Singapore Institute of Technology - University of Glasgow
Joint Degree in Computing Science Degree Programme

CSC3001 Capstone Project

Please complete the following form and attach it to the Capstone Report submitted.

Capstone Period: 28/08/2022 to 14/07/2023
Assessment Trimester: Final Trimester
Project Type: Academic

Academic Supervisor Details:
Name: Zhang Wei
Designation: Assistant Professor
Email Address: Wei.Zhang@singaporetech.edu.sg
Contact Number: +65 6592 2214

Student Particulars & Declaration:
Name of Student: Ng Dennison
Student ID: 2000591

I hereby acknowledge that I have engaged and discussed with my **Academic Supervisor** on the contents of this Capstone Interim Report and have sought approval to release the report to the Singapore Institute of Technology and the University of Glasgow.

Signature Dennison

Date: 14/07/2023

END OF FORM

Singapore Institute of Technology - University of Glasgow
Joint Degree in Computing Science Degree Programme

Final Capstone Report
Battery Image Classification: A Transfer Learning Approach

For **Final Trimester** from 01/05/2023 to 14/07/2023

Ng Dennison
Student ID: 2000591

Academic Supervisor: *Zhang Wei*

Submitted as part of the requirement for CSC3001 Capstone Project

Table Of Content

List of Figures

List of Tables

Acknowledgments

This project would not have been possible without the support of many people. I would like to thank my academic supervisor, Professor Zhang Wei, for his guidance, patience, encouragement, and assistance during my capstone project. I would also like to thank Syabil and Zheng Yi for their constructive recommendations during the implementation stages of this project.

1. Summary

The sorting of batteries from waste electrical and electronic equipment (WEEE) is currently being performed manually by human hands in recycling plants. This is due to the large variety of battery cells currently available in the market, each being used for various appliances. This ranges from small button cell batteries to larger lithium-ion battery packs such as those found within modern smartphones and laptops. As the amount of WEEE over time increases due to the increasing number of devices available on the market, the need for a more efficient sorting system is required.

Therefore, the feasibility of classifying waste batteries through image classification with deep learning is investigated. In this paper, the image classification of various popular brands of dry cell batteries is investigated. This serves as a good starting point for the feasibility of scaling up this research to be able to classify specific battery types or makeup. A huge challenge was the lack of domain specific practical battery images available on the internet. Therefore, a decision to merge datasets by battery brands and to use dry cell batteries was made. Techniques and approaches such as oversampling and transfer learning was also proposed to increase the data diversity and counteract the lack of available data by using pre-trained models as a base.

The results are promising, with an achieved weighted average f1-score of 0.86 for the classification of 8 different battery brands trained with a total of 855 images using the ResNet50V2 model as a base for transfer learning. A weighted average f1-score of 0.84 was also achieved using the MobileNet model as a base. An inference test of both models using an edge device was also performed using a Raspberry Pi 4 Model B (8GB RAM) and OpenCV. There were good results from the inference test of both models for brands that had great practical images within their datasets (e.g., Energizer, Ikea) whereas the inference model had difficulty making correct predictions when presenting it other brands that did not have as high a quality dataset. At the end, the MobileNet model was chosen as the better model on the edge-device due to its much lower inference time and smaller model size compared to the ResNet50V2 model whilst still being able to achieve a weighted average f1-score of 0.84 which is extremely close to the targeted weighted average f1-score of 0.85. This demonstrates the potential use of image classification models in the process of battery sorting automation within recycling plants.

2. Introduction

2.1. Problem Formulation

The recycling of batteries is a complex and challenging process. These batteries must be first sorted by type and then disassembled to extract the valuable materials contained within the battery cells. This process is often manual and labour-intensive, which can lead to error and safety hazards. It has also been shown that the collection of waste batteries and accumulators in the EU has steadily increased from 55 000 tonnes in 2010 to 88 000 tonnes in 2018 [1]. Likewise, the collection rate of portable batteries and accumulators has increased form 35% in 2011 to 48% in 2018 [1].

In [2] and [3], we can see that manual sorting by human hands is still a required step in the process of sorting batteries before recycling them. Therefore, with the increasing amount of battery waste, a more efficient and accurate way to sort batteries must be selected. To do so, computer vision and machine learning can be used to improve the efficiency and accuracy of battery recycling. For example, computer vision and machine learning can be used to automatically sort batteries by type. Upon knowing the type or brand of the battery, a database with records of each battery can be queried and used to determine the optimal disassembly process for each battery type.

In [4] – [6], several challenges are presented when it comes to battery recycling. These challenges will be described and discussed in the following paragraphs.

One of the challenges is the lack of standardization in the battery recycling processes. The papers note that there is no single, standardized process for recycling batteries as each type of battery requires a different recycling process depending on the materials used to make the battery. This can make it difficult to recycle batteries efficiently and effectively as batteries must first be sorted before being recycled using each battery's recommended recycling process. This challenge is also the greatest motivating factor for the conduction of this paper.

There is also a high cost associated with battery recycling due to the complexity of the recycling process and the low value of some of the recycled materials as discussed in the papers.

Last but not least, there is a negative environmental impact associated with the recycling of batteries as several harmful pollutants are released into the air and water during the recycling process.

2.2. Project Objectives/Design Specifications

2.2.1. Project Objectives

The primary objective of this project is to develop an image classification model capable of accurately identifying and categorizing dry cell batteries based on their brands. This objective aligns with the future need for an efficient and reliable solution for battery sorting in the waste recycling industry. By the end of the project, the aim is to achieve a weighted average f1-score of at least 0.85 on a test dataset consisting of dry cell battery images from multiple brands.

Furthermore, another objective of the project is the implementation of the image classification model on an edge device, specifically the Raspberry Pi 4 Model B (8GB RAM), to test the feasibility of the model for the use case of real-time inference for battery sorting. This objective aims to enhance the practicality and accessibility of the system by deploying it on a portable and cost-effective platform. The model will still be optimised for the highest accuracy possible within the duration of this project with the usability being tested on an edge-device.

2.2.2. Design Specifications

To measure the success of the final design in meeting the objectives and addressing the design problem, the following design specifications were defined.

Firstly, the image classification model should be able to achieve a weighted average f1-score of at least 0.85 on an unseen test dataset of battery images. This specification ensures that the model can classify batteries based on their brands.

Secondly, the model should be capable of classifying a single battery image in less than 1 second given an inference frame on the Raspberry Pi 4. This specification emphasizes the need for real-time performance, enabling efficient battery sorting without significant delays even if the model has not been optimised for the performance of real-time detection when deployed on an edge-device.

Thirdly, the size of the trained model should be limited to a maximum of 100 MB to ensure efficient storage and deployment on the edge device. This specification addresses the constraints of limited storage capacity on the Raspberry Pi 4, ensuring the practicality and feasibility of deploying the model.

Lastly, the design should be scalable to accommodate potential future addition of data or updates to the classification model. This specification highlights the need for a flexible and adaptable system architecture that can easily integrate and accommodate additional data added in the future should the need arise for the model to be retrained.

These design specifications provide clear and measurable metrics to assess the success of the final deliverable in meeting the aforementioned project objectives and addressing the identified constraints and requirements. By adhering to these specifications, the development of an effective and practical image classification system for battery sorting can be ensured.

3. Literature Review

3.1. Use of Computer Vision and Machine Learning in the Sorting of Batteries and E-Waste

In [7], the paper discusses the use of computer vision and optical character recognition to aid in the classification of batteries. The authors discuss the feasibility of classifying waste batteries by first recognizing text on an image of the battery with deep learning object character recognition (OCR), before proceeding to compare the extracted text with a database of known text information representing various different battery types.

The results from the paper are shown to be promising, with an achieved precision of 0.98 and recall of 0.67 for classification by battery group. This helps to show the use of computer vision and machine learning in the automated sorting of batteries.

However, what this paper fails to consider is the condition or the state of the text on an image of the battery by the time it reaches the recycling facility. The images could be worn out or faded which could hamper the ability of the computer vision algorithm to make an accurate prediction.

In such a case, being able to classify a battery by other features such as its overall shape or design could prove to be of better practical use in the domain of battery recycling.

In [8], the paper discusses the use of artificial intelligence (AI) and machine vision (MV) to improve the detection and recovery of batteries in e-waste. The authors propose a system called RoboCRM, an automated system for battery detection. The system aims to use computer vision systems and pattern recognition with an artificial intelligence engine to achieve its goal. The system's main use will be the identification and sorting of E-waste (Electronic Waste) containing batteries from the primary waste stream.

Another paper that proposes a similar system for the collection of E-waste is [9]. In this paper, the implementation of a mobile robot that identifies common electronic wastes based on transfer learning and serves as an attachment to existing municipality garbage trucks in India. The robot is expected to be able to autonomously identify and sort electronic waste based on a convolutional neural network-based (CNN) system. The CNN system has been able to achieve a 96% accuracy in the classification of E-waste within the paper. The paper also concludes by indicating that the prime benefit of such a system would be to relieve unskilled labour while providing a 20% decrease in costs over a 5-year period. It also notes the use of minimal human intervention in the collection of E-waste collection. This paper draws many parallels with the topic being discussed in this report as this report also aims to use transfer learning to sort batteries to reduce the manual labour required in the sorting of batteries.

3.2. Benefits of Transfer Learning

In [10], the paper discusses the use of transfer learning with deep convolutional neural networks for the classification of cellular morphological changes. Within their study, the authors applied the following networks: ResNet50, InceptionV3 and InceptionResnetV2. Using pretrained networks allowed to have much quicker model training. They were also able to obtain a higher predictive accuracy between 95% and 97%. This paper shows the two key benefits of transfer learning which is the quicker model training time as well as a higher predictive accuracy.

In [11], transfer learning is used to help make predictions of new diseases with little available data using the knowledge of pre-existing diseases. In the author's findings, they found that transfer learning offers the ability to approximate or even outperform comparator models trained from scratch on larger datasets of the targeted disease although the source disease has to be chosen carefully for the best results. This paper shows the large advantage of transfer learning even within a data-scarce environment, such as the sudden appearance of a new disease.

4. Methodology

4.1. Data Collection

In this section, the data collection methods used to select images for training and testing the image classification model will be discussed. The data was gathered from online sources, specifically Google and Bing, using a combination of manual selection and query-based search terms. A Python image downloader API: bing-image-downloader [12] was also used to retrieve images from Bing. The brands selected for data collection were Duracell, Energizer, Eveready, Exell, GP, Ikea, Klarus, Panasonic.

4.1.1. Manual selection of data

For Google and Bing, search terms were entered based on dry cell batteries form multiple popular brands. Multiple iterations of search queries were conducted to capture as wide a range of images as possible. The selected images were then downloaded to a local storage and included in the dataset if they were deemed to add value to the datasets. Certain conditions were used to deem if an image would add value to a dataset:

1. The entirety of the battery cell can be seen and is not obscured by any objects.
2. If the image contained a dry cell product in a brand that is not represented enough within the dataset. (e.g., Energizer Lithium-Ion, Energizer Max)
3. If the image contained a different view of the battery that could not be attained through image transformation methods. (Views attained from the 3-dimensional rotation of the dry-cell battery)
4. If the image contained a dry cell battery that was well isolated and contrasted against its background.

The above conditions were followed as much as possible as the ideal conditions for selected images. However, quantity was still an important factor to take into consideration when selecting images for training, therefore, even if the images could not meet all the above conditions, some would still be selected if they were deemed to be usable. An example of an ideal set of collected images would be for a brand can be seen in Figure 1.

Figure 1: Example set of Images of Duracell Alkaline Battery to show Ideal Images for Data Collection

4.1.2. Web sourcing of data

Bing was utilised for the web sourcing of data as Bing's API could still be easily accessed by the Python library bing-image-downloader unlike google-image-downloader where the API is no longer working for the library as of the writing of this paper.

The library was utilized to automate the retrieval of images from Bing using the API, search terms were queried to obtain images corresponding to the desired search terms entered such as "Brand X dry cell." 70 images were set to be downloaded for each category of batteries within the dataset as any number above 70 started to include many images that were unrelated to the search query. The library allowed for the efficient collection of a large number of images based on the search terms specified. These images were then downloaded and saved to local storage. Before incorporating the downloaded images within the dataset, the images were pruned to remove any irrelevant images or images that completely failed to meet the conditions specified in the manual selection of images.

4.1.3. Collected Data Summary

By combining both manual selection and automated image retrieval through the bing-image-downloader library, a comprehensive dataset was compiled for training and testing the image classification model. The inclusion of images from both Google and Bing ensured a broader spectrum of data to train the model.

Overall, a total of 854 images were collected over eight categories (brands). The image classes are slightly imbalanced as there was fewer quality data to collect from certain brands as compared to others. A summary of the collected data can be seen in Table 1.

Brand	Number of images
Duracell	139
Energizer	139
Eveready	95
Exell	86
GP	113
Ikea	84
Klarus	107
Panasonic	91

Table 1: Number of Images Collected by Brand

4.2. System Architecture

The overall system architecture for the image classification training as well as the edge-device implementation will be introduced within this section. The image classification training consists of five stages for the training of the image classification model: Data augmentation, transfer learning, training of the top classification layers and the fine-tuning of the model. The edge-device implementation architecture will consist of the input to the edge device and the prediction done by the model.

4.2.1. Image Classification Model

4.2.1.1 Data Augmentation

Due to the lack of images available for training, a data augmentation strategy was implemented using the ImageDataGenerator class from the tensorflow.keras.preprocessing module. The class provides a number of methods for the application of random transformations to images. Data augmentation can be useful for a lack of data through the means of oversampling. The ImageDataGenerator class will create augmented versions of the images in the dataset according to the specified parameters. These augmentations may come in the form of rotating, zooming, shifting, flipping, and vertically flipping the original images, thus artificially increasing the size of the dataset. Within the training of the image classification model, the dataset was split into a training and test set in a ratio of 90:10 while the training set was further split into a training and validation set in a ratio of 80:20. The following transformations were also applied:

Random Rotation: The images were randomly rotated by a maximum value of 10 degrees. This prevents the model from overfitting to a particular orientation or angle of the provided image.

Random Zoom: The images were randomly zoomed in or out by a maximum value of 20%. This prevents the model from overfitting to a particular scale of the image.
Random Height and Width Shift: The images were randomly shifted by a maximum fraction of 10% of the image's height or width. This prevents the model from overfitting to a particular state or position of the subject within the image.

Random Horizontal and Vertical Flip: The images were randomly flipped horizontally and vertically. This helps to account for the fact that objects in the real world can appear in either orientation. Vertical flip may be less commonly used that horizontal flip, however, within the context of this research, it is useful as the batteries will most often be classified in a top-down view in multiple orientations where the vertical orientation of the image does not affect the practicality of the image.

4.2.1.2. Transfer Learning Algorithm Selection

A transfer learning approach was selected due to its high effectiveness as a technique in deep learning while working with a limited amount of data. In the context of this capstone where there is a low number of images in each category, transfer learning offers significant advantages over training a model from scratch as shown in the literature review.

The considerations for the transfer learning model were that it had to be accurate, fast and of a small model size according to the design specifications. The following transfer learning models chosen to be further tested as a base for this image classification model: MobileNet, MobileNetV2, VGG16, VGG19, ResNet50, ResNet50V2. [13] The specifications of the narrowed down list of models have been inserted into Table 2 below.

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)
MobileNet	16	70.4%	89.5%	4.3M	55	22.6
MobileNetV2	14	71.3%	90.1%	3.5M	105	25.9
VGG16	528	71.3%	90.1%	138.4M	16	69.5
VGG19	549	71.3%	90.0%	143.7M	19	84.8
ResNet50	98	74.9 %	92.1%	25.6M	107	58.2
ResNet50V2	98	76%	93%	25.6M	103	45.6

Table 2: List of Specification of Narrowed Down Models

Before modification of the architecture and retraining was done, the base models' capabilities were tested against the data set to evaluate their base level performance.

Model	Weighted Average f-1 Score
MobileNet	0.18
MobileNetV2	0.13
VGG16	0.06
VGG19	0.07
ResNet50	0.02
ResNet50V2	0.15

Table 3: Base Transfer Model Comparison before Transfer Learning

The following models were then trained and tested against each other using similar hyperparameters and the collected dataset to select the model with the best results. VGG16 and VGG19 were narrowed down as a means to test if there would be any difference in accuracy with a much larger model size in comparison with the other four much smaller chosen models. A simple sequential model was also tested as a means to investigate how much transfer learning would improve the accuracy of the final image classification model. The tests were conducted on a Windows 11 PC. Tests were executed with the Anaconda Python interpreter of version 3.10.9 and the models were trained with an RTX 2060. Each training was run five times with the same hyperparameters and settings. The dataset was divided into a training and testing dataset in the ratio of 90:10. Each model's performance was then evaluated according to the mean of the weighted average f1-score across the five runs rounded to two decimal places. A larger value with the maximum of 1.0 is considered as an indicator of better performance.

Model	Weighted Average f-1 Score
Simple Sequential Model	0.02
MobileNet	0.83
MobileNetV2	0.78
VGG16	0.72
VGG19	0.68
ResNet50	0.61
ResNet50V2	0.85

Table 4: Comparison of test results across various models

From the gathered results in Table 3, the MobileNet and ResNet50V2 models showed the highest initial accuracy among the other models, differing by an f1-score of 0.03 between the models. There is also a notable difference in accuracy between the simple sequential model and the transfer learning models with the worst performing transfer learning model ResNet50 still beating out the Simple Sequential Model by a mean weighted average f1-score difference of +0.59. With these results, the MobileNet and ResNet50V2 models were chosen to be trained and fine-tuned further in an attempt to improve the results by tweaking the hyperparameters used to train the model. The MobileNet model was retained and considered as an alternative approach due to its small difference in accuracy compared to the ResNet50V2 model despite a much smaller model size and shorter inference step.

4.2.1.3. Model Architecture

In this section, the model architecture will be shown and discussed based on the model architecture shown in the figure below. The architecture consists of several layers which is shown in Figure 2 and will be further explained into this section. The Keras framework was used as a basis for the model architecture.

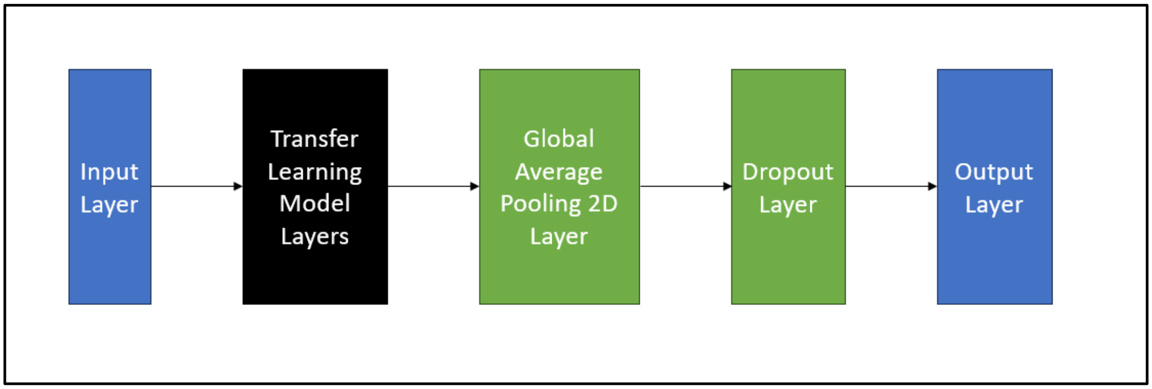


Figure 2: Simplified View of Transfer Learning Model Architecture

Input Layer: This layer contains the input layer function provided by the Keras library which is used to define the input shape of the model. In this input layer, the shape defined was 'shape = (224, 224, 3)' which was used to take in and transform an image into a height and width of 224 pixels each in an RGB format.

Transfer Learning Model Layers: This layer contains the layers of the pre-trained transfer learning model chosen from the previous section. All lower layers of the pre-trained model, which are responsible for feature extraction are frozen which means that their weights will not be updated during training. In this case, all lower layers of the pre-trained model are frozen apart from the top two layers which are responsible for classification.

Global Average Pooling 2D Layer: After the transfer learning model layers, a global average pooling 2D layer was added to perform spatial average pooling over the entire feature map, reducing each feature map to a single value. This layer helps to condense and compact the spatial information.

Dropout Layer: A dropout layer was introduced to increase generalisation of the model and helps to prevent overfitting. A dropout layer works by randomly turning off some of the inputs, setting them to zero. By using dropout, randomness is introduced to the model, ensuring the neurons do not rely too much on specific features. In this case, a dropout value of 0.5 was selected. This encourages the model to learn more flexible and general features that can work well on different examples.

Output Layer: The output layer is responsible for the final classification or prediction of the model. In this case, a dense layer was used. A softmax activation function was also used due to the multi-classification use case of this model.

4.2.1.4. Model fine-tuning

After the initial transfer learning stage, a model fine-tuning stage is added as an optional step before the model is saved. In this fine-tuning stage, the model is retrained again with a much lower learning rate ten times smaller than the original value while keeping all other parameters the same. [14] shows that an approach of lowering the learning rate helps a network to train better, especially in the case of fine-tuning.

Having a lower learning rate helps to introduce smaller adjustments to the existing weights, helping it to adjust slowly adapt to the dataset. This could also help prevent overfitting as smaller weight updates provide less opportunity for the model to memorise the examples shown to it during training. Lastly, fine-tuning the model helps it to converge better as a higher learning rate during the initial training stage could have caused the model to miss its optimal convergence point.

4.2.2. Edge-device implementation

A Raspberry Pi 4 running Raspbian OS was used as the edge device in this paper. A USB webcam was attached to it to capture visual data for model inference. OpenCV was used to take the video stream as an input before the frames from the video were processed and inferenced by the loaded model before displaying a predicted output on the OpenCV stream. The RaspberryPi was accessed and controlled through both PuTTY SSH and RealVNC® Viewer on a Windows 11 laptop using an Ethernet cable to establish a network connection between the laptop and the Raspberry Pi.

Figure 3 shows the setup and components used for the edge-device implementation.

Figure 3: Components Used for Deployment of Model on Raspberry Pi

4.2.2.1. Edge-device Model Inference

An OpenCV script was developed to integrate a trained Keras model for image classification on a Raspberry Pi platform. One model trained using ResNet50V2 and MobileNet was each chosen for the inference test based on their weighted average f1-scores. The focus of the edge-device model inference was not to verify the accuracy of the model but was instead to test if the real-time deployment of an image classification model was feasible. Therefore, the time taken for each inference step as well as the estimated personal visual confirmation of how close to real-time the video stream is was measured. The steps taken to obtain predictions from the model are as follows:

1. The OpenCV Python script was executed on the Raspberry Pi, initializing the web camera for capturing real-time video.

2. A dry cell battery was placed within the camera frame in a top-down view.

3. The captured video frames go through image processing to fit the input required by the model to make a prediction.

4. The processed video frames are input into the loaded model for classification. The model makes a prediction of the battery brand present within the video feed.

5. The classification predictions generated by the model is displayed on the OpenCV stream window in real-time.

6. Record down the time taken for the last ten inferences during any time period.

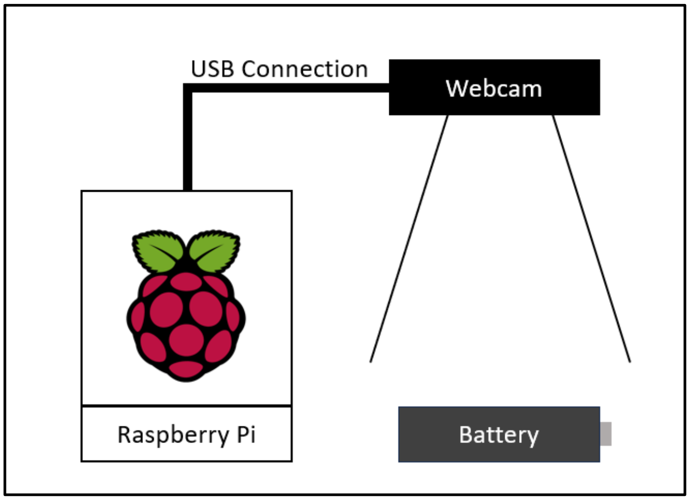


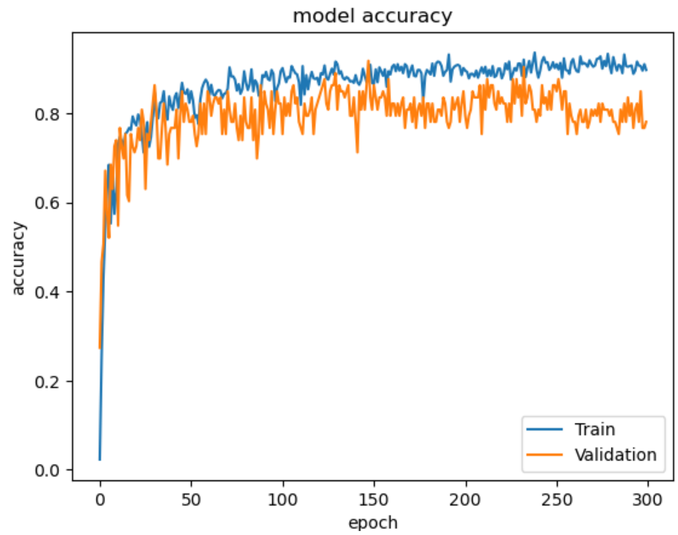
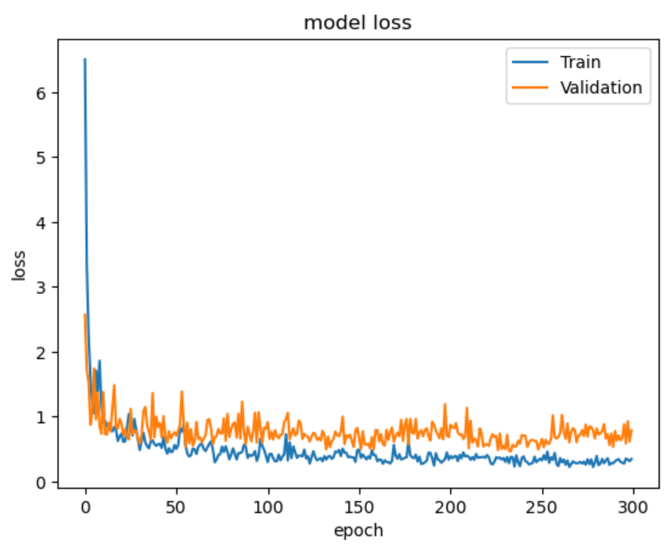
Figure 4: Representation of Inference Setup from Side-view

5. Results and Analysis

5.1. Results from Model Training

5.1.1. Accuracy and Loss Graphs

The model accuracy and loss graphs were plotted after the initial training as well as after the fine-tuning stage. The following are the accuracy and loss graphs for the ResNet50V2 Model and the MobileNet model, respectively.

ResNet50V2	
Accuracy Graph	
Loss Graph	

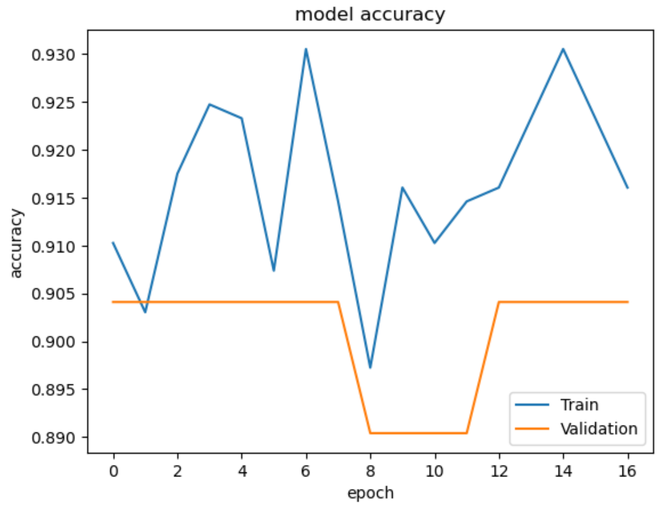
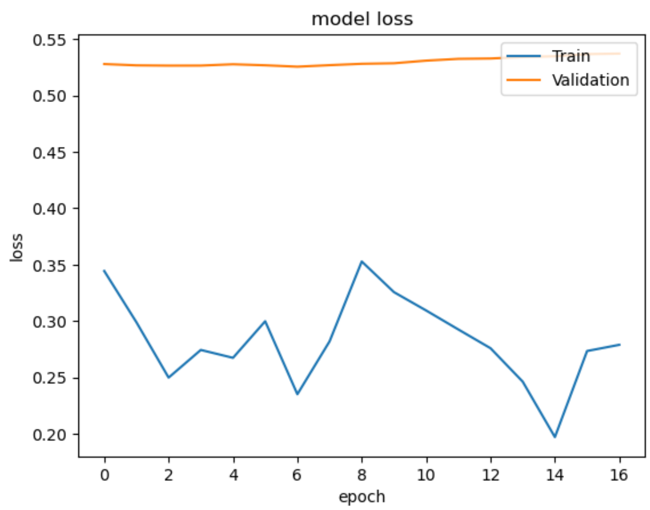
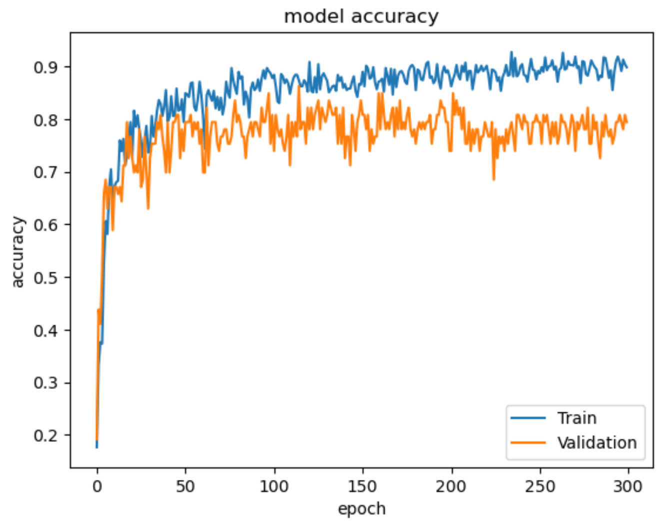
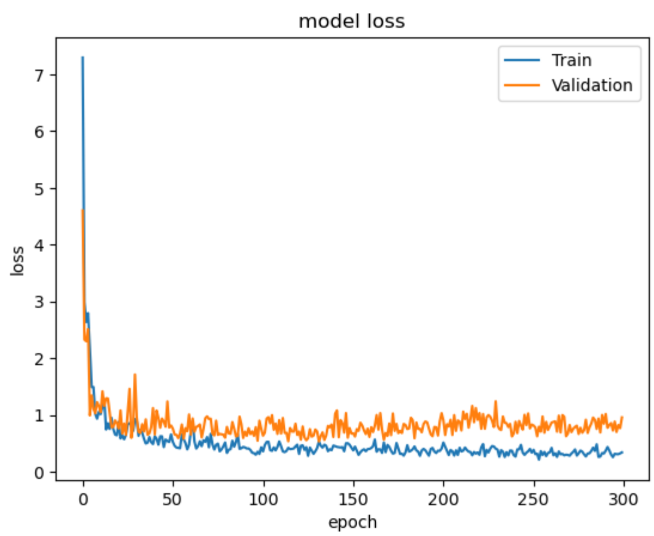
Fine-Tuned Accuracy Graph	
Fine-Tuned Loss Graph	

Table 5: Accuracy and Loss Graphs of ResNet50V2 Transfer Learning Model

<table>
<tr><th colspan="2">MobileNet</th></tr>
<tr><td>Accuracy Graph</td><td>
</td></tr>
<tr><td>Loss Graph</td><td>
</td></tr>
</table>

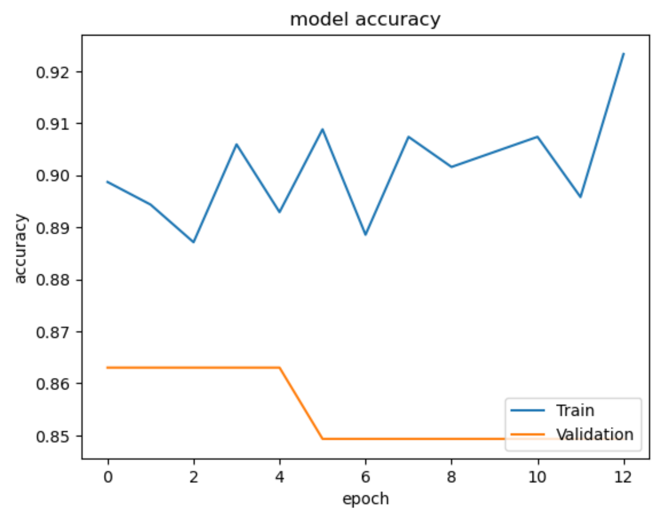
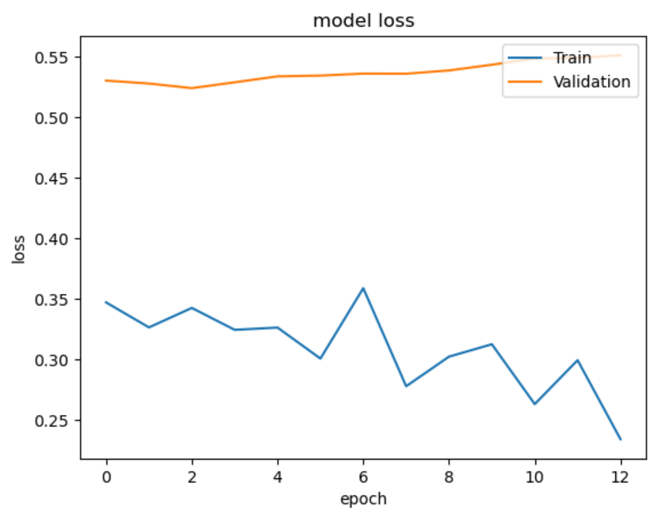
Fine-Tuned Accuracy Graph	
Fine-Tuned Loss Graph	

Table 6: Accuracy and Loss Graphs of MobileNet Transfer Learning Model

In Table 4 and 5, the accuracy graphs from both model's trainings show an increasing trend in accuracy on both the training and validation sets over the epochs until reaching a point where the accuracies fluctuate between the 0.7-0.9 range. This indicates that the model is learning and improving its predictions over time.

The loss graphs show a decreasing trend over the epochs on both the training and validation sets for both models. The loss decreases until it is below 2 where it begins to fluctuate between the values of 0 and 2. A decreasing loss signifies that the model is reducing its errors and improving its performance.

5.1.2. Classification Report

The classification reports of both models were generated after the model fine-tuning stage. The following are the classification reports for the ResNet50V2 Model and the MobileNet model, respectively.

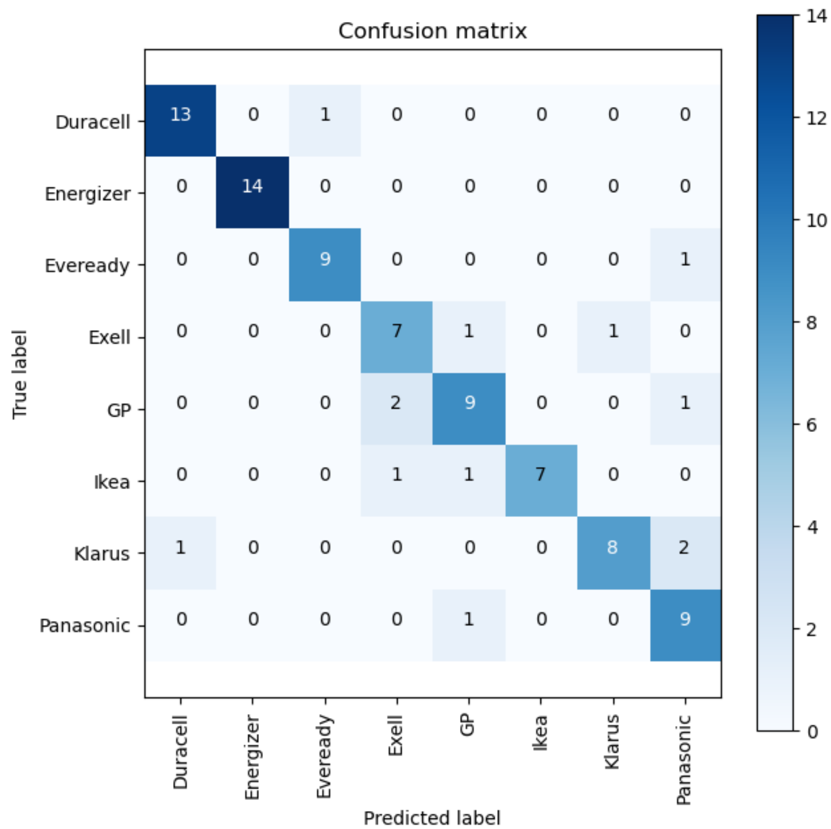
Classification Report					
ResNet50V2		precision	recall	f1-score	support
	Duracell	0.93	0.93	0.93	14
	Energizer	1.00	1.00	1.00	14
	Eveready	0.90	0.90	0.90	10
	Exell	0.70	0.78	0.74	9
	GP	0.75	0.75	0.75	12
	Ikea	1.00	0.78	0.88	9
	Klarus	0.89	0.73	0.80	11
	Panasonic	0.69	0.90	0.78	10
	accuracy			0.85	89
	macro avg	0.86	0.85	0.85	89
	weighted avg	0.87	0.85	0.86	89
MobileNet		precision	recall	f1-score	support
	Duracell	0.92	0.86	0.89	14
	Energizer	0.85	0.79	0.81	14
	Eveready	0.78	0.70	0.74	10
	Exell	0.80	0.89	0.84	9
	GP	0.92	0.92	0.92	12
	Ikea	0.90	1.00	0.95	9
	Klarus	0.77	0.91	0.83	11
	Panasonic	0.78	0.70	0.74	10
	accuracy			0.84	89
	macro avg	0.84	0.84	0.84	89
	weighted avg	0.84	0.84	0.84	89

Table 7: Classification Reports of ResNet50V2 and MobileNet Transfer Learning Models

In Table 6, both transfer learning models show relatively high f1-scores for most brands. However, the models are both weaker when predicting batteries from Panasonic. The ResNet50V2 model shows more difficulty in predicting Exell, GP, and Panasonic batteries, being the only brands having an f1-score below 0.8. On the other hand, the MobileNet model shows more difficulty predicting Eveready and Panasonic batteries, with Panasonic being a common weak point between the two models. The MobileNet model also shows a better median accuracy as compared to the ResNet50V2 model. Overall, both models show relatively good performance in accuracy, with both being extremely close to the targeted weighted average f1-score of 0.85 with an accuracy difference of ±0.01.

5.1.3. Confusion Matrix

The confusion matrices of both models were generated after the model fine-tuning stage. The following are the confusion matrices for the ResNet50V2 model and the MobileNet model, respectively.

<table>
<tr><th colspan="2">Confusion Matrix</th></tr>
<tr><td>ResNet50V2</td><td>
</td></tr>
</table>

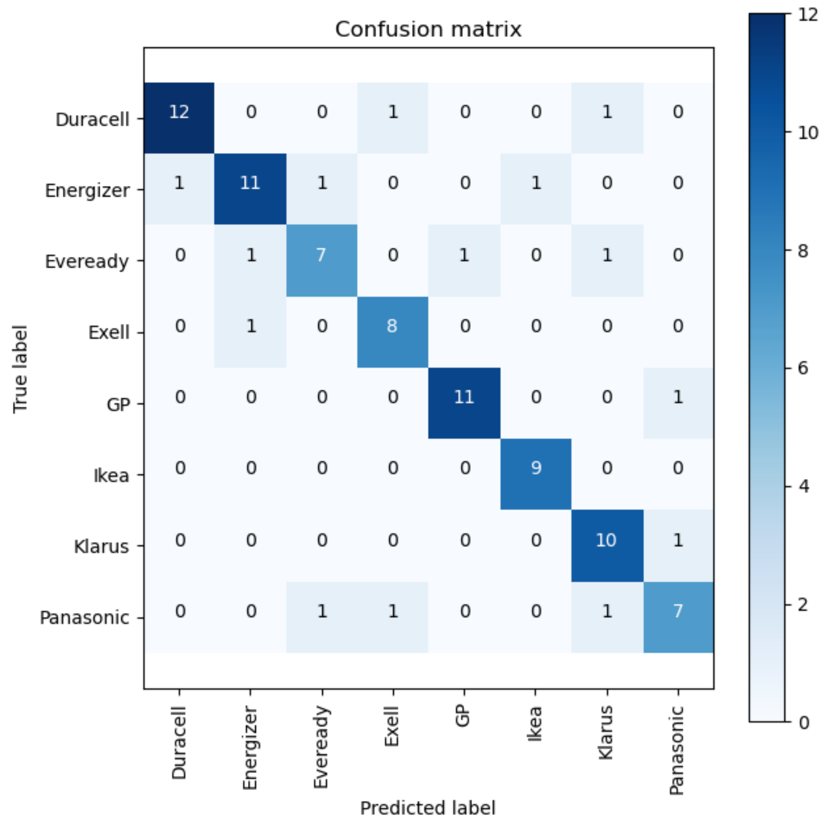
MobileNet	

Table 8: Confusion Matrices of ResNet50V2 and MobileNet Transfer Learning Models

Following the classification report in Table 7, both transfer learning models show relatively good predictive ability for most brands. However, the models are both weaker when predicting batteries from Panasonic. The ResNet50V2 model shows more difficulty in predicting Exell, GP, and Panasonic batteries, having less true label predictions for those brands. On the other hand, the MobileNet model shows more difficulty predicting Eveready and Panasonic batteries, with Panasonic being a common weak point between the two models. Overall, both models show relatively good accuracy in predicting the true label of the battery brands from the confusion matrix.

5.2. Inference Test Results

The model size and time taken for each inference step was both recorded from the edge-device model inference test. Ten consecutive inference steps were recorded while a battery was in frame and averaged out to obtain the average time taken for each inference step. The stream was also personally visually analysed to determine how close the stream was compared to real-time. The stream contains the prediction for the battery brand and the confidence of the prediction as a label in the top left-hand corner.

Model	Model Size (MB)	Inference Steps Recorded	Average Inference Step (ms)
ResNet50V2	90.4	1/1 [==============================] - 1s 554ms/step 1/1 [==============================] - 1s 543ms/step 1/1 [==============================] - 1s 536ms/step 1/1 [==============================] - 1s 540ms/step 1/1 [==============================] - 1s 543ms/step 1/1 [==============================] - 1s 540ms/step 1/1 [==============================] - 1s 534ms/step 1/1 [==============================] - 1s 530ms/step 1/1 [==============================] - 1s 538ms/step 1/1 [==============================] - 1s 545ms/step	540.3
MobileNet	12.6	1/1 [==============================] - 0s 188ms/step 1/1 [==============================] - 0s 187ms/step 1/1 [==============================] - 0s 183ms/step 1/1 [==============================] - 0s 183ms/step 1/1 [==============================] - 0s 183ms/step 1/1 [==============================] - 0s 185ms/step 1/1 [==============================] - 0s 188ms/step 1/1 [==============================] - 0s 185ms/step 1/1 [==============================] - 0s 180ms/step 1/1 [==============================] - 0s 179ms/step	184.1

Table 9: Model Size and Average Inference Step of ResNet50V2 and MobileNet models

Figure 5: OpenCV Stream of Real-Time Battery Image Classification Script

From Table 8, we can see that MobileNet has a smaller model size and a shorter average inference step, being 86% smaller in size and having an inference step that is 66% shorter. From the visual inspection of the stream, it was also determined that the video stream was much smoother and closer to real-time when the MobileNet model was loaded as compared to the ResNet50V2 model. Both models also performed well in a quick test done with dry cell batteries on hand (Energizer Max, Ikea, and GP Ultra) when predicting the Energizer and Ikea batteries but had difficulty predicting the GP battery correctly.

5.3. Analysis

5.3.1. Analysis of Results from Model Training

5.3.1.1. Strong Points

Both models show strong predictive across the board, being able to correctly predict and label most battery label brands based on the test set. They show a weighted average f1-score close to the desired target of 0.85 differing by ±0.01. A very strong predictive ability is shown especially for battery label brands where many ideal images were present in the dataset according to section 4.1 Data Collection. This is strongly proven by the Duracell and Energizer predictions in the classification reports and confusion matrices.

5.3.1.2. Weak Points

Both models show a much lower predictive ability when predicting batteries from Panasonic. This might be due to the large amounts of variety of Panasonic batteries within the dataset.

5.3.2. Analysis of Results from Inference Test

5.3.1.1. Strong Points

From the inference test, the MobileNet showed much better results as compared to the ResNe50V2 model when it came to the practicality of the models in a real-time use case on an edge-device. This is due to the model's smaller size and quicker inference time which reduces the delay in between each frame being fetched for the OpenCV stream. From the quick tests done with a few dry cell batteries on hand (Energizer Max, Ikea and GP Ultra), both models also performed well on differentiating and predicting the Energizer Max and Ikea batteries which could be due to the higher support and larger amounts of practical images collected within the Energizer dataset whilst the Ikea batteries are easily identifiable due to having a unique design amongst the batteries in the data collected.

5.3.1.2. Weak Points

The OpenCV stream still stuttered when visually inspected when both models are loaded and inferring. However, it is to be noted that the stream from the MobileNet model still showed a very practical, usable video stream. This is due to the inference time taken in between the fetching of each frame from the video stream. Both models also showed difficulty in predicting the brand label of the GP Ultra during the inference. Despite the MobileNet model showing good predictive accuracy in the classification report, it showed difficulty in predicting the GP battery as well. This could be due to the lack of practical usable data within the GP dataset that did not translate well to a real-world use scenario despite performing well in the test evaluation.

6. Project Management

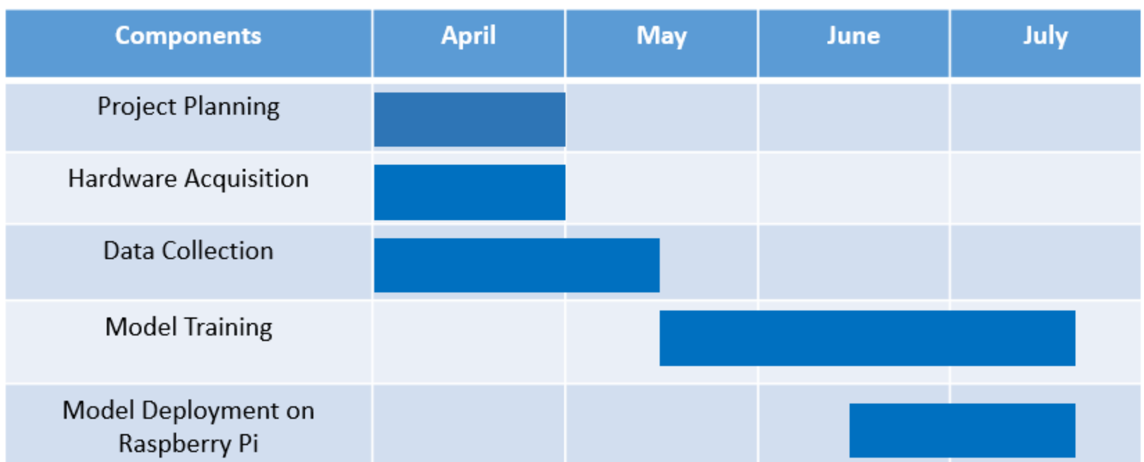


Figure 6: Gantt Chart of Project Plan

7. Conclusion

The results are promising, with an achieved weighted average f1-score of 0.86 for the classification of eight different battery brands trained with a total of 855 images using the ResNet50V2 model as a base for transfer learning.

A weighted average f1-score of 0.86 was also achieved using the ResNet50V2 model as a base whilst a weighted average f1-score of 0.84 was also achieved using the MobileNet model as a base.

An inference test of both models using an edge device was also performed using a Raspberry Pi 4 Model B (8GB RAM) and OpenCV. There were good results from the inference test of both models for brands that had great practical images within their datasets (e.g., Energizer, Ikea) whereas the inference model had difficulty making correct predictions when presenting it other brands that did not have as high a quality dataset.

At the end, the MobileNet model was chosen as the better model on the edge-device due to its much lower inference time and smaller model size compared to the ResNet50V2 model whilst still being able to achieve a weighted average f1-score of 0.84 which was extremely close to the targeted weighted average f1-score of 0.85. This demonstrates the potential use of image classification models in the process of battery sorting automation within recycling plants.

The researched solution also provides advantages over similar works that require text recognition. The text labels on batteries may not always be in good condition, which is a large oversight in other solutions, especially considering that the batteries have been discarded and are being prepared to be recycled instead of being in mint condition.

However, the same can also be applied to this research where the mentioned weakness is not as apparent. The batteries may not stay in good condition when they reach the recycling plant, however, the overall colours or design of the battery may still be effective enough to get a good prediction. The dataset is greatly lacking in images of batteries in less-than-ideal conditions as well as ideal practical battery images for most brands in general.

This work can be expanded much further with much more data; being able to expand the dataset to contain batteries in a range of conditions or including much more types, sizes or brands or varieties of batteries within a brand. With more data, the datasets do not have to be converged and can simply have a classification label for each type or variety of battery (Energizer Max, Duracell Procell etc.).

The work could also be expanded to a further, more practical stage focusing on deployment since the effectiveness of transfer learning has already been proven in this report. A transfer learning approach on a pre-trained Object Detection model as a base, especially if a large effective labelled image dataset was available. A good example would be the MobileNetSSDv2 model. Having an object detection model would allow recycling plants to detect and sort multiple batteries within the same frame which is much more practical.

8. References

[1] European Commission. (2020, July 23). Actual individual consumption (AIC) per capita in PPS, 2019. Eurostat News. Retrieved July 7, 2023, from https://ec.europa.eu/eurostat/web/products-eurostat-news/-/ddn-20200723-1

[2] Sorting batteries? This is how we do it. (n.d.). Bebat. Retrieved July 10, 2023, from https://www.bebat.be/en/blog/how-sorting-batteries

[3] Tomaszewski, K. (2022, June 15). Battery sorting. Cirbasolutions.com; Cirba Solutions. https://www.cirbasolutions.com/advancedsorting

[4] Costa, C. M., Barbosa, J. C., Gonçalves, R., Castro, H., Del Campo, F. J., & Lanceros-Méndez, S. (2021). Recycling and environmental issues of lithium-ion batteries: Advances, challenges and opportunities. Energy Storage Materials, 37, 433-465.

[5] Yu, X., Li, W., Gupta, V., Gao, H., Tran, D., Sarwar, S., & Chen, Z. (2022). Current challenges in efficient lithium-ion batteries' recycling: A perspective. Global Challenges (Hoboken, NJ), 6(12), 2200099. https://doi.org/10.1002/gch2.202200099

[6] (Anik), A. K. (1674386361000). Economics challenges of Li-Ion Battery Recycling from End-of-Life in India. Linkedin.com. Retrieved July 7, 2023, from https://www.linkedin.com/pulse/economics-challenges-li-ion-battery-recycling-from-kumar-anik-/

[7] Sterkens, W., Diaz-Romero, D. J., Xiao, B., Wu, B., Goedemé, T., Dewulf, W., & Peeters, J. (2022). Computer vision and optical character recognition for the classification of batteries from WEEE. Procedia CIRP, 105, 110–115. https://doi.org/10.1016/j.procir.2022.02.019

[8] Johnson, M., Khatoon, A., & Fitzpatrick, C. (2022). Application of AI and Machine Vision to improve battery detection and recovery in E-Waste Management. 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 1–6.

[9] Shreyas Madhav, A. V., Rajaraman, R., Harini, S., & Kiliroor, C. C. (2022). Application of artificial intelligence to enhance collection of E-waste: A potential solution for household WEEE collection and segregation in India. Waste Management & Research: The Journal of the International Solid Wastes and Public Cleansing Association, ISWA, 40(7), 1047–1053. https://doi.org/10.1177/0734242X211052846

[10] Kensert, A., Harrison, P. J., & Spjuth, O. (2019). Transfer learning with deep convolutional neural networks for classifying cellular morphological changes. SLAS Discovery, 24(4), 466–475. https://doi.org/10.1177/2472555218818756

[11] Roster, K., Connaughton, C., & Rodrigues, F. A. (2022). Forecasting new diseases in low-data settings using transfer learning. Chaos, Solitons, and Fractals, 161(112306), 112306. https://doi.org/10.1016/j.chaos.2022.112306

[12] Bing-image-downloader. (n.d.). PyPI. Retrieved July 13, 2023, from https://pypi.org/project/bing-image-downloader/

[13] Keras applications. (n.d.). Keras.Io. Retrieved July 13, 2023, from https://keras.io/api/applications/

[14] Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. ICT Express, 6(4), 312–315. https://doi.org/10.1016/j.icte.2020.04.010

9. Knowledge and Training Requirements

The section list all the knowledge, skillsets, and certifications both from the degree programme and beyond that was necessary for the successful completion of the capstone projects.

9.1. Applicable Knowledge from the Degree Programme

The prerequisite knowledge and skillsets from the degree programme that was necessary for the capstone projects are as follows:

No.	Module(s)	Knowledge(s) Applied
1	CSC1004	Knowledge from this module was required for the basic setup of the Raspberry Pi.
2	CSC3009	Knowledge from this module was required to understand how machine learning works and how to effectively train a machine learning model, however, the knowledge from the module was taught much too late to be of any use when it was actually required in the project.

9.2. Additional Knowledge, Skillsets, or Certifications Required

The following are the additional requirements and knowledge that were required for the capstone projects:

No.	Additional Requirement(s)	Knowledge(s) Applied
1	Knowledge of Keras and Tensorflow	Knowledge of Keras and Tensorflow was required to create and train the transfer learning models.
2	Knowledge of OpenCV	Knowledge of OpenCV was required to setup a video stream to capture frames to be fed into the trained model for inferencing.

3	Knowledge of Transfer Learning Models	Knowledge of Transfer Learning models was required to select and re-train a pre-trained network to classify batteries by their brands.
4	Knowledge of Data Collection Techniques	Knowledge of data collection techniques from the web through manual or scripted means was required to collect the necessary data for this project.

END OF REPORT