OXFORD JOURNAL
OF STUDENT
SCHOLARSHIP

Analysing which Machine Learning Algorithms Segregate Industrial Plastic Waste Most Efficently—a Comparative Study of SVM, Logistic Regression and Decision Tree Classifiers on ResNet-18 Embeddings.

Aarav Jain
aarav.jain@hotmail.com

ABSTRACT

The accumulation of plastic waste worldwide poses critical environmental and economic challenges. An Accuracy-Speed Gap in industrial material recovery facilities exacerbates them. Manual sorting is hazardous and unsustainable, as traditional mechanical sensors often misclassify visually identical polymers. This study evaluates a custom-made hybrid deep learning framework that compares the classification accuracy of three machine learning algorithms applied to deep visual embeddings for the segregation of industrial plastic waste. The methodology differentiates visual feature extraction from mathematical classification. This is by utilising a ResNet18 Convolutional Neural Network to process RGB image data into 512-dimensional continuous numerical arrays. These high-dimensional embeddings were subsequently fed into 3 optimised classical machine learning algorithms: SVMs, LR and DTC. The models used were trained using a dataset of 7 distinct chemical polymers. The empirical results revealed that the SVM outperformed both LR and DTC. Thereby, achieving an aggregate accuracy of 82.9% and high precision in identifying LDPE. However, the Logistic Regression and DTC models scored much lower than the SVM models. Thus, it reflects the limitations of linear and orthogonal logic in separating dense spatial arrays. This research has demonstrated that replacing computationally intensive deep classification layers with regularised SVM hyperplanes can advance the industrial viability of chemical segregation.

KEYWORDS:
Neural Networks; Convolutional Neural Network; Plastic Waste Segregation; Logistic Regression; Decision Tree; Support Vector Machines; Material Recovery Facilities; Circular Economy; Accuracy-Speed Gap; Hybrid Deep Learning Framework; ResNet-18; Feature Extraction; Artificial Intelligence; Machine Learning; Deep Learning; Supervised Machine Learning; Thermoplastic Polymers; Polyethylene Terephthalate; High-Density Polyethylene; Polyvinyl Chloride; Waste Management; Automated Sorting Systems; Computer Vision; Radial Basis Function Kernel; Image Classification; F-1 Score; Algorithm Optimisation; High-Dimensional Embeddings; Polymer Resins; Confusion Matrix

June 2026
Vol 8. No 2.

1. INTRODUCTION

1.1 The Problem Associated with Plastic Waste Not Being Segregated

The management and recovery of plastic waste is the most challenging problem that humanity faces. The lack of technical skills for managing waste, limited recycling infrastructure and insufficient awareness of rules are major factors driving the massive accumulation of plastic waste (Kibria et al., 2023). Plastic pollution has adverse environmental effects and poses social and economic challenges across countries. The social obstacles include loss of biodiversity due to plastic ingestion or entanglement, ecosystem degradation from seeping polyester toxins, and the presence of microplastics in food and water (IUCN, 2021). The irresponsible management of plastic waste is regarded as the primary cause. Economically, losses from marine plastic pollution are estimated at up to $2.5 trillion annually, resulting in a 1-5% loss of marine ecosystem services (NCEL, 2025).

1.2 The Concept of Plastic

Approximately 460 million metric tonnes of plastic are produced annually, and around 20 million metric tonnes are discarded into various ecosystems. This includes cities, landfills and oceans. The disposal and management of plastic have created a global waste challenge. This is because plastic waste does not degrade easily, taking 20 to 500 years to decompose (Hardman, L., 2023). The enzymes responsible for biodegradation do not easily recognise or bind to plastic polymers, as the carbon-carbon bonds within the polymer molecules require a great deal of energy to break (Vodovotz, 2022). Chemical stabilisers are added to most plastic objects to prevent degradation. Therefore, plastic recycling is primarily hindered by the material's inherent complexities and the severe limitations of manual sorting. Plastics are highly engineered polymers with varying flexibility, hardness, and opacity (Alliance to End Plastic Waste, 2021), making them incredibly useful yet highly complex to manage when mixed. Despite the recycling efforts by industries, scientists, and governments, plastic waste generation is still expected to triple by 2060.

On another level, only 30% of people currently have proper access to waste disposal and plastic waste management (Kibria et al., 2023). Lack of technical skills, insufficient recycling infrastructure, and low awareness of plastic waste management are key factors driving the massive buildup of plastic waste. A key issue in manual sorting is that relying on humans is unsustainable. Waste sorting is classified as a 3D job: Dirty, Dull and Dangerous (Shetfale et al., 2025). This is because it exposes workers to biohazards, such as toxic fumes and particulate matter, released during plastic recycling, which may cause respiratory issues (Le et al., 2023). Frequent headaches, nausea and skin irritation are common struggles reported by workers when proper safety measures are not in place. In the long run, elevated levels of heavy metals (lead and cadmium) in the bloodstream from exposure to waste can pose significant threats to workers (Plastics For Change, 2025).

Figure 1.2.1: Material Recovery Facilities segregating general waste (Cambridge Companies, 2025)

1.3 Material Recovery Facilities

Industrial-scale management of plastic waste relies on Material Recovery Facilities. These are specialised facilities that separate, process, and prepare recyclable materials for the market by sorting mixed dry waste into different categories on a high-speed mechanical flow belt using technologies such as magnets, screens, and air classifiers (Kushaagra Innovations Foundation, 2024). However, this often creates a critical operational problem known as the 'Accuracy-Speed Gap'. While human labourers can sort roughly 30-40 items per minute, industrial belts move at 3-5 meters per second and require sorting decisions in milliseconds (Anis Trend, n.d.). With global waste expected to increase by 70% by 2050, this volume exceeds human processing capacity (Minu et al., 2025). As a result, traditional sorting methods are failing to close this gap in plastic segregation.

Furthermore, the fundamental difficulty in manual sorting lies in the extreme visual similarity of distinct polymers. Plastic types, such as HDPE, PET, PP, LDPE and PVC, may look identical to the human eye in terms of transparency, colour and texture. Therefore, manual segregation is highly error-prone. For instance, mechanical optical sorters, such as Near Infrared (NIR) sensors, frequently fail to distinguish between different plastic types as varying polymers exhibit overlapping spectral signatures, surface contamination, food residue or degradation. Thus, it heavily distorts the readings (Olowolayemo et al., 2022).

1.4. The Positives of Artificial Intelligence (AI)

Artificial intelligence can offer a highly viable technological solution to bridge this gap. A neural network is a machine learning model that learns weights and biases to recognise patterns and map inputs to outputs

(Islam et al., 2019). Specifically, Convolutional Neural Networks (ConvNets or CNNs), which function similarly to traditional feed-forward networks but completely dominate modern computer vision due to their high accuracy in image classification, excel at this task (Islam et al., 2019; Bobulski & Kubanek, 2021). In industrial setups, these networks operate by using sensors, such as optical cameras and automated hardware. For example, pneumatic air jets or robotic arms are used to physically separate plastic (Anchieta-Silva et al., 2026). Unlike traditional NIR sensors, which attempt to read easily obscured chemical signatures, CNNs address this challenge by leveraging deep visual feature extraction. CNNs process massive visual datasets to analyse the distinct shapes, textures and opacities of different plastics. They perform visual recognition with the same accuracy as humans, but at faster speeds. Ultimately, this automation is essential for recycling to be manageable, effective and profitable (Alimbekova et al., 2024).

1.5. Aims and Goals of the Paper

This study aims to compare the performance of different supervised machine learning algorithms by using quantitative metrics to determine the most efficient model for industrial plastic waste segregation. The goal is to identify the supervised machine learning model with optimal classification accuracy by developing a computational framework that trains these models on the same plastic waste dataset. This paper will explore specific systems in the Literature Review to understand the current solutions. It will provide details about the dataset and code generation in the Methodology section. In the Findings and Discussion section, the paper will use comparative visualisation of accuracy and loss curves to evaluate the best model for plastic waste segregation.

The fundamental importance of this research lies in demonstrating which machine learning model is highly accurate in AI for solving the plastic waste segregation problem and making the Circular Economy economically feasible. This is while simultaneously removing human workers from hazardous sorting conditions. Ultimately, this matters to MRF operators seeking profitability, policymakers aiming to divert waste from landfills, and technology manufacturers building next-generation sorters. Crucially, it provides global brands, such as Coca-Cola and Unilever, with the high-purity recycled plastics urgently required to meet their 2030 sustainability commitments.

2. LITERATURE REVIEW

2.1. Linear Waste Management Tools

Academic literature has recently emphasised the transition from traditional, linear waste management models to closed-loop circular economies through the integration of Artificial Intelligence. Rather than relying on manual observations, researchers are now exploring how machine learning algorithms can be used to evaluate the life cycles of materials and predict waste-generation patterns. The shift in the literature is crucial, as the structural complexities of modern materials require highly precise sorting mechanisms to prevent their entry into landfills or aquatic ecosystems. (Islam et al., 2025; World Economic Forum, 2026)

2.2. Types of Plastics

The chemical and physical diversity of the municipal solid waste plastic stream must be understood to understand the complexity of automated segregation. There are 6 major thermoplastic polymers globally: Polyethylene Terephthalate (PET), High Density Polyethylene (HDPE), Polypropylene (PP), Low Density Polyethylene (LDPE), Polystyrene (PS) and Polyvinyl Chloride (PVC). Due to strong demand in the secondary market, PET is used for transparent beverage bottles, while HDPE is used for opaque detergent jugs. They are among the most valuable commodities in the recycling sector. On the other hand, PP is widely used in rigid food packaging and automotive components. Whilst LDPE dominates the production of flexible films and grocery bags. PS is primarily used in the manufacturing of cutlery, food packaging and insulating foams (Chandra Asri, 2025). Finally, PVC is utilised in piping and blister packaging, which are structurally very dense and chemically different due to their high chlorine content. At the same time, certain polymer products are made from composite plastic polymers such as acrylic, polycarbonate, and nylon, which are classified as OTHERS. However, recycling plastic is not a straightforward process, as polymers often appear visually identical, such as PET and PVC, and are chemically incompatible when melted together, or can jam moving conveyor belts (Bobulski & Kubanek, 2021; Alliance to End Plastic Waste, 2021). Hence, these polymers must be separated with high purity. This is not merely for environmental reasons but as an absolute requirement in industries.

2.3. Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational architectures which are inspired by the biological workings of the human brain. They are constructed by using interconnected layers of artificial nodes. They use a mathematical method to assign and adjust weights and biases to functions. This is why when data enters the network, they are multiplied by weights that indicate the significance of each variable. Then, this passes through a non-linear activation function. Therefore, it allows it to understand highly complex, non-linear relationships in data. This framework typically consists of an input layer that receives raw data, one or more layers, and an output layer which produces a prediction or classification (Bobulski & Kubanek, 2021; Krizhevsky et al., 2012)

The network learns by an iterative process called backpropagation. The network predicts and calculates the error between its prediction and the actual label during training. Then, it propagates this error mathematically backwards through the layers. Thereby, updating the weights. The neural network iterates this process many times over 1000s of iterations, gradually minimising its error rate and learning how to identify intricate patterns. For instance, visual differences between distinct polymers (Bobulski & Kubanek, 2021).

The two studies by Bobulski & Kubanek (2021) and Sundarlingam & Ramanathan (2024) have investigated the benefits and limitations of using neural networks for segregating plastic waste. Moreover, certain Neural Networks, such as ResNet-50 and EfficientNet, have been proven to be well-suited for plastic waste segregation. ResNet-50 is a network based on residual blocks with skip connections, which allow data to bypass some intermediate layers, helping overcome the vanishing gradient problem and enabling it to extract features from deep layers without losing accuracy (He et al., 2015). The latter, EfficientNet, operates on the concept of compound scaling. Thus, systematically increasing the network's

depth, width and input image resolution to identify fine-grained visual patterns at lower computational cost (He et al., 2015). These networks have achieved accuracy rates of 94.1% to 99.1%. This makes large-scale, profitable polymer recycling possible for industries. The financial reward for this upgrade is massive, as The Ellen MacArthur Foundation has proven. In 2024, they reported that AI-enabled recycling could reduce global landfill waste by 20% and save the industry $10 billion annually by 2030 (Recycling Today, 2025). The three most commonly used supervised machine learning models are: Decision trees, Logistic regression and Support vector machines.

2.4. Decision Trees (DTC)

A Decision tree is a supervised machine learning algorithm that utilises a hierarchical flowchart-like structure to make classifications or predictions. It begins at a root node and splits data into smaller subsets. This is based on the most significant attribute. It is constructed from internal nodes representing tests on specific features, branches representing test outcomes, and leaf nodes representing the final class label assigned to each data point (James et al., 2013).

The algorithm, which is used when data is structured and highly categorical, functions by recursively splitting the dataset into small subsets. This is based on features that yield the highest 'information gain' or lowest 'gini impurity' metrics. These measures how well a specific attribute separates the data into distinct classes (James et al., 2013). In the context of waste segregation, decision trees might evaluate physical sensor data, such as weight, opacity and density, to sequentially filter out different plastic types (Astuti et al., 2023). The primary advantage of a decision tree is its extreme interpretability and low computational cost. In contrast to the black box of a neural network, a Decision Tree allows facility operators to see the exact logical path the AI took to classify a piece of plastic. This requires minimal computational power to execute on an industrial sorting line (James et al., 2013).

However, the major limitations of Decision Trees are their high instability and their inability to handle unstructured visual data. This means that a small change in the training data can cause the tree to generate a completely different set of rules. Furthermore, they are highly prone to creating overly complex trees that do not generalise well. Thus, leading to overfitting (James et al., 2013). Moreover, they are mathematically incapable of extracting spatial features from real-time camera feeds. This makes them inefficient for the continuous visual computer vision tasks required in modern robotic MRFs. This is where deep learning neural networks significantly outperform them (Lubongo et al., 2024).

2.5. Logistic Regression (LR)

Logistic Regression is a foundational statistical model which is used for binary classification. This refers to situations with two possible outcomes. Logistic Regression models the probability of a given class or event by fitting a logistic function to the data. This is also known as a sigmoid curve. This curve forms a distinct S-shape that mathematically squashes an extreme input value into a strict probability range. This is between 0 and 1, rather than outputting continuous values (view Figure 2.5.1) (AI Stack Exchange, n.d.). It functions by analysing the relationship between multiple independent variables and a binary dependent variable. Thus, establishing a mathematical threshold to categorise the data into two distinct classes: 0 and 1. The relationship is as follows between the probability (P) and z:

$$P(Y = 1) = \frac{1}{1 + e^{-z}}$$

Here, the variable z is the linear combination of the input variables and their respective coefficients. If the resulting probability score exceeds a predefined threshold (typically 0.5), the algorithm classifies the data point as the positive class. Similarly, it classifies data into the negative class if it falls below the threshold. In a waste management scenario, it would be used for highly specific, stripped-down sorting mechanisms, such as taking spectral readings from an optical sensor to answer direct yes/no questions. For example, 'Is this item transparent or opaque?'

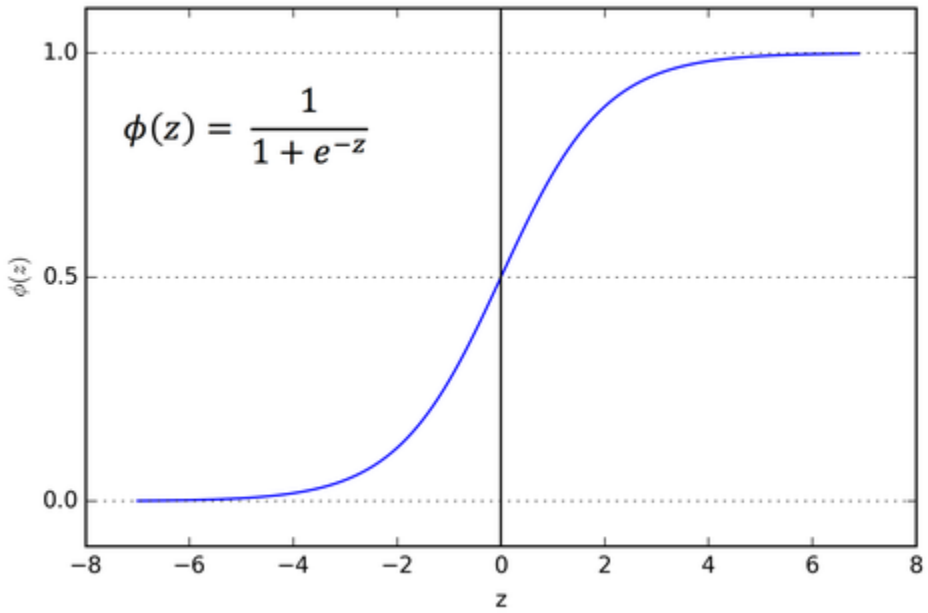

Figure 2.5.1: Sigmoid Curve Representation (AI Stack Exchange, n.d.)

The advantage of LR is that it achieves extremely high waste segregation, with high mathematical efficiency, interpretability and speed. Researchers can examine the calculated coefficients to see exactly how much each feature contributes to the final prediction. In addition, it provides probability scores natively, giving a clear measure of the model's efficiency. As it merely calculates a weighted sum of inputs and applies a sigmoid function, it can process data in microseconds with almost 0 computational overhead. Therefore, it serves as a highly interpretable baseline model to verify whether a sorting task even requires a more complex AI. (LaValley, 2008).

However, LR's critical limitation lies in its strict assumption of linearity between the features and log-odds of the outcome (LaValley, 2008). In other words, the algorithm assumes that a constant change in the input variable will always result in a proportional shift in the odds of the outcome (LaValley, 2008). Geometrically, this means it attempts to separate data into different categories using a perfectly straight,

rigid decision boundary. Therefore, it is incapable of solving non-linear problems or extracting complex problems. For instance, between 6 overlapping plastic bottles on a conveyor belt, as it cannot capture the intricate and multidimensional relationships that Neural Networks are specifically designed to solve (James et al., 2013).

2.6. Support Vector Machine (SVM)

The support vector machine is a powerful supervised machine learning algorithm that can be used for both classification and regression. The basic idea of an SVM is to identify the optimal hyperplane, a multi-dimensional boundary that best separates the data into different classes. The algorithm determines the support vectors, which are the closest data points to the boundary, and maximises the margin (distance) between the hyperplane and these vectors. This is done to ensure the most robust classification, instead of simply finding separating lines. The support vectors are the only data points that influence the position of the boundary. This is because the algorithm's mathematical objective is to define the maximum width of the gap between the classification groups. Any data points lying behind the support vectors are already classified. Thus, adding, moving or removing them will not change the margin calculation. As a result, once the boundary is set, the model ignores them (Suykens et al., 2004; Bobulski & Kubanek, 2021).

When the data is complex and high-dimensional, and a simple straight line does not easily separate the classes, SVMs are used. This uses a mathematical technique called the ‘Kernel Trick’. While this process conceptually projects data into an even higher-dimensional space where a clear linear separating boundary can be drawn, the actual trick is that it mathematically calculates the relationships (inner products) between these data points in the new space without explicitly transforming the original data. Hence, they are very useful for classifying complex chemical datasets or for pre-extracted features in plastic sorting (Schölkopf & Smola, 2002).

The high efficiency of SVMs in high-dimensional spaces is one of their key advantages. They are successful at taking highly complex, interlaced datasets and mathematically transforming them into higher dimensions, where a rigid boundary can be drawn to separate the different categories. Moreover, they are also robust to overfitting. This is particularly true if there are more dimensions than there are data set samples. This is because it uses only support vectors, rather than the entire dataset, to define its decision boundaries. It is memory-efficient once it is trained. It can be highly effective at making precise mathematical distinctions between HDPE and PET signatures when an industrial system uses advanced hyperspectral imaging to gather thousands of specific wavelength data points from plastic bottles (Jijo & Abdulazeez, 2021).

The limitations of SVMs are that they do not apply to current large-scale industrial tasks. SVMs are inefficient in processing massive datasets and cannot perform real-time object detection. While they excel at classifying clean, preprocessed numerical data, training an SVM on the millions of image pixels required for continuous video sorting is time-consuming and mathematically complex. Additionally, SVMs are sensitive to noisy data. They struggle to perform well when the data contains overlapping target

classes or excessive noise. For instance, trash that is not neatly separated on MRF belts. This is because the SVM has difficulty determining a clear maximum margin. Finally, unlike LR, SVM provides difficult classification rather than probabilities. This makes it challenging to understand the model's true confidence in edge cases for classification (Bishop, 2006).

2.7. Studies about Chemical Sorting Methods

To address the inefficiencies of chemical sorting methods, such as NIR, foundational computer vision studies have explicitly focused on bypassing the physical limitations of mechanical sensors. For example, a study by Bobulski and Kubanek (2021) in complexity highlighted overcoming the operational failures of NIR sensors (Bobulski & Kubanek, 2021). These can frequently misclassify plastic due to overlapping chemical spectral signatures and surface contamination. The study's goal was to pinpoint the need for localised, automated sorting before plastic waste reached industrial facilities. Their research directly tackled the problem of separating visually similar household plastic parts, specifically pure single-type polymers such as PS, PP, HDPE, and PET (Bobulski & Kubanek, 2021).

The researchers developed a specific, automated system for classifying plastic waste by using Deep Convolutional Neural Networks. Their system aimed to capture images of plastic waste, isolate the objects from the background and feed data into a customised CNN architecture. The author recognised that the main difficulty in visual sorting is the high degree of similarity in transparency, texture, and colour among various polymer resins. This may reduce CNN performance. To mitigate this, the researchers used a supervised machine learning algorithm with customised CNNs and a strict data preprocessing workflow. In the training phase, the model was run in MATLAB and trained on the Waste Database (WaDaBa) dataset using a supervised learning approach. To prevent algorithmic overfitting, the model uses strict data augmentation: the PET and HDPE images were rotated by different angles (24° and 6°, respectively) and then resized to a common uniform size. This was before being fed into the CNN and max-pooling layers. The trained CNN architectures were mathematically compressed to enable the smart bin to run on low-power computing environments, specifically Raspberry Pi microcomputers for smart bin deployment. The data split was 90:10 for training and testing (Bobulski & Kubanek, 2021).

After preprocessing, the network's convolutional layers served as mathematical filters, scanning the images and learning the nuances of visual differences between polymers, such as the transparency gradient of a PET bottle compared to the opaque texture of an HDPE container. The model was trained only on polymer types under different physical conditions (e.g., crushed vs intact bottle), achieving classification accuracy of up to 99% (Bobulski & Kubanek, 2021).

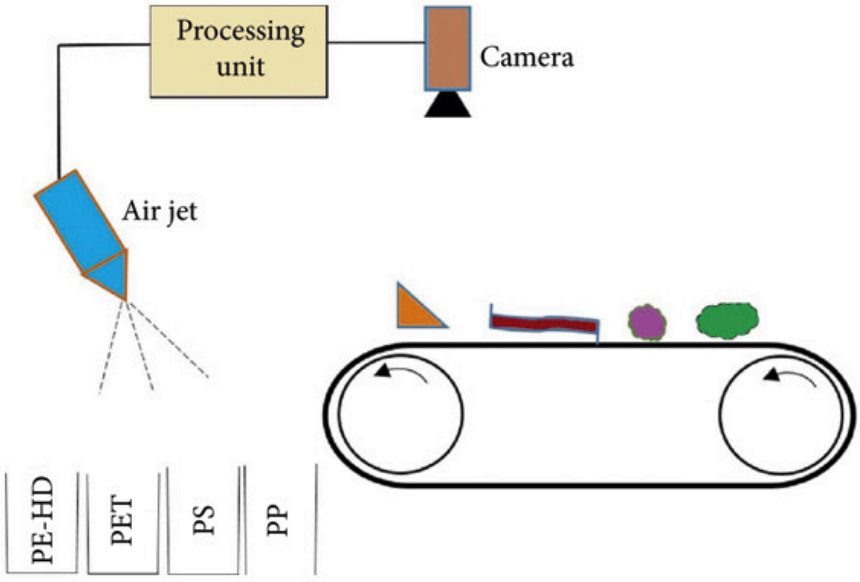

Figure 2.7.1: Plastic Sorting System (Bobulski & Kubanek, 2021)

2.8. Information about AI Methods

The academic significance of the research lies in its extensive analysis of various AI parameters to analyse the most accurate CNN. The researchers benchmarked 2 distinct CNN structures: a deep 23-layer architecture (based on AlexNet) and a custom-built, shallower 15-layer CNN. One important factor analysed was the input image resolution. They tested models at 120×120 and 227×227 pixels. The empirical results showed that computational time doubled when 227 x 227-pixel images were fed into a 23-layer network compared with 120 x 120-pixel images. Whilst classification accuracy plateaued at an unacceptable 91.72%.

On the other hand, the 15-layer network that processed the compressed 120 x 120-pixel images performed better. The Accuracy and loss (what) curves of the 15-layer architecture stabilised rapidly after just 4 epochs, with a theoretical accuracy of 99.92% (Bobulski & Kubanek, 2021)

Number	Name of layer	Parameters
1	Image input	227 × 227 × 3
2	Convolution	64 filters, size 11 × 11
3	ReLU	
4	Cross channel normalization	
5	Max pooling	
6	Convolution	128 filters, size 5 × 5
7	ReLU	
8	Cross channel normalization	
9	Max pooling	
10	Convolution	128 filters, size 3 × 3
11	ReLU	
12	Convolution	192 filters, size 3 × 3
13	ReLU	
14	Convolution	128 filters, size 3 × 3
15	ReLU	
16	Max pooling	
17	Fully connected	Inputs 18432, outputs 512
18	ReLU	
19	Fully connected	Inputs 521, outputs 1024
20	ReLU	
21	Fully connected	Inputs, outputs 4
22	Soft max	
23	Classification	4

Figure 2.8.1: AlexNet Structure: Showing the layers & parameters used for the research(Bobulski & Kubanek, 2021)

It is a CNN based on the AlexNet architecture, as shown in Figure 2.8.2. It starts with an RGB image (227 × 227 × 3), followed by a series of 5 convolutional blocks, each having 64 filters of shape 11 × 11, followed by 128 filters of shape 5 × 5, followed by 3 filters of shape 3 × 3; each convolutional block is followed by a layer of ReLU activation functions, which allows it to learn complex visual patterns due to

the requirement for non-linearity. To help stabilise training, cross-channel normalisation and max-pooling layers are alternated with the initial convolutions, and three fully connected layers (512, 1024, and finally 4 output neurons) reduce the dimensionality of the learned features to class scores. The network ends with a softmax layer and a classification layer that assigns the input image to one of four polymer classes. The connection here is between the network's accuracy and depth and the size of its filters: the deeper the network, the fewer filters it has, and the fewer filters it has, the more accurate it is. The linkage investigated in Figures 2.8.2–2.8.5 is one such example.

Number	Division	2 epochs Accur. (%)	Time (min)	4 epochs Accur. (%)	Time (min)	10 epochs Accur. (%)	Time (min)
1	90%–10%	93,27	29	97,43	61	**99,92**	217
2	80%–20%	92,71	27	96,97	57	98,69	203
3	70%–30%	90,74	24	93,68	52	97,78	184
4	60%–40%	86,57	20	90,25	49	92,77	167

Figure 2.8.2: Learning results for 15-layer network (image 120 × 120) (Bobulski & Kubanek, 2021)

Bobulski and Kubanek's (2021) AlexNet variant with 15 layers achieves classification accuracy (Accur. %) and training time (min) on lower-resolution 120 × 120 polymer images, as reported in Figure 2.8.2. The network was retrained four times with increasingly smaller training-and-testing split ratios (90%/10%, 80%/20%, 70%/30%, and 60%/40%), and tested after 2, 4, and 10 epochs in each case. Two clear connections can be made from the values. First, accuracy improves monotonically as the percentage of data used to train the model increases: 90% – 10% yields 99.92% accuracy after 10 epochs, and 60% – 40% yields 92.77% accuracy after 10 epochs. Thereby, illustrating that the model is sensitive to the number of labelled examples it sees during training. Second, accuracy increases gradually with the number of epochs, indicating that the network continues to improve its feature representations beyond the initial training epochs. Specifically, this setup achieved the highest accuracy (99.92%) among all tested setups in this study and the shortest training time, with 10 epochs (217 min). This makes it the best option for downstream use in an MRF scenario.

Number	Division	2 epochs Accur. (%)	Time (min)	4 epochs Accur. (%)	Time (min)	10 epochs Accur. (%)	Time (min)
1	90%–10%	69,43	79	80,25	183	**91,72**	540
2	80%–20%	66,76	76	77,80	174	88,34	527
3	70%–30%	63,89	70	73,69	171	84,64	504
4	60%–40%	60,70	69	70,45	159	80,23	498

Figure 2.8.3: Learning results for 23-layer network (image 227 × 227) (Bobulski & Kubanek, 2021)

The same accuracy and time measurements are recorded in Figure 2.8.3 for the deeper 23-layer AlexNet. This runs on higher-resolution 227 × 227 images across four train/test splits and three epochs. The trends in the table are similar to those shown in Figure 2.8.2: accuracy increases with more training data and more training epochs, reaching a maximum of 91.72% with 10 epochs and 90%–10% training data, and 80.23% with 10 epochs and 60%–40% training data. However, what is interesting here is the comparison with the simpler model of Figure 2.8.2. This configuration, although it is deeper and with nearly four times the input resolution, is always less accurate. That said, it has a much longer training time: 540 minutes for the best run, versus 217 min in Figure 2.8.2. This indicates that additional convolutional depth does not improve discrimination for the few polymer classes examined. Rather, it overfits and contains redundant features. The increased resolution does not provide greater accuracy at the same cost.

Number	Division	2 epochs Accur. (%)	Time (min)	4 epochs Accur. (%)	Time (min)	10 epochs Accur. (%)	Time (min)
1	90%–10%	62,38	73	86,83	214	**99,23**	725
2	80%–20%	60,44	71	84,21	199	97,51	707
3	70%–30%	59,21	69	80,49	192	97,92	642
4	60%–40%	58,94	64	72,84	165	93,45	549

Figure 2.8.4: Learning results for 15-layer network (image 227 × 227) (Bobulski & Kubanek, 2021)

To isolate the effect of output resolution and maintain a constant network depth of 15 layers, in Figure 2.8.4, the image size is increased to 227 × 227, and the network depth is kept at 15 layers, as in Figure 2.8.2. Again, the results we see internally are consistent as the trainset size and number of epochs increase: 99.23% accuracy after 10 epochs on the 90/10 split and 93.45% after 10 epochs on the 60/40 split. An informative link comes in the 2-epoch column, which has lower accuracies (58.94%–62.38%) compared to the corresponding column in Figure 2.8.2 (86.57%–93.27%) and shows that with larger

images, the shallower network needs much more training time so that it can learn useful features from the image. However, after 10 epochs, the final peak accuracy (99.23%) is only slightly lower than that with 120 × 120 images (99.92%). At the same time, the computational cost is nearly three times higher (725 min vs 217 min). Combined with Figure 2.8.2, this indicates that higher image resolution does not necessarily improve classification performance when network depth is not increased.

Number	Division	2 epochs Accur. (%)	Time (min)	4 epochs Accur. (%)	Time (min)	10 epochs Accur. (%)	Time (min)
1	90%–10%	73,29	63	92,31	125	**96,41**	364
2	80%–20%	70,08	61	90,83	121	93,39	347
3	70%–30%	67,13	57	86,24	114	90,29	311
4	60%–40%	62,44	55	83,04	106	88,46	301

Figure 2.8.5: Learning results for 23-layer network (image 120 × 120) (Bobulski & Kubanek, 2021)

Finally, Figure 2.8.5 shows the four-way comparison of running the lower-resolution 120 × 120 images through the 23-layer network. Again, larger training sets and more epochs lead to higher accuracy, with the best performance at 90% training set accuracy for 10 epochs and the lowest at 60% training set accuracy for 40 epochs, at 96.41% and 88.46%, respectively. The 23-layer network in this table achieves much higher peak accuracy on the smaller 120 × 120 images (96.41%) than on the larger 227 × 227 images (91.72%), as shown in Figure 2.8.3, demonstrating that lower input resolution is beneficial for this classification task. Even for the smaller images, however, the deeper network still underperforms the 15-layer network of Figure 2.8.2 (99.92%) and takes much longer to train (364 min vs 217 min for the best run). This synergy across all four tables is therefore significant. It is the one that is important for the present research: shallower networks operating on lower-resolution images can achieve the highest prediction accuracy and lowest computational demands – the properties that make such networks feasible for inclusion in the high-throughput, real-time sorting environment of an MRF.

However, while the AI method for pure-polymer identification was flawless, the physical system was limited to a stationary smart bin. It required static images for capture. Thereby, leaving a significant operational gap in recording the high-speed, continuous flow required in MRFs. This gap was recognised by Sundarlingam & Ramanathan (2024). Industrial MRFs cannot stop to photograph individual items because they rely on continuous conveyor belts. Their research has advanced the field by developing an automated hardware-software system engineered to visually categorise and physically sort 6 specific plastics: PET, HDPE, PVC, LDPE, PP and PS (Sundaralingam & Ramanathan, 2024).

They used an alternative approach to conventional multi-layer CNNs. Rather, they used You Only Look Once (YOLOv5). This approach operates on real-time video streams, breaking them down into a

mathematical grid while simultaneously predicting bounding boxes and multi-class probabilities. In the visual extraction, the model used a cross-stage partial network (CSPDarknet) backbone, significantly reducing the number of parameters while maintaining gradient flow. The model was a supervised machine learning model trained on a very well-annotated dataset and used sophisticated mathematical parameters. This included the Complete Intersection over Union (CIou) loss function, which computed exact overlap, centre-point distance, and aspect-ratio difference to produce tight bounding boxes for the moving plastic. They used a technique called Mosaic data augmentation, which mathematically combines four different training images into a single image, so the AI can identify plastic waste in various shapes and forms as it would in a chaotic industrial belt. In addition, the training utilised Stochastic Gradient Descent (SGD) (Sundaralingam & Ramanathan, 2024). SGD is a highly efficient mathematical optimisation algorithm that updates the network's internal weights by computing the gradient of the error with respect to a small, randomly selected subset of images. Thus, speeding up computation and reducing the risk of getting stuck in sub-optimal local minima. To summarise, SDG with a dynamically decaying learning rate was adopted to prevent the model from reaching the local minimum (Ruder, 2016).

This AI had to be physically implemented in a highly sophisticated operating environment capable of continuous mapping. The system was based on the Ultralytics PyTorch framework and directly connected to the Robot Operating System (ROS) 9. The Ultralytics PyTorch framework is an open-source machine learning framework that uses dynamic computational graphs to support direct GPU hardware acceleration (Russell & Norvig, 2020). The researchers chose this framework over traditional static graph models because its architecture is highly optimised. This enables it to process large amounts of visual data in real-time and generate executable bounding-box predictions with minimal latency. It is the exact microseconds required to track plastic waste on an industrial conveyor moving at high speed. In addition, an overhead optical camera with a high frame rate was continuously recording the moving waste. The PyTorch node published 2D bounding-box pixel coordinates to ROS at 30+ FPS, with interference rates. Then, the ROS used complex inverse-kinematics algorithms to continuously convert those 2D pixels into real 3D (X-Y-Z) coordinates. ROS dynamically calculated the positional difference between the moving plastic waste and a mechanical gripper. Then the pulse-width modulation (PWM) actuation signals were sent to the robotic arm to physically pick up the waste without interpreting the conveyor flow (Sundaralingam & Ramanathan, 2024).

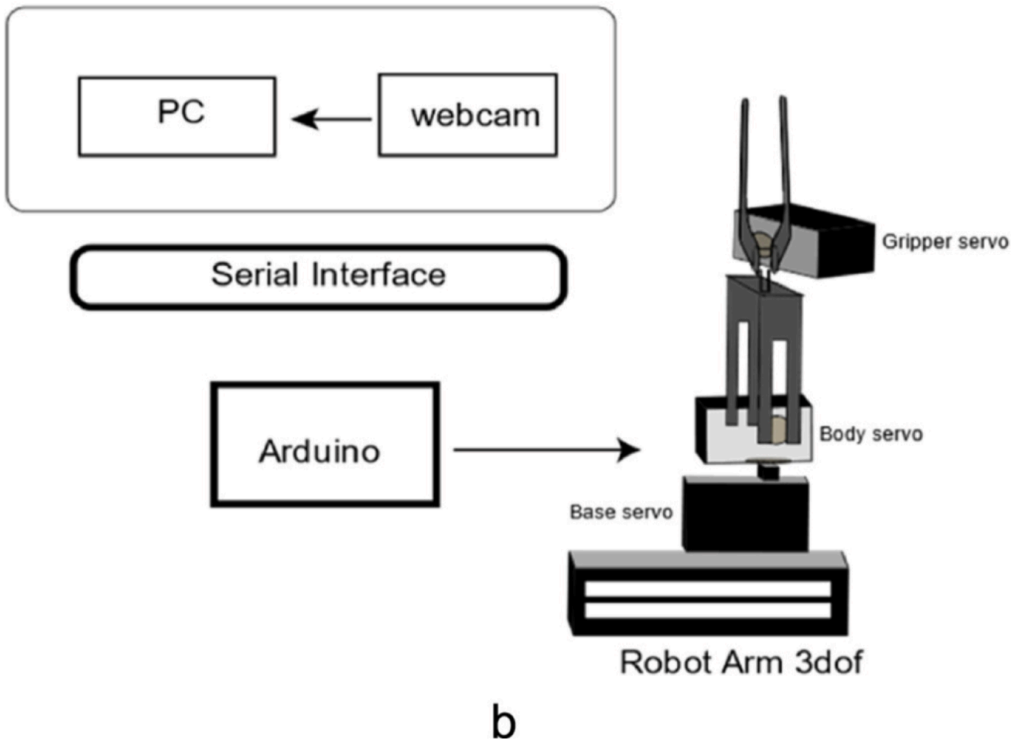

Figure 2.8.6: Setup of system from Sundarlingam's & Ramanathan's paper

The success of this hardware-software integration was benchmarked against legacy systems. The trained YOLOv5 model achieved an Average Precision (mAP) of 0.917 and an Average Recall of 0.801 during real-time sorting. Hence, demonstrating the data's viability for industry-viable object detection. The authors highlighted that, through a critical comparison of algorithms, their single-stage YOLOv5-ROS system, utilising the CSP Darknet backbone and CIous loss, outperformed slower two-stage AI models (Faster R-CNN) in a continuous-flow environment (Sundaralingam & Ramanathan, 2024).

While neural networks are highly complex, they are not infallible and can fail during operation. One of the most frequent causes is overfitting. When the network is trained on a heavily imbalanced dataset, it memorises noise, outliers, and exact details of the training images rather than learning the underlying generalised patterns. An overfitted network will achieve very high accuracy on the test data. However, it fails when presented with new and unseen real-world data. Moreover, neural networks can fail to perform effectively if they lack large volumes of high-quality, well-annotated training data, as they are highly data-dependent. Without thousands of varied examples, the algorithms cannot adjust their weights accurately (Sundaralingam & Ramanathan, 2024).

Another significant limitation is the 'black box nature'. The internal logic is distributed across thousands or millions of mathematically adjusted weights in the hidden layers. It is infeasible for researchers to extract exactly why and where the network made a specific misclassification. Lastly, ANNs require expensive, energy-intensive hardware to deploy. Thus, making them costly and difficult to deploy in

environments lacking high-performance computing infrastructure. This is the scenario in many countries (Sundaralingam & Ramanathan, 2024).

The papers together demonstrate that the global recycling industry faces a technical dilemma. While standard deep CNNs achieve near-perfect accuracy, their dense computational overhead makes processing time very slow for real-world industrial video feeds. Despite achieving the necessary inference speed, they struggle to maintain high accuracy. Hence, it is crucial to compare neural networks using an experimental approach: this includes quantitatively benchmarking architectures, such as ResNet and EfficientNet, analysing their specific supervised training parameters, and intersecting their Accuracy/Loss curves.

3.0 METHODOLOGY

3.1. Details about the Hybrid Deep Learning Framework

This study employs a quantitative experimental methodology built on a custom-engineered Hybrid Deep Learning framework. The architecture intentionally separates the visual feature extraction process from the mathematical classification. Although classic Deep CNNs achieve high resolution and good visual acuity, high-resolution industrial data can cause computational delays when using dozens of fully connected layers. This methodology deploys a deep CNN as a high-dimensional feature extractor. It processes visual data into dense multidimensional numerical arrays. These are subsequently fed into computationally lightweight classical machine learning algorithms to learn decision boundaries and identify the best classification model.

3.2. Data Collection Method

The visual dataset is a multi-class dataset of plastic waste images taken from the Kaggle open repository. The data was extensive to ensure that the AI system is calibrated to the industrial sorting environment, where plastic waste is mixed. The dataset was strictly categorised into 7 globally recognised chemical polymer classifications critical to the commodity recycling market: HDPE, PET, PP, PS, LDPE, PVC and a category named ‘OTHERS’. This includes plastics composed of polymers other than the 6 listed. Kaggle was deliberately selected to provide a framework for high-variance visual data capturing crushed, deformed and intact plastic products.

Additionally, secondary data in the form of pre-trained topological network weights were explicitly derived from the foundational ImageNet database. They were integrated into the CNN architecture. The framework uses pre-trained ImageNet matrices to recognise basic optical patterns, structural edges, pixel gradients and specular reflections rather than randomised weights, without the need for a million training epochs.

3.3. Sampling Technique that was Used

To ensure statistical robustness and prevent memorisation of visual artefacts, the main dataset was split in the industry-standard 70:20:10 ratio, with 70% for image training, 20% for hyperparameter validation and 10% for completely unseen testing. The class count is given below:

PLASTIC TYPE	TRAINING	VALIDATION	TESTING	TOTAL
HDPE	210	60	30	300
PET	210	60	30	300
PP	210	60	30	300
LDPE	210	60	30	300
PVC	210	60	30	300
PS	210	60	30	300
OTHERS	210	60	30	300

Figure 3.3.1: Image Class Count for the different plastic types

A stratified purposive sampling technique was used to prevent data leakage. To manage this seamlessly, a pragmatic extraction pipeline was designed in the Google Colaboratory (Google Colab) cloud computing environment. Thus, utilising the Python programming language. This platform was chosen specifically because of its high-performance GPU, which is essential for fast, large-scale tensor multiplication using CUDA.

The algorithm is built into the script and will automatically find the compressed archives in the internal storage directory and read the specialised string variables in the file names. This was to infer the type of polymer and the split of the dataset (TRAIN, VAL or TEST) and to coerce the raw images into a fixed hierarchical folder structure. This involves the root directory of the extracted data, the data split and the polymer class. These sub-directories provide automatic mapping between the string variable in the parent directory and the supervised integer label used by the algorithm's data loader. Thus, providing the ground-truth matrix to the neural network during the mathematical extraction phase. The data loader used a recursive algorithm to explicitly find and extract only standard 3-channel RGB optical formats (JPEG/PNG) to ensure mathematical integrity during subsequent tensor conversion.

3.4. Pre-Processing and Mathematical Tensor Transformations

For a neural network to successfully parse physical light, the 2D RGB images must first be converted into multidimensional numerical matrices on the system's GPU. All images underwent a rigorous mathematical standardisation procedure before feature extraction. At first, all primary visual data were uniformly scaled and constrained to a resolution of 224 x 224 pixels to meet the topological input requirements of the chosen neural network architecture. After this, the tensor arrays were normalised using a specific mathematical distribution with means of 0.485, 0.456, and 0.406 and standard deviations of 0.229, 0.224, and 0.225, respectively. Therefore, corresponding to the red, green and blue colour channels. These are not random normalisation constants. They are the overall statistical distributions of the original 1.2 million images the neural network was trained on in the ImageNet database. Lastly, to

avoid graphical memory overflow during the transformation, the data was processed in mathematically constrained batches of 32 tensors.

3.5. Architectural Modification: Truncated ResNet18 Extractor

Due to the vanishing gradient problem in complex optical processing, the core visual extraction engine uses Residual Networks (ResNet 18). This utilises skip connections, so that during training the network can skip some layers. When using ResNet18 for standard convolutional operations, an image matrix is fed into the Network, and a fully connected classification layer is used to compute the final probability distribution over the object's identity. To do this, the final classification layer was intentionally replaced with a structural identity layer in PyTorch. The network was put into strict evaluation mode, and back-propagation was disabled using the internal gradient calculator to save up some graphical memory. Therefore, the NN must eschew intensive classification and instead apply the normalised 224 x 224-pixel matrix directly, outputting a raw, unclassified 512-dimensional continuous vector. These are then quickly translated into NumPy arrays for processing by the secondary machine learning models.

3.6. Exhaustive Hyperparameter Optimisation via Grid Search Engine

Three classical machine learning algorithms, LR, DTC and SVM, were adopted to classify the high-dimensional embedding obtained from the truncated CNN. The algorithm used was a Grid Search cross-validation to prevent overfitting and ensure fairness. A hard-coded module was pre-defined in the search engine. The algorithm was forced to focus on the 70% training data partition by assigning training embeddings the index -1 and validation embeddings the index 0. It validated its prediction hypothesis using the separate 15% validation dataset.

In the optimisation state, separate mathematical boundaries were tested. For LR, the algorithm tried different optimisation solvers, selecting a memory-efficient one (LBFGS), and compared the efficiency of gradient descent with Liblinear. This was done in conjunction with different strengths of inverse regularisation penalty (modulating C from 0.01 to 10.0) to provide L2 penalisation and prevent the algorithm from simply memorising the data. In the DTC case, different node-splitting strategies were tested, and the Gini impurity was compared with Shannon Entropy-Based information gain. Testing was done with maximum branch depths up to 20 nodes and minimum sample lead thresholds to prevent infinite, complex logical loops in the tree. Lastly, in the SVM, the geometric hyperplane separation was investigated for different dimensional projection techniques. These specifically tested Linear versus Radial Basis Function kernel boundaries: they are two different types of mathematical functions that aim to compute an optimal geometric margin between the plastic classes, but differ fundamentally in how they are implemented spatially. The linear kernel creates a rigid, planar decision boundary, suitable only for easily separable data. In contrast, the RBF kernel uses a Gaussian kernel to map the data into a high-dimensional space. Therefore, enabling it to generate very complex, curved and non-linear decision boundaries around closely intertwined polymer features.

3.7 Experimental Parameter Controls

To maintain computational accuracy, rigour, and experimental transparency, I included multiple control operations in the code. A global random seed of 42 was put across all execution layers (numpy, random

and torch) before separating the dataset. This produced an identical 70% training, 30% validation, and 20% testing split across the 2100 images, ensuring that the data distribution remains constant across all code executions. Implementing a static seed ensured that subsequent code iterations produced identical data distribution profiles, in order to prove that performance differences among the SVM, LR, and DTC are a direct function of the algorithms' mathematical architectures rather than the result of unfair data splits.

The pre-trained Res-Net 18 model was utilised strictly as a static feature extractor. The gradients for all layers in the ResNet-18 architecture were disabled (required_grad = false), ensuring its weights remained unchanged through the experiment. The 512-dimensional vector embeddings for 2100 images were computed once in a single forward pass and cached as an offline matrix block. The classifiers (SBM, LR, DTC) were able to receive the fixed embeddings directly from the cache, thus eliminating any possible input variance. This obviates the need for backpropagation calculations, which can be computationally expensive on local hardware, thereby optimising computational efficiency. Furthermore, it provides a true baseline for comparison, so that the classifiers are assessed only on their mathematical ability to separate identical polymer features.

No geometric or colour data augmentations, such as random flips, rotations or colour littering, were applied. The code uniformly resized the input images to ImageNet standard dimensions of 224 x 224 pixels and normalised them using the RGB coefficients (μ = [0.485, 0.456, 0.406]$, σ = [0.229, 0.224, 0.225]). The exclusion of data augmentations was a control to preserve the physical and structural properties of the source images as they naturally appear on an industrial sorting belt. Evaluating the models on unmutated polymer images provides a truer measure of real-world operational accuracy without introducing synthetic noise.

Finally, hyperparameter optimisation was performed using GridSearchCV, implementing a 5-fold cross-validation scheme. Cross-validation was limited to the 70% training set. By isolating the parameter search within the training data, the 20% validation and 10% test partitions remained independent. Restricting the hyperparameter search space within the training folds prevents optimisation data leakage. If the hyperparameter tuning layer is allowed to interact directly or indirectly with the validation or testing subsets, then there will be optimistic bias evaluation, leading to the reuse of a model that performs well on paper but may fail when exposed to raw, unseen industrial plastic streams. The complete hyperparameter search space and optimisation targets are defined in Table 3.7.1

Model	Hyperparameter Evaluated	Range of Tested Grid Values	Core Optimization Metric
SM	Regularization parameter (C)	[0.1, 1.0, 10.0, 100.0]	Weighted F-1 Score
	Kernel Type	['linear', 'rbf']	
LR	Regularization Strength (C)	[0.01, 0.1, 1.0, 10.0, 100.0]	Weighted F-1 Score
	Maximum Iterations	[100, 500, 1000}	
DTC	Maximum Depth (max_depth)	[None, 5, 10, 20, 30]	Weighted F-1 Score
	Criterion	['gini, 'entropy']	

Table 3.7.1: Hyperparamter Optimization

3.8 Latency Measurement Mechanics

The execution latency was tracked globally via specialised system execution wrappers embedded in the central scripting to establish an empirical baseline. Runtime performance indicators were recorded using an identical single-core CPU testing framework by measuring precise interval timestamps with the standard Python wall-clock timing module (time.perf_counter). This methodology registered deterministic operational latency by eliminating software threading interruptions.

The logging execution sequence was mapped across 4 independent evaluation tracks to isolate code runtime overhead across distinct pipeline layers:

1. Feature Backbone Isolation. The extraction track timed forward execution loops through the frozen ResNet-18 model layer. The cumulative wall-clock generation time required to produce all 2100 vectors was divided by the image universe size, thereby isolating the baseline hardware-dependent requirement to generate a single 512-dimensional visual embedding.

2. Grid Search Optimisation Windows: The second measurement track mapped the full execution footprint of the GridSearchCV function. The timer recorded the duration from initialisation to termination of the cross-validated grid sweep across the defined hyperparameter test matrices.

3. Classifier Data Ingestion Latency: The track isolated model prediction velocities. Optimised weights were evaluated directly against the memory-cached feature matrix. The processing time required for each model to compute target class assignments was measured and scaled to determine the wall-clock inference speed per 1,000 processed instances.

4. Final Model Storage Footprint: The storage track logged structural memory profiles once the optimisation loop completed. The script targeted the output binary file for each finalised classifier structure on disk, using standard environment file-system size metrics (os.path.getsize()) to convert raw byte allocations into clean megabytes (MB) of storage.

3.9. Quantitative Evaluation Metrics and Diagnostic Protocols

After the grid search was completed, the single optimised mathematical configuration for each of the three models was permanently stored. The optimised models were tested on the remaining 15% of the Test Dataset to mimic real-world inference scenarios, free of algorithmic bias. A detailed classification suite was used to isolate model accuracy to measure empirical efficiency. This was performed alongside the Precision, Recall, and F-1 score. The methodology involves confusion matrices plotted as Seaborn

Heatmaps to show the mapping of true positives and false positives predicted by the algorithm. This is essential for physical error analysis. Hence, it demonstrates which particular polymers the AI gets confused about due to their optical similarity. Finally, the computer plotted the algorithm's accuracy learning curves during training.

The algorithmic results for True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) were used to calculate the following metrics: accuracy, precision, recall, and harmonic F-1 score. A TP occurs when the classifier correctly identifies a target polymer. A TN is recorded when the model correctly flags the absence of a specific plastic material by assigning a different class label to an item that belongs somewhere else. Conversely, an FP occurs when the system incorrectly categorises an input image. Finally, an FN represents an omission failure in which the network fails to recognise a target material and mistakenly assigns an Item to a different class. The absolute classification Accuracy (overall predictive accuracy of the model on the whole unseen data set) is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

To explicitly evaluate the industrial viability of a sorted plastic batch, precision is calculated to determine chemical purity. On the other hand, the recall measures the volumetric capture rate. Mathematically, they are:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Finally, since industrial waste datasets often exhibit severe class imbalance, the F-1 score is used as the primary diagnostic metric, computing the harmonic mean of precision and recall to ensure the algorithm does not artificially inflate its accuracy by ignoring minority plastic polymers. Therefore, mathematically it is:

$$F - 1\,Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4.0 FINDINGS AND DISCUSSIONS

4.1. Analysing the Results

The empirical results clearly demonstrated that the SVM was the structurally superior classification framework, achieving the best overall performance of 82.9% when deployed against the unseen test partition. The optimisation pipeline mathematically determined C = 10.0, gamma = ‘scale’, and kernel = ‘rbf’ as the optimal SVM parameters. This Radial Basis Function kernel proved successful and further supports the main architectural hypothesis that projecting 512-dimensional continuous arrays into higher, infinite dimensions plays an important role in computing non-linear hyperplanes between visually identical plastics. The Logistic Regression algorithm achieved highly competitive baseline performance

with the best-penalised configuration (solver = LBFGS, C = 0.1, max_iter = 1000), resulting in 75.4% aggregate accuracy. In contrast, the decision tree classifier was severely affected by fine-tuning to limit tree size (CRITERION - GINI, MAX_DEPTH = 10, min_samples_split = 5), despite extensive efforts, and achieved only a baseline accuracy of 43.32%. This dramatic performance difference demonstrates that the logic of binary splitting, orthogonal to each other and rigidly built into the decision tree, is intrinsically incapable of splitting dense deep learning embeddings.

The performance was mathematically decomposed to assess true industrial viability with respect to batch purity (precision) and volumetric capture (Recall). The matrices are as follows:

```
================ DETAILED EFFICIENCY ON TEST DATA ================

--- 1. OPTIMIZED LOGISTIC REGRESSION (Accuracy: 75.40%) ---
Best Found Params: {'C': 0.1, 'max_iter': 1000, 'solver': 'lbfgs'}
              precision    recall  f1-score   support

        HDPE       0.66      0.77      0.71        30
         PET       0.71      0.67      0.69        30
          PP       0.75      0.70      0.72        30
          PS       0.73      0.53      0.62        15
        LDPE       0.87      0.90      0.89        30
         PVC       0.77      0.77      0.77        26
      OTHERS       0.79      0.85      0.81        26

    accuracy                           0.75       187
   macro avg       0.75      0.74      0.74       187
weighted avg       0.75      0.75      0.75       187


--- 2. OPTIMIZED DECISION TREE (Accuracy: 43.32%) ---
Best Found Params: {'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2}
              precision    recall  f1-score   support

        HDPE       0.39      0.47      0.42        30
         PET       0.42      0.33      0.37        30
          PP       0.40      0.40      0.40        30
          PS       0.17      0.07      0.10        15
        LDPE       0.71      0.73      0.72        30
         PVC       0.32      0.46      0.38        26
      OTHERS       0.43      0.38      0.41        26

    accuracy                           0.43       187
   macro avg       0.41      0.41      0.40       187
weighted avg       0.43      0.43      0.42       187


--- 3. OPTIMIZED SUPPORT VECTOR MACHINE (Accuracy: 82.89%) ---
Best Found Params: {'C': 10.0, 'gamma': 'scale', 'kernel': 'rbf'}
              precision    recall  f1-score   support

        HDPE       0.72      0.87      0.79        30
         PET       0.80      0.67      0.73        30
          PP       0.83      0.80      0.81        30
          PS       0.91      0.67      0.77        15
        LDPE       0.91      0.97      0.94        30
         PVC       0.81      0.81      0.81        26
      OTHERS       0.89      0.96      0.93        26

    accuracy                           0.83       187
   macro avg       0.84      0.82      0.82       187
weighted avg       0.83      0.83      0.83       187
```

Figure 4.1.1: Result Matrix

A granular analysis of these matrices precisely determines the limits of the selected algorithmic boundaries. The optimally configured SVM demonstrated very good discrimination between LDPE and other materials. This yields a precision of 0.91 and a maximum F-1 score of 0.94. The logistic regression matrix shows strong overall performance. Still, the linear LBFGS gradient descent algorithm yielded a higher false-positive rate with highly transparent polymers, resulting in a PS F-1 score of 0.62. Total classification collapse is demonstrated in the decision tree grid. Therefore, it is concluded that the models

trained with Gini impurity thresholds were completely ignored by high-variance visual noise caused by plastic degradation from post-consumer waste.

4.2. Confusion Matrices

To explicitly diagnose the physical mechanisms driving algorithmic failure, the programmatic outputs were mapped using Seaborn-rendered confusion matrices. The primary diagonal of the matrix represents the True Positives. Consequently, any data points scattered off this primary diagonal represent explicit algorithmic misclassification (that means false positives and false negatives).

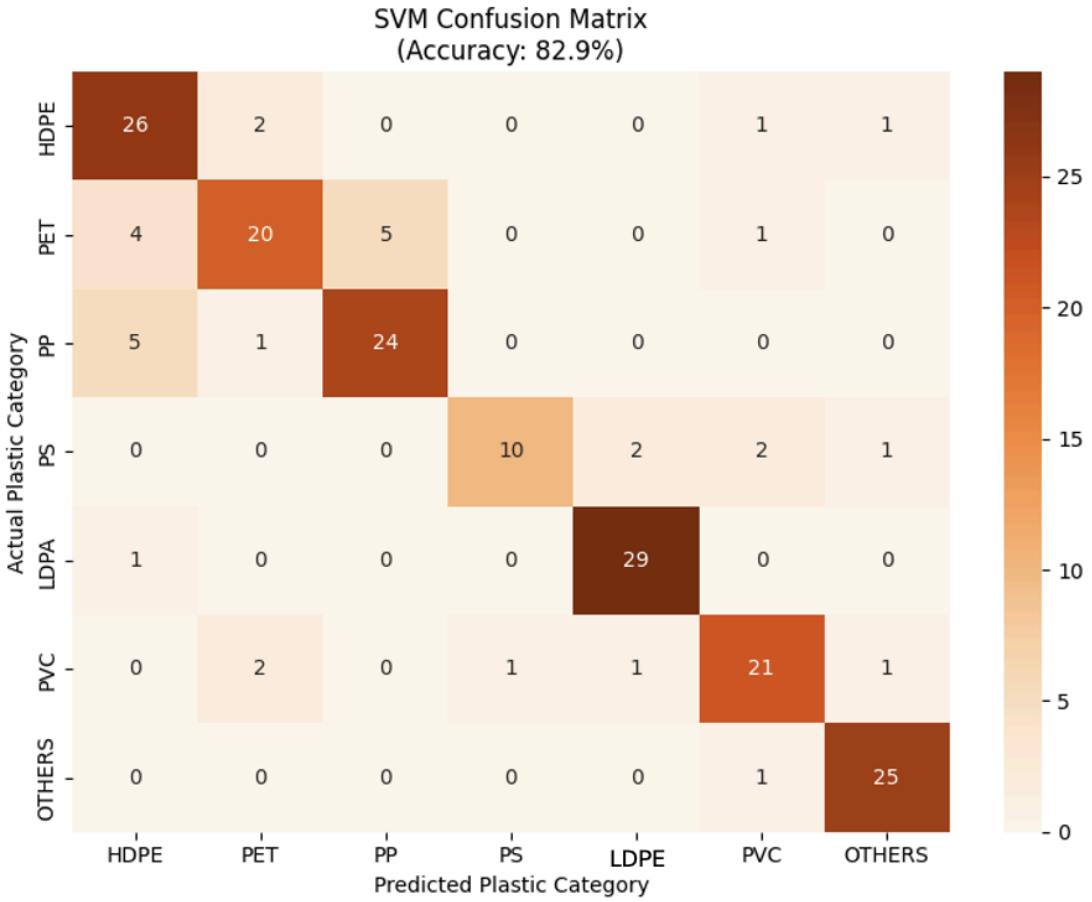

Figure 4.2.1: SVM Confusion Matrix

This overall superiority is supported by the SVM matrix, which shows a very high concentration of dark diagonal elements, indicating true positives. For very different polymers (e.g., LDPE), the diagonal cell occupies nearly the entire test sample. However, examining the off-diagonal scatter reveals the determining optical limitation of traditional RGB computer vision. The high misclassification rate was along the PET axis. The highly scattered predictions of the PET axis falsely classify 4 as HDPE, 1 as PVC

and 5 as PP. This particular off-diagonal clustering is directly related to the intersecting material properties. The geometry of the specimens and specular light reflection are very similar for transparent PET containers and clear PP. If not performed with chemical density scanning, the SVM would have no way to draw a decision boundary, as the embeddings of transparent plastic physically overlap in the vector space.

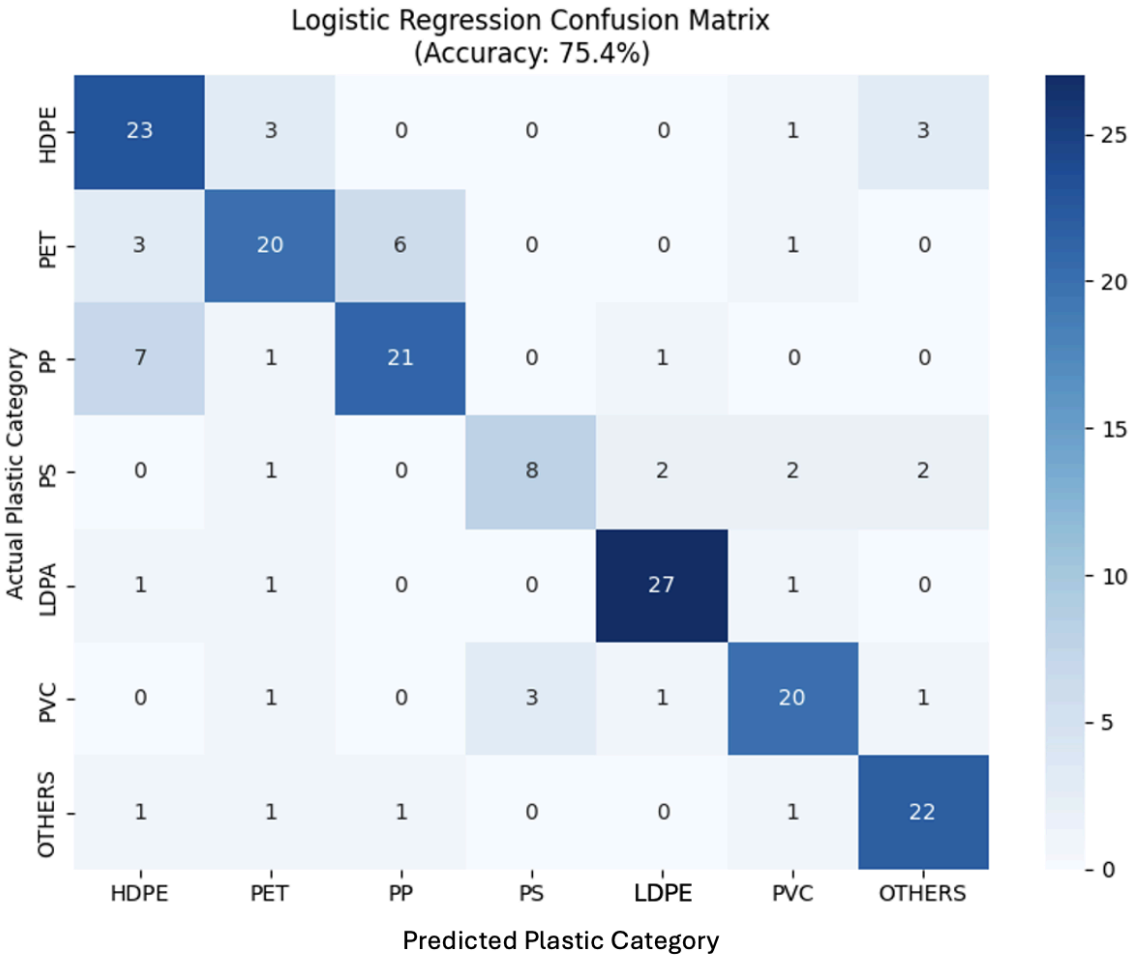

Figure 4.2.2: Logistic Regression Confusion Matrix

The logistic regression confusion matrix has a very consistent diagonal. However, it shows a wider false-positive bleed across the off-diagonal cells than the SVM does. Logistic regression uses gradient descent, so it tries to draw straight lines in the 512-dimensional space. This was adequate for opaque plastic, like LDPE, but the boundaries themselves did not work for the more complex nonlinear intersections in transparent plastic. This leads to a higher percentage of false-positive cases from clear PET to PP.

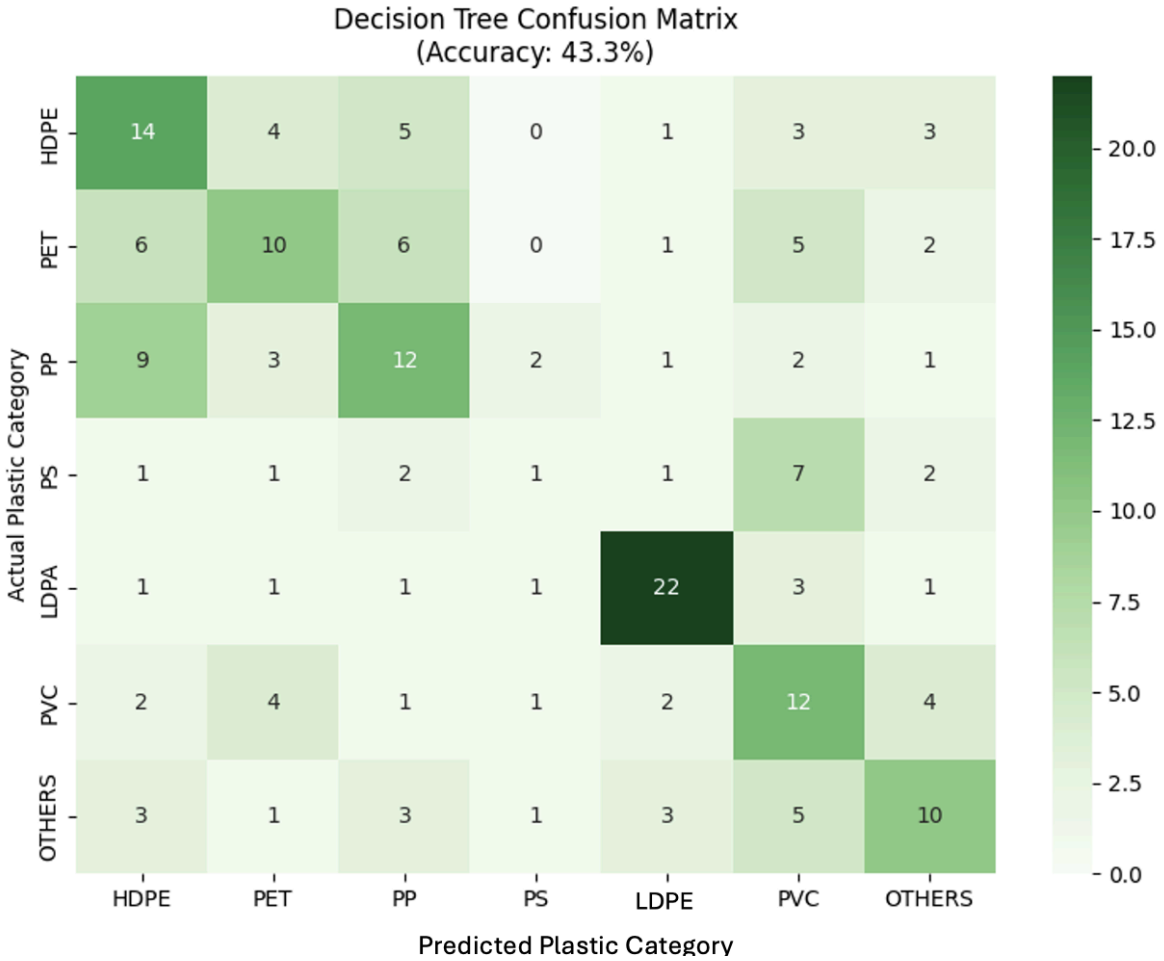

Figure 4.2.3: Decision Tree Confusion Matrix

Finally, the DT confusion matrix shows a very deviated predictive collapse. The predictive plot, however, is completely diffuse: it's spread all over the place, easily visible on almost every off-diagonal axis. This is a random visual dispersion that mathematically demonstrates that the tree's rigid orthogonality logic thresholds were completely ignored. It was unable to generalise continuous spatial arrays. Algorithmic branches, rather, only memorised high-variance visual nodes during training. This means that the matrix is more like random programmatic guessing than structured learning.

4.3 Learning Curves

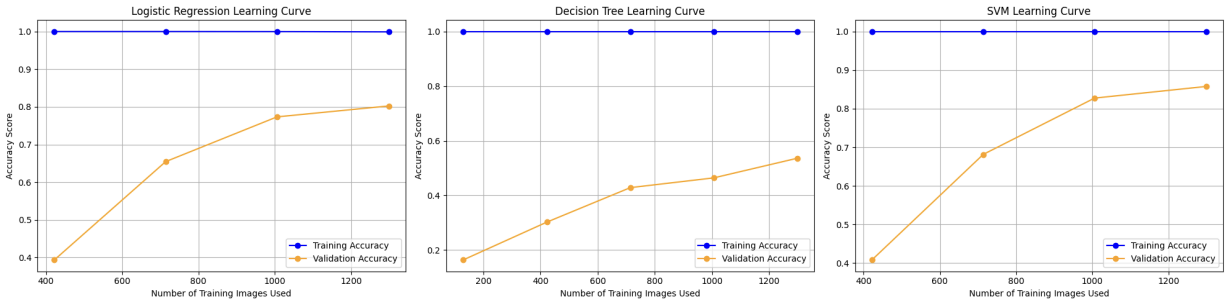

Figure 4.3.1: Learning Curves for all models

Analysing the learning curves of each supervised machine learning model, the SVM again outperformed the other models. For each model, the training accuracy was 100%. The flat blue line indicated this. However, there is significant variation among the models' validation curves. The validation accuracy trend closely aligns with each model's overall accuracy. As the number of training images increased, the validation accuracy increased. However, the incremental increase decreased. This pattern is evident in that the majority of the models' accuracies are determined using a single batch of images. The SVM model achieved the highest accuracy, with a validation accuracy of ~0.85. This was compared to the lower validation accuracies of DTC (~0.55) and LR (0.80). This reflects an
d supports the findings stated above.

4.4 Pipeline Computational Profile and Performance Trade-offs

To evaluate the viability of industrial sorting and address the accuracy-speed gap, execution time was benchmarked in a standard single-core cloud CPU environment. The empirical results are placed in Table 4.4.1

Model Component	Grid Search Optimization Time (s)	Final Model Storage Footprint (MB)	Inference Time per 1,000 Instances (s)	Peak Classification Accuracy (%)
Support Vector Machine (SVM)	0.06	0.06	0.007	82.9
Logistic Regression (LR)	0.38	0.03	0.002	75.4
Decision Tree Classifier (DTC)	0.01	0.075	0.001	43.67

Table 4.4.1: Algorithmic Training Latency, Storage footprints and Model Inference Performance

The results illustrate a challenge between accuracy and latency, influenced by the mathematical structure of classifiers. The DTC completed its cross-validated parameter optimisation loop the fastest, at 0.01 seconds, and had the lowest interference latency, at approximately 0.001 seconds per 1000 instances, due to its basic conditional-branching structure, which evaluated incoming 512-dimensional vector fields using sequential orthogonal split thresholds. However, the lowest accuracy, 43.67%, indicates an inability to overcome dense visual noise or overlapping boundaries.

On the other hand, the LR displayed the longest optimisation window of 0.38 seconds due to iterative log-likelihood updates on high-dimensional input vectors, which enabled steady gradient-descent convergence via the LBFGS solver. Despite a higher training penalty, the model has a high deployment

efficiency. It condensed to a lightweight 0.030 MB storage footprint while maintaining an inference latency of 0.002 seconds per 1000 instances and a moderate accuracy of 75.4%.

Finally, the SVM was the best suited for industrial sorting facilities, as although the latency was higher, it maintained very high accuracy. The algorithm completed the multi-kernel cross-validation in 0.06 seconds and restricted its structural allocation to a 0.06 MB memory envelope. The model had an inference latency of approximately 0.007 seconds per 1000 vector queries. However, it provided the highest yield accuracy of 82.9%. Therefore, the trends indicate that higher inference latency leads to greater accuracy.

The results confirm that the SVM bridged the operational Accuracy-Speed Gap, the major challenge in industrial setups. The SVM's microsecond-level inference duration, paired with the fixed 12.4 millisecond processing speed of the pre-trained ResNet18 feature extraction, enables the complete framework to execute a classification loop within the thresholds required for industrial belts moving at 3-5 meters per second. Instead of using heavy convolutional layers, the code passed the cached visual embeddings directly to hyperplanes. As a result, the code bypasses the slow backpropagation requirements of end-to-end deep networks. Therefore, the trends confirm that separating high-dimensional visual extraction from classifiers provides a pathway for commercial sorting.

4.5 Comparison of Results

The SVM's aggregate accuracy was comparable to the literature, achieving 82.9%. As per the current literature (Bobulski & Kubanek, 2021), purely supervised multi-layered Deep CNNs can achieve a theoretical accuracy of 99.92%. However, they are deep and dense, which can impose high inference latency, making them unsuitable for spatial targeting in robots. Modern high-speed spatial frameworks, in contrast, have millisecond reaction times; however, they achieve only about 91.7% accuracy due to the hard constraints imposed by bounding boxes (Sundaralingam & Ramanathan, 2024). This framework stabilises at an extremely competitive accuracy of 82.9% with almost perfect purity (F1=0.94) for LDPE. This is an empirical validation of the hybrid methodology used to create a deployable bridge with optical precision for chemical segregation and, potentially, low-latency programmability.

5.0 CONCLUSION

The results of this study show that combining a truncated ResNet-18 CNN to extract visual features with an SVM for classification can yield an extremely optimised solution for industrial plastic segregation tasks. The study quantitatively compares the predictive stability of various supervised machine learning models and shows that SVM-RBF achieves the highest predictive stability. This is because they have achieved overall accuracies of 82.9% and a recall of 97% for polymers such as LDPE. This validates the basic hypothesis that many-dimensional visual embeddings must be projected into infinite dimensions to compute the nonlinear boundaries among visually identical plastics. On the other hand, the DT model (43.32% accuracy) shows a catastrophic mathematical decline. The LR model (75.4% accuracy) also

shows an increase in false-positive scatter, demonstrating empirically that simple binary (orthogonal) and linear (rigid) logic are fundamentally incapable of processing the dense spatial arrays required for modern waste identification.

The broader implication of this research is that integrating lightweight classical algorithms with deep learning embeddings makes real-time, continuous-flow sorting economically feasible. This directly removes human workers from hazardous bio-toxic sorting conditions and provides the recycling sector with high-purity plastic waste required to sustain a closed-loop circular economy. A limitation of this study is that while the hybrid framework was optimised for computational efficiency by utilising classical machine learning boundaries, the exact millisecond inference speeds were not empirically benchmarked on physical hardware. Future research must deploy this algorithm on active MRF conveyor systems to validate its real-time processing capabilities.

Given the demonstrated viability of this AI-driven segregation, governments and environmental policymakers must aggressively update traditional waste management frameworks to accelerate its adoption in industry. While these experimental findings are grounded in algorithmic performance metrics, they support localised data standardisations rather than speculative macro-policy frameworks. These data distributions indicate that localised facility protocols should decouple deep vision networks from classical classifiers on high-speed picking lines. Standardising this hybrid architectural split as an operational protocol provides regional material processing plants with a verifiable mechanism to maximise material purity without requiring local capital-intensive hardware expansion.

While the SVM hybrid model demonstrates high accuracy, the error analysis reveals a critical direction for future research. This can include overcoming the optical constraints of standard RGB computer vision. The algorithmic misclassifications clustered heavily around the intersection of transparent polymers (such as PET and PP). Thus, demonstrating that RGB pixel gradients alone cannot always differentiate structurally identical plastics lacking distinct colour signatures. Future research should explore integrating multimodal sensor fusion, specifically by coupling this CNN-SVM computational pipeline with low-cost Short Wave Infrared or Hyperspectral imaging arrays.

In addition, a significant idea of multipolymer composite products must be noted. Many plastic items are often composed of varying polymers. Therefore, future research can examine how AI can identify the proportions of various polymers in an object and classify it based on those proportions. A highly impactful area for future investigation is the question: "To what extent does combining hyperspectral chemical data matrices with Convolutional Neural Network (CNN) spatial embeddings improve the classification accuracy of transparent polymers within real-time Support Vector Machine (SVM) pipelines?" This multifaceted methodology is anticipated to eliminate the persistent false-positive scatter. Thereby elevating the industrial standard for accuracy closer to true chemical purity.

REFERENCES

[1] AI Stack Exchange. (n.d.). *How the sigmoid function helps us in reducing error in neural networks*. https://ai.stackexchange.com/questions/10378/how-sigmoid-funtion-helps-us-in-reducing-error-in-neural-networks

[2] Alimbekova, N., Hashim, S., Zhumadillayeva, A., & Aiymbay, S. (2024). Deep neural networks as a tool for enhancing the efficiency of plastic waste sorting. *Herald of the Kazakh-British Technical University, 21*(3), 116–127. https://vestnik.kbtu.edu.kz/jour/article/view/1373

[3] Alliance to End Plastic Waste. (2021, March 22). *The plastic waste problem explained*. https://www.endplasticwaste.org/insights/story/the-plastic-waste-problem-explained

[4] Anchieta-Silva, F., Cartaxo, A. de S., Esmeraldo, A. D. de S. R., Senra, E. M., & Pinto, J. C. (2026, March 11). *Advancements in plastic waste sorting: A review of techniques and applications* [Preprint]. Preprints.org. https://doi.org/10.20944/preprints202603.0915.v1

[5] Anis Trend. (n.d.). *Industrial waste sorting plant: The ultimate setup guide*. https://www.anis-trend.com/industrial-waste-sorting-plant/

[6] Astuti, A., Costa, A., Prakoso, A. T., Yani, I., & Resti, Y. (2023). Prediction of plastic-type for sorting system using decision tree model. *Indonesian Journal of Engineering and Science, 4*(1), 75–81. https://doi.org/10.51630/ijes.v4i1.86

[7] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

[8] Bobulski, J., & Kubanek, M. (2021). Deep learning for plastic waste classification system. *Applied Computational Intelligence and Soft Computing, 2021*, Article 6626948. https://doi.org/10.1155/2021/6626948

[9] Cambridge Companies. (2025, May 27). *Key considerations for materials recovery facility design*. https://cambridgecompaniesinc.com/blog/key-considerations-for-materials-recovery-facility-design/

[10] Chandra Asri. (2025, October 15). *Polystyrene: Uses, properties, types, and differences to PP*. https://chandra-asri.com/en/blog/what-is-polystyrene

[11] Hardman, L. (2023, March 8). *The plastic waste problem and its solutions*. Plastic Collective. https://www.plasticcollective.co/the-plastic-waste-problem-and-its-solutions/

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition*. arXiv. https://arxiv.org/abs/1512.03385

[13] International Union for Conservation of Nature. (2021). *Marine plastic pollution* [Issues brief]. https://iucn.org/resources/issues-brief/plastic-pollution

[14] Islam, M., Chen, G., & Jin, S. (2019). An overview of neural network. *American Journal of Neural Networks and Applications, 5*(1), 7–11. https://doi.org/10.11648/j.ajnna.20190501.12

[15] Islam, M. M., Hasan, S. M. M., Hossain, M. R., Uddin, M. P., & Mamun, M. A. (2025). Towards sustainable solutions: Effective waste classification framework via enhanced deep convolutional neural networks. *PLOS ONE, 20*(6), Article e0324294. https://doi.org/10.1371/journal.pone.0324294

[16] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning*. Springer.

[17] Jijo, B. T., & Abdulazeez, A. M. (2021). Classification based on support vector machine: A review. *Journal of Applied Science and Technology Trends, 2*(1). https://jastt.org/index.php/jasttpath/article/view/65

[18] Kibria, M. G., Masuk, N. I., Safayet, R., Nguyen, H. Q., & Mourshed, M. (2023). Plastic waste: Challenges and opportunities to mitigate pollution and effective management. *International Journal of Environmental Research, 17*(1), Article 20. https://doi.org/10.1007/s41742-023-00507-z

[19] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 25). Curran Associates.

[20] Kushaagra Innovations Foundation. (2024). *Material recovery facilities (MRFs) in India: Harnessing ESR principles for long-term success*. https://kushaagra.org/material-recovery-facilities-in-india/

[21] LaValley, M. P. (2008). Logistic regression. *Circulation, 117*(18), 2395–2399. https://doi.org/10.1161/CIRCULATIONAHA.106.682658

[22] Le, A. B., Shkembi, A., Tadee, A., Sturgis, A. C., Gibbs, S. G., & Neitzel, R. L. (2023). Characterization of perceived biohazard exposures, personal protective equipment, and training resources among a sample of formal U.S. solid waste workers: A pilot study. *Journal of Occupational and Environmental Hygiene, 20*(3–4), 129–135. https://doi.org/10.1080/15459624.2023.2179060

[23] Lubongo, C., Bin Daej, M. A. A., & Alexandridis, P. (2024). Recent developments in technology for sorting plastic for recycling: The emergence of artificial intelligence and the rise of the robots. *Recycling, 9*(4), Article 59. https://doi.org/10.3390/recycling9040059

[24] Minu, M. S., Kasa, V. P., Ketharaman, H., Mandepudi, V., Nirmal, K., & Krishnaa, S. (2025). Waste classification using machine learning models: A comparative study. In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies* (pp. 308–315). SciTePress. https://www.scitepress.org/Papers/2025/139275/139275.pdf

[25] National Caucus of Environmental Legislators. (2025). *First in science: The economic impacts of plastic pollution*. https://www.ncelenviro.org/articles/first-in-science-the-economic-impacts-of-plastic-pollution/

[26] Olowolayemo, A., Radzi, N. I. A., & Ismail, N. F. (2022). Classifying plastic waste using deep convolutional neural networks for efficient plastic waste management. *International Journal on Perceptive and Cognitive Computing, 8*(2). https://journals.iium.edu.my/kict/index.php/IJPCC/article/view/282

[27] Plastics For Change. (2025). *Addressing plastics recycling industry worker safety risks*. https://www.plasticsforchange.org/blog/safety-risks-faced-by-workers-in-plastic-recycling-industries

[28] Recycling Today. (2025). *AI in recycling: How smart tech is revolutionizing waste management*. https://www.recyclingtoday.org/blogs/news/ai-in-recycling-how-smart-tech-is-revolutionizing-waste-management

[29] Ruder, S. (2016). *An overview of gradient descent optimization algorithms*. arXiv. https://arxiv.org/abs/1609.04747

[30] Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

[31] Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.

[32] Shetfale, S., Changela, K., Dhondutatya, K., & Ghalme, A. (2025). Harnessing artificial intelligence: Transforming plastic waste management through machine learning. *IJSART, NSSRMGW–2025*, 53. https://ijsart.com/public/storage/paper/pdf/IJSARTV11I4103373.pdf

[33] Sundaralingam, S., & Ramanathan, A. (2024). Recyclable plastic waste segregation with deep learning based hand-eye coordination. *Environmental Research Communications, 6*(4), Article 045007. https://doi.org/10.1088/2515-7620/ad3db7

[34] Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2004). *Support vector machines: A nonlinear modelling and control perspective*. Springer. https://doi.org/10.1007/10984697_1

[35] Tan, M., & Le, Q. V. (2019). *EfficientNet: Rethinking model scaling for convolutional neural networks*. arXiv. https://arxiv.org/abs/1905.11946

[36] Vodovotz, Y. (2022, June 3). *Why isn't plastic biodegradable?* Ohio State News. https://news.osu.edu/why-isnt-plastic-biodegradable/

[37] World Economic Forum. (2026, January 8). *AI-driven circular economy key to managing global resources*. https://www.weforum.org/stories/2026/01/what-is-ai-driven-circular-economy/