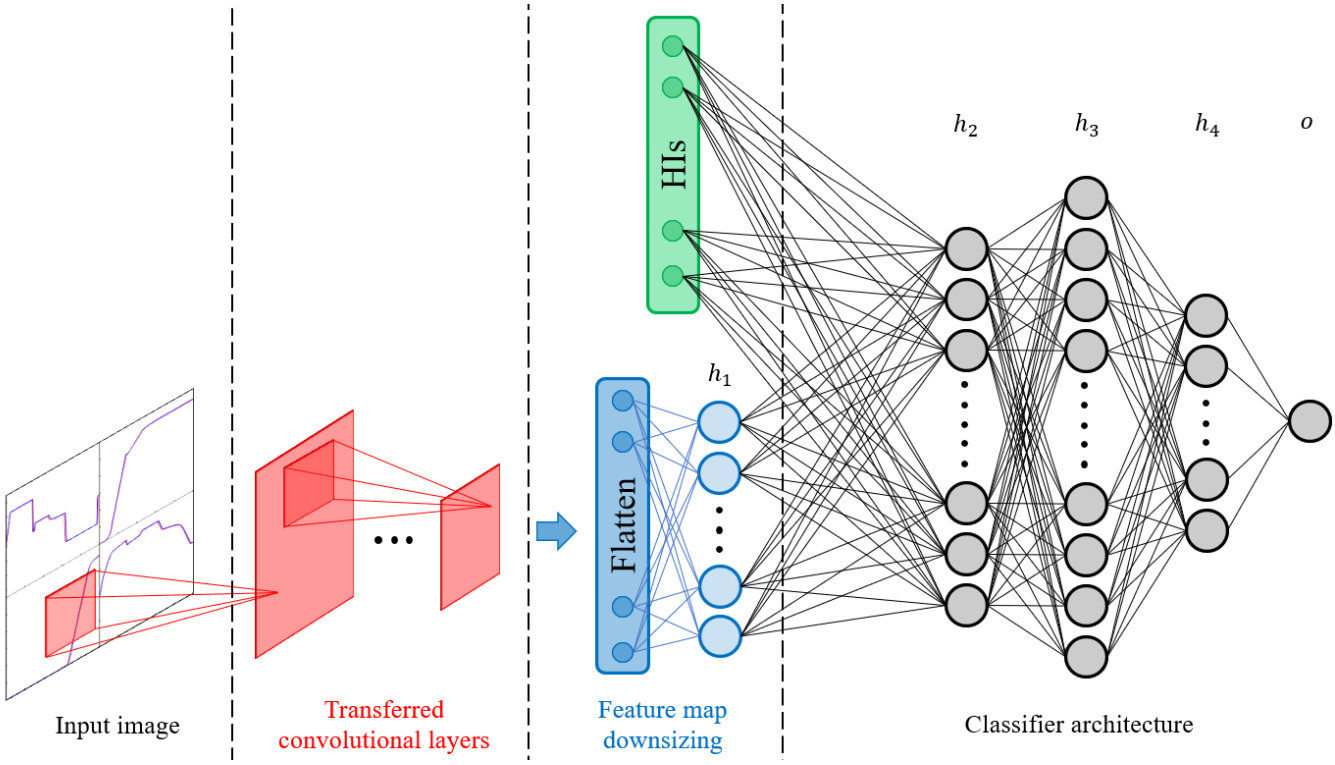


Figure 12. Proposed TLPH network architecture

Table III. TLPH network's hyperparameters

Layer	h_1	h_2	h_3	h_4	o
Number of neurons	256	512	1048	256	1

It can be seen that a single layer is added after the convolutional outputs but before their combination with the health indicators. This is due to some transferred networks

having a very large number of outputs which could have easily dominated the relatively small number of indicators used. With this same reasoning, a normalization layer was implemented between $h1 - h2$ to normalize the inputs to $h2$. This helps the following layer learn more independently from the previous one[70].

4.2.4. Capsule network architecture

Finally, to provide the proof of concept of the intuitive learning of the capsule networks and their applicability to regression tasks, the following architecture is proposed.

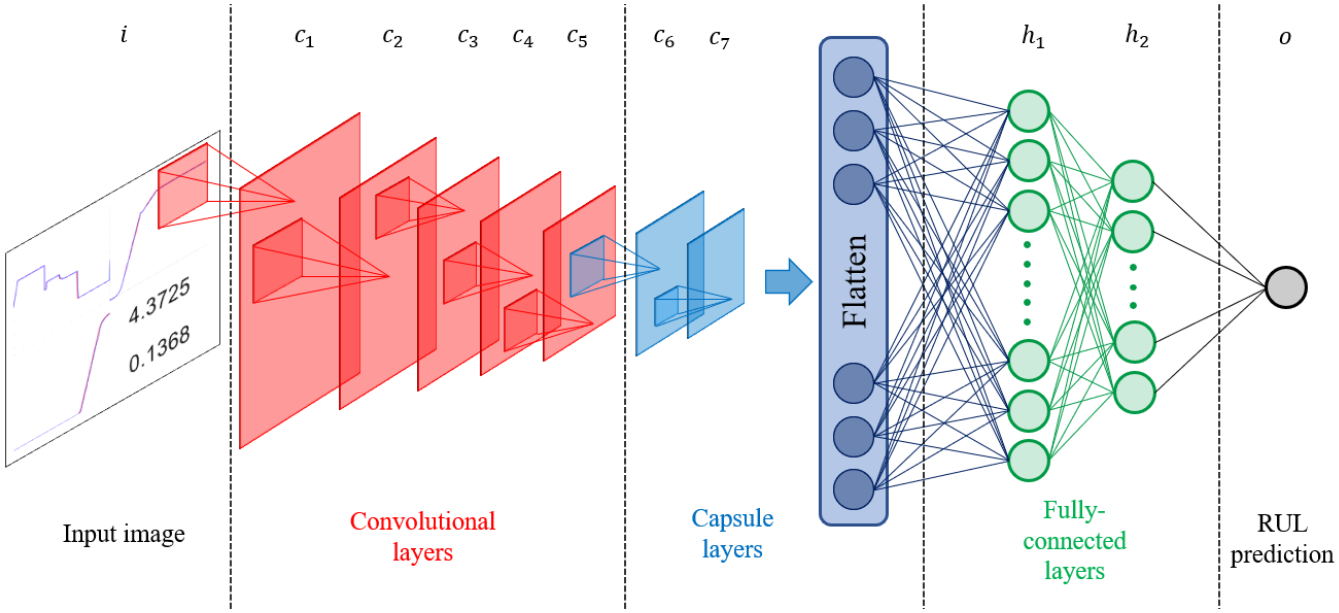


Figure 13. Proposed capsule network architecture

Table IV. Proposed capsule network hyperparameters

Layer	c_1	c_2	c_3	c_4	c_5	c_6	c_7	h_1	h_2	o
Matrix dimension / Number of neurons	5x5	5x5	3x3	5x5	5x5	3x3	4096	512	256	1

The original network developed by Sabour *et al.*[65] was tested on the relatively small images of the MNIST Handwritten Digit Classification Dataset[71] which consisted of 28 by 28 pixels images. Due to the low picture quality, only a few convolutional layers were originally implemented prior to the capsule layers. Since this study proposes the use of

significantly larger images, more convolutional layers are added to reduce the parameter space and increase the receptive field. The proposed layers are also of smaller sizes which should reduce the computational complexity of the network without hurting the overall accuracy[72].

4.3. Accuracy metrics

This section will cover the accuracy metrics employed with the neural networks that allow the performance assessment of the proposed methodologies and compare them with the available literature. Firstly, the mean absolute percentage error (MAPE), alongside the mean absolute error (MAE) and the root-mean-squared error (RMSE) are gathered from the networks on their respective testing datasets. These accuracy metrics are quite standard within the battery research literature, as well as the machine learning field. The MAPE can be seen in Equation 11.

$$MAPE(\%) = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{A_i - F_i}{A_i}\right| \tag{11}$$

where A_i and F_i are the i^{th} actual and predicted value, respectively, and n is the population's size.

This equation is very useful to provide the user with the overall percentage of the prediction error attained by the network, but it should be noted that this value can easily be skewed in favor of accuracy. Many publications within the literature neglected the prediction of the RUL of battery cells when the actual value was relatively low. This in turn has the effect of reducing the MAPE since it is a more penalizing metric when the actual value, A, is low. The metric then becomes superficial, as it no longer represents the

accuracy of the network over the entire field of predictions. It should be imperative for researchers developing machine learning implementations that the accuracy metrics used are not skewed in any way when presenting the results of their studies. The following accuracy metric, being the MAE, is presented in the following equation.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|A_i - F_i| \tag{12}$$

This metric, given in cycles, represents the mean prediction error of the network. This is a very beneficial metric but lacks the inclusion of the robustness of the network under a different range of data. It would be possible to have a very reasonable MAE value that would indicate a good average error, but this value might be calculated from an even better accuracy at high RUL values, but worst on low values, or vice-versa. This in turn requires the need for the RMSE, which penalizes large errors more harshly than the MAE, and is shown in Equation 13.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(A_i - F_i)^2} \tag{13}$$

Chapter 5. Results

This section will present the results of this study and a comparison of its proposed methodologies with state-of-the-art models from the literature. Firstly, the fully-connected neural network is trained and tested on various cells to obtain a baseline value of the potential accuracy of the health indicators. Then the transfer learning networks' results will be discussed, followed by the TLPH, and then the capsule network. A summary of the presented models' performance will then be presented.

The contents of this chapter were published in J. Couture & X. Lin[1, 2].

Three papers were selected from the literature for comparison to prove the robustness and effectiveness of our suggested methodologies. In this thesis, the term robust is used to describe a methodology that is able to perform accurately given a wide range of data, thus making it less susceptible to outliers. Severson *et al.* presented the Elastic Net, which used regularisation techniques to accomplish model selection and fitting simultaneously. Four-fold cross-validation and Monte Carlo sampling are used for hyperparameter optimization. The inputs to the networks are the key distinction between this methodology and the one provided in this thesis. Specifically, Severson *et al.* advocated that the first 100 cycles of a battery's lifetime be used rather than randomly picked cycles over the battery's life. As a result, the network has substantially less variability to learn within the inputted data and also has a relatively large number of inputs, both of which are important factors to consider when observing the high accuracy that was obtained. Because such a volume of historical data is unlikely to be available to the battery management system in real-world electric vehicle applications, the current study chose random cycles throughout the battery's life

cycle instead. Secondly, the article written by Hong *et al.* presented a dilated convolutional neural network (DCNN), which leverages this architecture to filter raw cycling data and extract useful features via the first CNN layers, which are then used as inputs to the subsequent fully-connected layers to obtain an RUL prediction. Their proposed model is very similar to the one present in this study, however, the leading discrepancy can be found within the inputs used. Hong *et al.* made use of large numerical arrays, whereas the methodology proposed within this thesis opted to use images of data curves in the form of pixel arrays. Finally, the model given by Sanz-Gorrachategui *et al.* will be used for comparison. These researchers employed a variety of health indicators to classify the observed battery into short- and long-RUL to then employ an appropriate "expert" model to forecast the output. The theory behind this model was that by doing so, the overall variability that each of the regression networks would have to learn would be reduced, resulting in higher prediction accuracy. The Multiple Expert System (MES) was proposed to be a combination of numeral smaller algorithms that when combined, constituted a single prediction pipeline.

5.1. Transfer learning results

5.1.1. Fully-connected neural network results

Using the architecture presented in Section 4.2.1, an RUL prediction accuracy of 82.05% was achieved with the vanilla fully-connected neural network with the use of the data from 10 consecutive cycles. The following table covers the results of this developed implementation alongside methodologies from the literature which made use of the same dataset.

Table V. Results of the FCNN on the health indicator inputs

Model	# of Trainable Parameters	Shown cycles	MAE (cycles)	RMSE (cycles)	Accuracy (%)
FCNN_1	1,137,409	1	66.74	108.9	81.26
FCNN_5	1,141,505	5	63.51	102.88	81.98
FCNN_10	1,141,505	10	62.57	109.23	82.05
Elastic Net[68]	Unknown	100	Unknown	214	89.3
DCNN[47]	2,393,468	4	65	Unknown	80.3
MES[48]	Unknown	10	Unknown	49	84.8

It is clear that the fully-connected neural network's performance is comparable to the existing literature, even overcoming some of the methodologies in some metrics. It should also be noted that although the Elastic Net model achieved a considerably higher accuracy, it did so at the cost of significantly more shown cycles. To the best of the author's knowledge, the previously fastest model was the DCNN, which this study managed to slightly outperform whilst using fewer cycles of data, making it faster and more accurate. This in turn validates the health indicators chosen for this study and provides a baseline accuracy to compare the improvements available from the hybridization of models.

5.1.2. Transferred neural networks results

This section will highlight the various results when using the generated image dataset solely on the pre-trained neural network. During this training, only the final fully-connected layers are reinitialized and trained according to the architecture presented in Section 4.2.2. The following tables cover the metrics of these networks over the different number of cycles shown on the image, according to the generated datasets. As previously mentioned, only a handful of the available networks are used within this study for comparison. Only top-performing models were selected to provide proof of concept.

Table VI. Results of the transferred networks on the 10-cycle image dataset

Model	Total # of Parameters	# of Trainable Parameters	MAE (cycles)	RMSE (cycles)	Accuracy (%)
AlexNet[73]	7,320,385	4,850,689	57.64	89.21	85.75
Resnet18[74]	11,570,753	394,241	66.12	101.75	80.85
Resnet50[74]	24,688,705	1,180,673	69.54	105.57	77.75
Resnet152[74]	59,324,481	1,180,673	68.42	102.47	82.41
VGG11[72]	22,197,633	12,977,153	54.14	83.83	86.62
GoogleNet[75]	6,256,289	656,385	64.83	97.22	79.10
EfficientNet[76]	65,229,777	1,442,817	69.55	104.32	79.77
DenseNet[77]	27,734,593	1,262,593	61.05	93.36	82.76

Table VII. Results of the transferred networks on the 5-cycle image dataset

Model	Total # of Parameters	# of Trainable Parameters	MAE (cycles)	RMSE (cycles)	Accuracy (%)
AlexNet	7,320,385	4,850,689	57.90	88.08	85.50
Resnet18	11,570,753	394,241	67.61	104.08	80.99
Resnet50	24,688,705	1,180,673	67.89	102.08	78.13
Resnet152	59,324,481	1,180,673	68.08	99.42	76.19
VGG11	22,197,633	12,977,153	55.60	84.89	86.32
GoogleNet	6,256,289	656,385	66.68	100.71	77.61
EfficientNet	65,229,777	1,442,817	70.83	107.44	78.50
DenseNet	27,734,593	1,262,593	60.02	92.03	82.89

Table VIII. Results of the transferred networks on the 1-cycle image dataset

Model	Total # of Parameters	# of Trainable Parameters	MAE (cycles)	RMSE (cycles)	Accuracy (%)
AlexNet	7,320,385	4,850,689	56.81	87.42	86.20
Resnet18	11,570,753	394,241	64.21	98.55	83.32
Resnet50	24,688,705	1,180,673	64.93	97.13	81.84
Resnet152	59,324,481	1,180,673	65.01	97.86	81.81
VGG11	22,197,633	12,977,153	54.99	85.05	86.86
GoogleNet	6,256,289	656,385	65.32	100.33	82.28
EfficientNet	65,229,777	1,442,817	69.63	105.23	80.40
DenseNet	27,734,593	1,262,593	61.15	93.48	84.18

It should be noted here that the large fluctuation in the number of total and trainable parameters is caused by the varying architectures of these networks. Even when keeping the final layers of these networks consistent, the number of trainable parameters still varies widely due to the size of the convolutional outputs.

Surprisingly, the transferred networks yielded superior accuracy by showing only a single cycle of data on the image, which makes this proposed methodology one of the fastest and most accurate methods to date. This is most likely due to having less overlap between the shown curves.

From these results, it can be concluded that if the existing methodology would completely discard the use of health indicators in favor of images of data curves, image-based machine learning would be able to accurately predict the RUL of batteries. The VGG11 network achieved the highest accuracy performance with 86.86%, which represents a 5.6% improvement over the baseline FCNN that was previously presented. Many of these image-based networks managed to outperform the baseline value using only the capacity, current, and voltage curves, indicating a very strong ability to learn from the

images shown. This is due to the strong adaptability of these networks which learn to separate the important information from the unvaluable one, and then categorize this information into prediction.

At this point, it would seem that there is a correlation between the number of trainable parameters and the prediction accuracy of the overall network. Although, it is interesting to observe that the total number of parameters does not seem to translate to any additional performance, such as with the EfficientNet and the Resnet152 achieving a lower performance than the benchmark FCNN.

A quick comparison is provided to emphasize the benefits of transfer learning for such an application in order to better demonstrate its advantages. To begin, the previously acquired findings for the AlexNet and VGG11 networks, which illustrate the use of transferred pre-trained weights in the neural networks, are compared with the same architecture but with randomly initialized weights. As a result, the entire network is trained on the batteries' pictures of data curves, resulting in a difference in the number of trainable parameters while the total number of parameters remains the same in both instances. The average time spent per training epoch was acquired using an NVIDIA RTX 3090 GPU, and the results in Table 9 are based on the 1 cycle dataset.

Table IX. Comparison of transfer learning models using pre-trained and randomly initialized weights

Model	# of trainable parameters	Percentage of the network to train (%)	MAE (cycles)	RMSE (cycles)	Accuracy (%)	Average time taken per training epoch (sec)
Transferred AlexNet	4,850,689	66.26	56.81	87.42	86.20	29.84
Randomly initialized AlexNet	7,320,385	100.00	60.05	86.77	85.37	34.22
Transferred VGG11	12,977,153	58.46	54.99	85.05	86.86	71.03
Randomly initialized VGG11	22,197,633	100.00	55.11	87.00	87.10	127.59

The accuracy of the networks for the pre-trained and randomly initialized models stays similar, based on these findings. This similarity is likely to fade if the dataset would be smaller, as the randomly initialized neural networks might not have enough data to properly train their complete architecture. This experiment, on the other hand, shows that pattern knowledge observed from the ImageNet training dataset is effective for detecting pictures of data curves and extracting usable information from them. It should also be noted that, when compared to their counterparts, the pre-trained models provided a far faster training and testing response time. Because computing hardware may be constrained, this is particularly useful for offline circumstances.

Furthermore, when using pre-made models, the pre-processing labor required to generate these complex architectures is completely removed, allowing them to be used by a much larger audience because they require significantly less in-depth knowledge to

develop. Transferring such models allows everyday people to benefit from huge technological businesses' considerable expertise and, more importantly, resources.

5.1.3. Transfer learning parallel hybrid results

This section will present the results achieved when combining both previously shown methodologies into a single parallel hybrid network. This novel model is now able to make use of numerical values, in the form of health indicators, and images. The combination of these two inputs should increase the overall information seen by the network and thus, theoretically, increase its prediction accuracy. The following tables show the accuracy metrics of the developed TLPH methodology when using images and health indicators as inputs to the network.

Table X. Results of the TLPH model on the 10-cycle dataset

Model	Total # of Parameters	# of Trainable Parameters	MAE (cycles)	RMSE (cycles)	Accuracy (%)
AlexNet	5,773,937	3,304,241	55.83	90.23	85.99
Resnet18	12,252,529	1,076,017	59.60	97.52	85.30
Resnet50	24,977,265	1,469,233	56.28	95.33	85.37
Resnet152	59,613,041	1,469,233	63.24	108.11	83.48
VGG11	16,587,953	7,367,473	51.04	79.71	87.20
GoogleNet	6,806,993	1,207,089	61.17	107.06	83.72
EfficientNet	65,387,265	1,600,305	68.15	131.69	81.30
DenseNet	27,982,193	1,510,193	58.24	99.32	84.73

Table XI. Results of the TLPH model on the 5-cycle dataset

Model	Total # of Parameters	# of Trainable Parameters	MAE (cycles)	RMSE (cycles)	Accuracy (%)
AlexNet	5,773,937	3,304,241	54.16	86.94	86.38
Resnet18	12,252,529	1,076,017	60.27	98.63	84.85
Resnet50	24,977,265	1,469,233	56.75	92.33	84.51
Resnet152	59,613,041	1,469,233	57.38	93.66	84.35
VGG11	16,587,953	7,367,473	50.60	79.07	87.15
GoogleNet	6,806,993	1,207,089	56.75	92.33	84.51
EfficientNet	65,387,265	1,600,305	67.03	111.79	83.27
DenseNet	27,982,193	1,510,193	57.05	88.52	84.68

Table XII. Results of the TLPH model on the 1-cycle dataset

Model	Total # of Parameters	# of Trainable Parameters	MAE (cycles)	RMSE (cycles)	Accuracy (%)
AlexNet	5,772,909	3,303,213	46.03	72.59	87.84
Resnet18	12,251,501	1,074,989	55.06	95.05	86.00
Resnet50	24,976,237	1,468,205	60.54	100.90	85.50
Resnet152	59,612,013	1,468,205	56.07	90.46	85.71
VGG11	16,586,925	7,366,445	47.67	73.35	87.98
GoogleNet	6,805,965	1,206,061	62.55	107.50	83.86
EfficientNet	65,386,237	1,599,277	69.59	121.49	82.08
DenseNet	27.981.165	1.509.165	52.74	83.32	85.64

The VGG11 network seems to have once again outperformed its competition, achieving an 87.98% prediction accuracy which represents an improvement of 6.72% over the benchmark FCNN that used only a single cycle of data as inputs. This performance is followed closely by the AlexNet, which achieved 87.84% accuracy but uses almost a third of the total number of parameters, making it a very strong contender when under strict hardware limitations.

5.2. Capsule network results

Following the success of the transfer learning method and its hybridization with the conventional health indicator method, a capsule network was developed and trained for the same task to determine whether there could be any significant improvements when using a network in which its convolutional layers were solely trained on data curves.

5.2.1. Capsule network results

This section will cover the results of the proposed CapsNet architecture when trained on the 4 generated image datasets presented in Section 3.4.5 one at a time, reinitializing the network between each dataset. The following table contains the accuracy metrics from this proposed methodology.

Table XIII. Results from the proposed capsule network model trained on each individual dataset

Cycles	# of Trainable Parameters	MAE (cycles)	RMSE (cycles)	Accuracy (%)
1	45,506,049	36.71	45.05	90.57
3	45,506,049	36.97	44.34	89.95
5	45,506,049	36.13	44.21	90.02
10	45,506,049	35.64	45.21	90.20

Interestingly, the results would indicate that the network is once again able to gather sufficient information from a single cycle of data to accurately predict the battery's RUL. Adding more data curves to the image negatively impacts the performance of the method, which is consistent with the findings of the transfer learning method. This would indicate that the best way to improve the overall accuracy of this methodology would not be to increase the number of shown cycles, unlike conventional methods, instead, performance gains might be possible from using different inputs or formatting.

5.2.2. Transfer learning capsule network results

Within this section, all four previously mentioned generated datasets for the capsule network are combined. This effectively quadruples the training dataset, which would be beneficial for any machine learning application. The reason for inflating the dataset at this point is not to achieve high accuracy, but instead, to better train the feature extraction portion of the network, which will then be available for transfer to a new network. The same training and testing datasets were used for each iteration of these methods, and so the testing batteries are never shown to the network during training. The following table presents the averaged results of the transferred capsule network when retraining the fully-connected layers using the individual datasets. This method allows the training of only 2.2 million parameters instead of the entire network, containing just over 45.5 million

parameters. This reduction in computational requirements represents a reduction of over 95% in training parameters, which significantly reduces the training time and kick-starts the training process for the individual dataset training.

Table XIV. Results from the proposed transfer learning capsule network model trained on each individual dataset.

Cycles	# of Trainable Parameters	MAE (cycles)	RMSE (cycles)	Accuracy (%)
1	2,229,249	35.62	41.50	90.85
3	2,229,249	38.03	43.97	90.01
5	2,229,249	35.57	41.09	90.49
10	2,229,249	38.85	44.64	89.60

It can be seen that the overall accuracy of the model remains relatively unchanged, at some points even increasing slightly. This means that the artificial dataset inflation can hold significant advantages for training only portions of the entire network, which can then reduce considerably the hardware limitations. For context, training this capsule network on an RTX 2060 Super graphics card took less than 45 minutes with the use of transfer learning techniques, and sometimes even within a single epoch. For comparison, the model from the previous section, which did not make use of transfer learning, took on average over 8 hours to train.

5.3. Comparison with literature

This section will briefly compare the results of this thesis with some previously mentioned studies taken from the literature. The best performer was taken from each methodology, representing the VGG11 network for both transfer learning models. These results can be seen in Table XIV.

Table XV. Results from this thesis compared with the literature

Model	# of Trainable Parameters	Shown Cycles	MAE (cycles)	RMSE (cycles)	Accuracy (%)
Elastic Net[68]	Unknown	100	Unknown	214	89.3
DCNN[47]	2,393,468	4	65	Unknown	80.3
MES[48]	Unknown	10	Unknown	49	84.8
Transfer Learning	12,977,153	1	54.99	85.05	86.86
TLPH	7,366,445	1	47.67	73.35	87.98
CapsNet	45,506,049	1	36.71	45.05	90.57
TL CapsNet	2,229,249	1	35.62	41.50	90.85

It is clear that these proposed models offer a great alternative to the existing battery RUL prediction methodology. The transfer learning capsule network model managed to achieve higher accuracy with a single cycle of data than the Elastic Net, which made use of the first 100 cycles of the battery's life. This significant increase in speed has huge potential for BMS applications and more accurate second-life classification algorithms.

That being said, the proposed TLPH also offers a good improvement margin at very little labor and computational cost. This method doesn't seek to replace the existing methodologies, instead, it adds itself in parallel to the current infrastructure. This also has a wide array of applications, even outside of battery research and in other regression fields.

5.4. Summary

To summarize the performance of the proposed methodologies, firstly, the fully-connected network served only to validate the extracted health indicators and to provide baseline performance metrics to quantify the advantage of parallelizing a pre-trained image-recognition network alongside traditional methodology. The transfer learning network on its own managed to outperform state-of-the-art models using only images,

proving that these can be used independently with great results, which requires limited in-depth knowledge of the field and pre-processing labor. The TLPH method was then developed to be even less computationally expensive than the transferred network and has the benefit of simply adding itself alongside existing methodologies. This proposed model doesn't claim to have the highest available accuracy, instead, it has the potential to simply increase the current performance of pre-existing models with little added labor and computational complexity. Then, a capsule network architecture was developed and proven to yield high RUL prediction accuracy performance, especially when artificially inflating the training dataset for later transfer of the feature extraction portion of the network. However, this model had significantly more trainable parameters when compared to other proposed models, making its transferring capabilities highly beneficial.

Chapter 6. Conclusion and Future Work

6.1. Conclusion

To conclude, the work within this thesis has significant potential within the battery prognostics prediction field, but also within other areas of regression applications where data curves can be derived from raw signals. The proven efficiency of pre-trained image recognition neural networks to recognize useful information from an entirely different dataset also has a wide array of functions. Firstly, publicly available transfer learning networks were imported and the final few fully-connected layers were retrained to predict the remaining useful life of the shown battery cells. This method was shown to outperform popular state-of-the-art methodologies, both in terms of accuracy and speed, requiring only a single cycle's worth of data. These models also have the often overseen benefit of requiring significantly less computational requirements due to not having to train the entirety of the network. This methodology was also proven to be functional in tandem with existing practices, by combining a health indicator methodology with the outputs of the convolutional layers, a slight increase in accuracy could be observed.

This methodology was then expanded to a capsule network architecture, which has increased spatial awareness when compared to traditional convolutional neural networks. Because of this, it was theorized that a capsule network would be more suitable to recognize numbers on a given image and assign them values. To validate this concept, numbers were added to the input image, and it was observed that the prediction accuracy increased. Although the transfer learning models delivered a more compact efficiency when taking into account their computational requirements, the capsule network delivered the most

robust, and accurate results to date. However, it should be noted that once trained, the capsule network can simply be transferred just the same to limit its computational requirements, allowing it to update in online settings. Using a larger dataset to train only the image extraction portion of the network also proved to bear more accurate prediction results, which is on par with overall data-driven models.

6.2. Future Work

Further improvements could be made to the work contained within this thesis by further testing and optimizing the inputs to the neural networks, alongside more advanced fine-tuning methodologies for the parameters of the networks themselves. It would be appealing to see whether these models could be made to predict various battery chemistries all at once, having a one-size-fits-all prediction model would be an asset for battery second-life applications. However, the developed models couldn't be validated on different chemistries other than LFP due to the lack of large battery cycling datasets and the absence of all the necessary input parameters. Future work should concentrate on the development of larger and more varied battery datasets to allow the development of more complex and robust machine learning models. Having real-life battery cycling data would also be an asset to the battery prognostics field, not only for RUL prediction, but also for state-of-health prediction, the development of efficient charging protocols, and integrated safety features. Also, the minimization of computational requirements and online updating capabilities should be made a priority to assure the applicability of the proposed methods. In addition, there must be some standardization within the accuracy metrics presented within regression fields to provide readers with a full scope of the efficiency of the proposed models.

References

1. Couture, J. & Lin, X. Novel Image-Based Rapid RUL Prediction for Li-Ion Batteries Using a Capsule Network and Transfer Learning. *IEEE Transactions on Transportation Electrification* 1 (2022) doi:10.1109/TTE.2022.3173918.

2. Couture, J. & Lin, X. Image- and health indicator-based transfer learning hybridization for battery RUL prediction. *Engineering Applications of Artificial Intelligence* **114**, 105120 (2022).

3. Global Emissions. *Center for Climate and Energy Solutions* https://www.c2es.org/content/international-emissions/ (2019).

4. Witt, J. Costs of Electric Car Battery Replacement. *Recurrent Auto* https://www.recurrentauto.com/research/costs-ev-battery-replacement (2022).

5. Ritchie, H., Roser, M. & Rosado, P. Emissions by sector. *CO2 and Greenhouse Gas Emissions* https://ourworldindata.org/emissions-by-sector (2020).

6. Pueyo, C. EV Batteries can reach 70% CO2 emissions reduction when they have a second life. *BeePlanet Factory* https://beeplanetfactory.com/en/2021/01/06/second-life-and-emissions-reduction/ (2021).

7. Pastor-Fernández, C., Bruen, T., Widanage, W. D., Gama-Valdez, M. A. & Marco, J. A Study of Cell-to-Cell Interactions and Degradation in Parallel Strings: Implications for the Battery Management System. *Journal of Power Sources* **329**, 574–585 (2016).

8. McCulloch, W. S. & Walter, P. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* **5**, 115–133 (1943).

9. Transfer Learning. *CS231n* https://cs231n.github.io/transfer-learning/.

10. Kabir, M. M. & Demirocak, D. E. Degradation mechanisms in Li-ion batteries: a state-of-the-art review. *International Journal of Energy Research* **41**, 1963–1986 (2017).

11. Demirocak, D. E. & Bhushan, B. Probing the aging effects on nanomechanical properties of a LiFePO4 cathode in a large format prismatic cell. *Journal of Power Sources2* **280**, 256–262 (2015).

12. Lipu, M. S. H. *et al.* A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: Challenges and recommendations. *Journal of Cleaner Production* **205**, 115–133 (2018).

13. Ge, M.-F., Liu, Y., Jiang, X. & Liu, J. A review on state of health estimations and remaining useful life prognostics of lithium-ion batteries. *Measurement* **174**, (2021).

14. Goebel, K., Saha, B., Saxena, A., Celaya, J. R. & Christophersen, J. P. Prognostics in Battery Health Management. *IEEE Instrumentation & Measurement Magazine* **11**, 33–40 (2008).

15. Ramadass, P., Haran, B., White, R. & Popov, B. N. Mathematical modeling of the capacity fade of Li-ion cells. *Journal of Power Sources* **123**, 230–240 (2003).

16. Ramadass, P., Haran, B., Parthasarathy, M. G., White, R. & Popov, B. N. Development of First Principles Capacity Fade Model for Li-Ion Cells. *Journal of The Electrochemical Society* **151**, (2004).

17. Ning, G., White, R. E. & Popov, B. N. A generalized cycle life model of rechargeable Li-ion batteries. *Electrochimica Acta* **51**, 2012–2022 (2006).

18. Li, J., Wang, L., Lyu, C., Wang, H. & Liu, X. New method for parameter estimation of an electrochemical-thermal coupling model for LiCoO2 battery. *Journal of Power Sources* **307**, 220–230 (2016).

19. Stamps, A. T., Holland, C. E., White, R. E. & Gatzke, E. P. Analysis of capacity fade in a lithium ion battery. *Journal of Power Sources* **150**, 229–239 (2005).

20. Methekar, R. N., Northrop, P. W. C., Chen, K., Braatz, R. D. & Subramanian, V. R. Kinetic Monte Carlo Simulation of Surface Heterogeneity in Graphite Anodes for Lithium-Ion Batteries: Passive Layer Formation. *Journal of The Electrochemical Society* **158**, (2011).

21. Leung, K. & Budzien, J. L. Ab initio molecular dynamics simulations of the initial stages of solid–electrolyte interphase formation on lithium ion battery graphitic anodes. *Physical Chemistry Chemical Physics* (2010) doi:DOI https://doi.org/10.1039/B925853A.

22. Leung, K. Electronic Structure Modeling of Electrochemical Reactions at Electrode/Electrolyte Interfaces in Lithium Ion Batteries. *The Journal of Physical Chemistry C* **117**, 1539–1547 (2013).

23. Röder, F., Braatz, R. D. & Krewer, U. Multi-Scale Simulation of Heterogeneous Surface Film Growth Mechanisms in Lithium-Ion Batteries. *Journal of The Electrochemical Society* **164**, E3335--E3344 (2017).

24. Saha, B., Poll, S., Goebel, K. & Christophersen, J. An integrated approach to battery health monitoring using bayesian regression and state estimation. in *2007 IEEE Autotestcon* 646–653 (2007). doi:10.1109/AUTEST.2007.4374280.

25. Laayouj, N. & Jamouli, H. Lithium-ion Battery Degradation Assessment and Remaining Useful Life Estimation in Hybrid Electric Vehicle. *Renewable Energy and Sustainable Development* **2**, (2016).

26. Guha, A., Patra, A. & Vaisakh, K. v. Remaining useful life estimation of lithium-ion batteries based on the internal resistance growth model. in *2017 Indian Control Conference (ICC)* 33–38 (2017). doi:10.1109/INDIANCC.2017.7846448.

27. Saha, B., Goebel, K., Poll, S. & Christophersen, J. Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework. *IEEE Transactions on Instrumentation and Measurement* **58**, 291–296 (2009).

28. Li, D. Z., Wang, W. & Ismail, F. A Mutated Particle Filter Technique for System State Estimation and Battery Life Prediction. *IEEE Transactions on Instrumentation and Measurement* **63**, 2034–2043 (2014).

29. Ahwiadi, M. & Wang, W. An Enhanced Mutated Particle Filter Technique for System State Estimation and Battery Life Prediction. *IEEE Transactions on Instrumentation and Measurement* **68**, 923–935 (2019).

30. Liu, J., Wang, W. & Ma, F. A regularized auxiliary particle filtering approach for system state estimation and battery life prediction. *Smart Materials and Structures* **20**, 75021 (2011).

31. Guha, A. & Patra, A. Online Estimation of the Electrochemical Impedance Spectrum and Remaining Useful Life of Lithium-Ion Batteries. *IEEE Transactions on Instrumentation and Measurement* **67**, 1836–1849 (2018).

32. Hu, X., Xu, L., Lin, X. & Pecht, M. Battery Lifetime Prognostics. *Joule* **4**, 310–346 (2020).

33. Ng, S. S. Y., Xing, Y. & Tsui, K. L. A naive Bayes model for robust remaining useful life prediction of lithium-ion battery. *Applied Energy* **118**, 114–123 (2014).

34. Saha, B. & Goebel, K. Battery Data Set. *NASA Ames Prognostics Data Repository* (2007).

35. Wang, S., Zhao, L., Su, X. & Ma, P. Prognostics of Lithium-Ion Batteries Based on Battery Performance Analysis and Flexible Support Vector Regression. *Energies (Basel)* **7**, 6492–6508 (2014).

36. Nuhic, A., Terzimehic, T., Soczka-Guth, T., Buchholz, M. & Dietmayer, K. Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *Journal of Power Sources* **239**, 680–688 (2013).

37. Tao, T. & Zhao, W. A support vector regression-based prognostic method for li-ion batteries working in variable operating states. in *2016 Prognostics and System*

Health Management Conference (PHM-Chengdu) 1–5 (2016). doi:10.1109/PHM.2016.7819787.

38. Zhou, J., Liu, D., Peng, Y. & Peng, X. Dynamic battery remaining useful life estimation: An on-line data-driven approach. in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings* 2196–2199 (2012). doi:10.1109/I2MTC.2012.6229280.

39. Zhao, Q., Qin, X., Zhao, H. & Feng, W. A novel prediction method based on the support vector regression for the remaining useful life of lithium-ion batteries. *Microelectronics Reliability* **85**, 99–108 (2018).

40. Zhou, Y., Huang, M., Chen, Y. & Tao, Y. A novel health indicator for on-line lithium-ion batteries remaining useful life prediction. *Journal of Power Sources* **321**, 1–10 (2016).

41. Zhao, G., Zhang, G., Liu, Y., Zhang, B. & Hu, C. Lithium-ion battery remaining useful life prediction with Deep Belief Network and Relevance Vector Machine. in *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)* 7–13 (2017). doi:10.1109/ICPHM.2017.7998298.

42. Parthiban, T., Ravi, R. & Kalaiselvi, N. Exploration of artificial neural network [ANN] to predict the electrochemical characteristics of lithium-ion cells. *Electrochimica Acta* **53**, 1877–1882 (2007).

43. Hai, H., Cui, N., Shang, Y. & Zhang, C. Aging performances and cycle-life predictions of Li-ion battery. in *2016 35th Chinese Control Conference (CCC)* 8710–8715 (2016). doi:10.1109/ChiCC.2016.7554748.

44. Wu, J., Zhang, C. & Chen, Z. An online method for lithium-ion battery remaining useful life estimation using importance sampling and neural networks. *Applied Energy* **173**, 134–140 (2016).

45. Eddahech, A., Briat, O., Bertrand, N., Delétage, J.-Y. & Vinassa, J.-M. Behavior and state-of-health monitoring of Li-ion batteries using impedance spectroscopy and recurrent neural networks. *International Journal of Electrical Power & Energy Systems* **42**, 487–494 (2012).

46. Park, K., Choi, Y., Choi, W. J., Ryu, H.-Y. & Kim, H. LSTM-Based Battery Remaining Useful Life Prediction With Multi-Channel Charging Profiles. *IEEE Access* **8**, 20786–20798 (2020).

47. Hong, J., Lee, D., Jeong, E. R. & Yi, Y. Towards the swift prediction of the remaining useful life of lithium-ion batteries with end-to-end deep learning. *Applied Energy* **278**, 115646 (2020).

48. Sanz-Gorrachategui, I. *et al.* Remaining Useful Life Estimation for LFP Cells in Second-Life Applications. *IEEE Transactions on Instrumentation and Measurement* **70**, 1–10 (2021).

49. He, W., Williard, N., Osterman, M. & Pecht, M. Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method. *Journal of Power Sources* **196**, 10314–10321 (2011).

50. Peng, X., Zhang, C., Yu, Y. & Zhou, Y. Battery remaining useful life prediction algorithm based on support vector regression and unscented particle filter. in *2016*

IEEE International Conference on Prognostics and Health Management (ICPHM) 1–6 (2016). doi:10.1109/ICPHM.2016.7542844.

51. Dong, H., Jin, X., Lou, Y. & Wang, C. Lithium-ion battery state of health monitoring and remaining useful life prediction based on support vector regression-particle filter. *Journal of Power Sources* **271**, 114–123 (2014).

52. Zhang, X., Miao, Q. & Liu, Z. Remaining useful life prediction of lithium-ion battery using an improved UPF method based on MCMC. *Microelectronics Reliability* **75**, 288–295 (2017).

53. Zheng, X., Wu, H. & Chen, Y. Remaining useful life prediction of lithium-ion battery using a hybrid model-based filtering and data-driven approach. in *2017 11th Asian Control Conference (ASCC)* 2698–2703 (2017). doi:10.1109/ASCC.2017.8287603.

54. Liu, D. *et al.* Lithium-ion battery remaining useful life estimation based on fusion nonlinear degradation AR model and RPF algorithm. *Neural Computing and Applications* **25**, 557–572 (2014).

55. Chandra, A. L. McCulloch-Pitts Neuron — Mankind's First Mathematical Model Of A Biological Neuron. https://towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1 (2018).

56. Kumar, N. McCulloch Pitts Neuron — Deep Learning Building Block. https://medium.com/hackernoon/mcculloch-pitts-neuron-deep-learning-building-blocks-7928f4e0504d (2019).

57. Lefkowitz, M. Professor's perceptron paved the way for AI – 60 years too soon. *Cornell University* https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon (2019).

58. SimpliLearn. What is Perceptron: A Beginners Guide for Perceptron. https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron#:~:text=A Perceptron is a neural network unit that does certain,value "f(x). (2022).

59. McGonagle, J. *et al.* Backpropagation. https://brilliant.org/wiki/backpropagation/#:~:text=Backpropagation%2C short for %22backward propagation,to the neural network's weights.

60. Ramos, D. Real-Life and Business Applications of Neural Networks. https://www.smartsheet.com/neural-network-applications (2021).

61. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).

62. Kwiatkowsky, R. Gradient Descent Algorithm — a deep dive. *Towards Data Science* https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21#:~:text=Gradient descent (GD) is an,e.g. in a linear regression). (2021).

63. le Cun, Y. *et al.* Handwritten digit recognition: applications of neural network chips and automatic learning. *IEEE Communications Magazine* **27**, 41–46 (1989).

64. Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. in *Advances in Neural Information Processing Systems* (2012).

65. Sabour, S., Frosst, N. & Geoffrey E., H. Dynamic Routing Between Capsules. *Advances in Neural Information Processing Systems* **30**, (2017).

66. Bozinovski, S. & Fulgosi, A. The influence of pattern similarity and transfer of learning upon training of a base perceptron B2. *Proceedings of Symposium Informatica* (1976).

67. Kocer, B. & Arslan, A. Genetic Transfer Learning. *Expert System with Applications* 6997–7002 (2010).

68. Severson, K. A. *et al.* Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy* **4**, 383–391 (2019).

69. Hasib, S. A. *et al.* A Comprehensive Review of Available Battery Datasets, RUL Prediction Approaches, and Advanced Battery Management. *IEEE Access* **9**, 86166–86193 (2021).

70. Dwivedi, R. Everything You Should Know About Dropouts And BatchNormalization In CNN. *DEVELOPERS CORNER* https://analyticsindiamag.com/everything-you-should-know-about-dropouts-and-batchnormalization-in-cnn/ (2020).

71. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 2278–2324 (1998).

72. Zisserman, A. & Simonyan, K. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv: Computer Vision and Pattern Recognition* (2015).

73. Krizhevsky, A. One weird trick for parallelizing convolutional neural networks. *arXiv* (2014).

74. He, K., Zhang, X., Ren, S. & S., J. Deep residual learning for image recognition. *arXiv* (2015).

75. Szegedy, C., Liu, W., Jia, Y. & S., P. Going deeper with convolutions. *arXiv* (2014).

76. Tan, M. & Le, Q. v. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* (2020).

77. Huang, G., Liu, Z., Maaten, L. van der & W., K. Q. Densely connected convolutional networks. *arXiv* (2018).