**Proceedings of the ASME 2021**
**International Design Engineering Technical Conferences and**
**Computers and Information in Engineering Conference**
**IDETC/CIE2021**
**August 17-20, 2021, Virtual, Online**

**DETC2021-71403**

# DEEP LEARNING AND MACHINE LEARNING TECHNIQUES TO CLASSIFY ELECTRICAL AND ELECTRONIC EQUIPMENT

**Shuaizhou Hu**
Graduate Research Assistant
Environmental Engineering Sciences
University of Florida, Gainesville, FL, 32611
s.hu@ufl.edu

**Xinyao Zhang**
Graduate Research Assistant
Environmental Engineering Sciences
University of Florida, Gainesville, FL, 32611
xinyaozhang@ufl.edu

**Hao-yu Liao**
Graduate Research Assistant
Environmental Engineering Sciences
University of Florida, Gainesville, FL, 32611
haoyuliao@ufl.edu

**Xiao Liang**
Assistant Professor
Civil, Structural and Environmental Engineering
University at Buffalo, Buffalo, NY, 14260
liangx@buffalo.edu

**Minghui Zheng**
Assistant Professor
Mechanical and Aerospace Engineering
University at Buffalo, Buffalo, NY, 14260
mhzheng@buffalo.edu

**Sara Behdad***
Associate Professor
Environmental Engineering Sciences
University of Florida, Gainesville, FL, 32611
sarabehdad@ufl.edu

## ABSTRACT

Remanufacturing sites often receive products with different brands, models, conditions, and quality levels. Proper sorting and classification of the waste stream is a primary step in efficiently recovering and handling used products. The correct classification is particularly crucial in future electronic waste (e-waste) management sites equipped with Artificial Intelligence (AI) and robotic technologies. Robots should be enabled with proper algorithms to recognize and classify products with different features and prepare them for assembly and disassembly tasks. In this study, two categories of Machine Learning (ML) and Deep Learning (DL) techniques are used to classify consumer electronics. ML models include Naïve Bayes with Bernoulli, Gaussian, Multinomial distributions, and Support Vector Machine (SVM) algorithms with four kernels of Linear, Radial Basis Function (RBF), Polynomial, and Sigmoid. While DL models include VGG-16, GoogLeNet, Inception-v3, Inception-v4, and ResNet-50. The above-mentioned models are used to classify three laptop brands, including Apple, HP, and ThinkPad. First the Edge Histogram Descriptor (EHD) and Scale Invariant Feature Transform (SIFT) are used to extract features as inputs to ML models for classification. DL models use laptop images without pre-processing on feature extraction. The trained models are slightly overfitting due to the limited dataset and complexity of model parameters. Despite slight overfitting, the models can identify each brand. The findings prove that DL models outperform them of ML. Among DL models, GoogLeNet has the highest performance in identifying the laptop brands.

*Keywords:* Machine learning, Deep learning, Laptop brand, Classification, Consumer electronic

## NOMENCLATURE

| | |
|---|---|
| $P(C_k\|X)$ | Posterior in Bayes' theorem |
| $P(X\|C_k)$ | Likelihood in Bayes' theorem |
| $P(C_k)$ | Prior in Bayes' theorem |
| $P(X)$ | Evidence in Bayes' theorem |
| $p_{k_i}^{x_i}$ | Probability when event $k_i$ occurs for sample $x_i$. |
| $u_k$ | The mean of class $k$ |
| $\sigma_k^2$ | The variance of class $k$ |
| Accuracy | Accuracy of classifiers |

## 1. INTRODUCTION

Waste Electrical and Electronic Equipment (WEEE) is a complex mixture of materials and components, such as computers, televisions, fridges, and mobile phones. It is currently considered the fastest-growing waste stream globally, with an estimated growth rate of 3% to 5% per year [1]. In 2005, the U.S. discarded 1.36–1.72 million metric tons of WEEE into landfills, and only 0.31–0.34 million metric tons were recycled [2]. Currently, it is estimated that about 80% of WEEE globally is not documented or recycled, meaning its value is lost, and its management is likely to be rudimentary. Consequently, it is necessary to contribute to the efficient use of resources, reduce the amount of e-waste entering the landfill, and encourage the reuse and recycling of consumer electronics.

The benefit of WEEE recovery is not limited to environmental consequences. The efficient recovery of used electronics avoids economic loss and makes the remanufacturing industry economically viable. Currently, the efficiency of remanufacturing operations is affected by many factors ranging from the uncertainty in incoming products to the lack of sufficient technology supporting the yield of remanufacturing operations. Robotics and artificial intelligence techniques are not sufficiently integrated into remanufacturing operations. To address this gap, we aim to show the application of AI techniques in helping remanufacturers sort different brands of certain consumer electronics. The AI algorithms are essential for equipping robots with image recognition methods to properly sort used electronics and ultimately manage the complexity and yield of recovery operations. The focus of this study is on sorting different brands of laptops.

Laptops are selected as the case study for several reasons. The high volume of laptops available for recovery and the potential market for refurbished laptops make it a good case study. In the US, around 14-20 million personal laptops are discarded every year [3], 75% of which can be recovered [4]. Furthermore, 55% of laptops can be reused or harvested to retrieve specific components. Laptops often fail due to the failure of some critical components such as hard disk drive and battery, so they can be repaired and retested after removing failed parts [5]. In addition, the toxic materials in e-waste are dangerous for human health. A multi-elemental analysis in a facility dismantling lead batteries showed the presence of heavy metals in the air samples [6]. Besides, the recovery of rare earth elements in end-of-use laptops and hard disk drives is of utmost importance for the consumer electronic industry [7]. In addition to selecting laptops as a representative of electrical and electronic equipment, brands of laptops are the specific subject of this study. E-waste remanufacturers often collect and recycle the same type of equipment together but ignore recycling cost variations among brands [8] due to the enormously broad scope of equipment brands. Different brands may require different sets of remanufacturing operations which makes brand recycling to be expensive. Hence, developing accurate models for the separation of used electronics based on different features such as shape, color, brand, and geometry helps remanufacturers efficiently manage e-waste received in their facility. In this paper, we would like to highlight the importance of ML and DL techniques in the efficiency of brands sorting processes.

ML and DL have been widely used in different applications. In the current information age, digital images carry important information for storing, describing, and sharing. Storing information in images is a cost-effective method with indispensable use in industry. Besides, image processing is critical in retrieving information stored in image databases. For large databases, it is essential to find useful information in a timely manner [9]. Therefore, the need for image recognition and classification methods has been growing rapidly. ML and DL have developed many image classification algorithms with excellent performance. ML uses statistical methods to enable image classification. Image recognition in ML is a supervised learning approach, where the model learns from a provided dataset and uses the obtained knowledge to perform classification on unseen data [10]. Besides ML techniques, recent developments in DL architectures for computer vision have drastically improved the ability to detect and recognize objects on images [11][12]. Nowakowski and Pamuła [11] have already applied DL for e-waste classification. They used a region-based convolutional neural network (R-CNN) to recognize the category and size of used equipment. However, the current study's focus is not on product type but on detecting product features such as brand.

This study applies several ML and DL techniques to classify different brands of laptops. The primary purpose is to use classification algorithms to enhance the sorting process accuracy in remanufacturing sites.

## 2. DATASET AND EXPERIMENTAL DESIGN

Three popular brands of laptops, including Apple, HP, and ThinkPad, are selected for classification. The high popularity and sales volume of these brands highlights their world-wide usage as well as their recycling needs. Laptops of these three brands have multiple differences both internally and externally, such as the location of the brand logo, attractive colors, keyboard design, and so on. Different brands require different disassembly and remanufacturing operations. Accurate sorting methods can avoid the time and labor cost of recycling.

The dataset used in this study contains a total of 210 images with the size of 200x200 pixels. Each brand has 70 images. The images are collected from Google Images, as shown in Figure 1. Different angles, sizes, and views of images are considered when training ML and DL models. The proposed procedure is shown in Figure 2. For ML, we applied cross-validation with k-fold of 10. And the dataset is randomly split into the training and test sets for DL, where the training-set contains 80% of the data and the test-set contains 20%. In order to acquire more training data, the data augmentation is applied by random resizing, translating rotating, and flipping the original dataset. The best hyperparameters are trained by computing accuracy. For the ML part, the focus is on Naïve Bayes and SVM models, and for the DL part, five different architectures, including VGG-16, GoogLeNet, Inception-v3, ResNet-50, and Inception-v4, are used.

**FIGURE 1:** DIFFERENT ANGLES, SIZES, AND VIEWS OF PICTURES WILL BE USED IN MACHINE LEARNING MODELS.

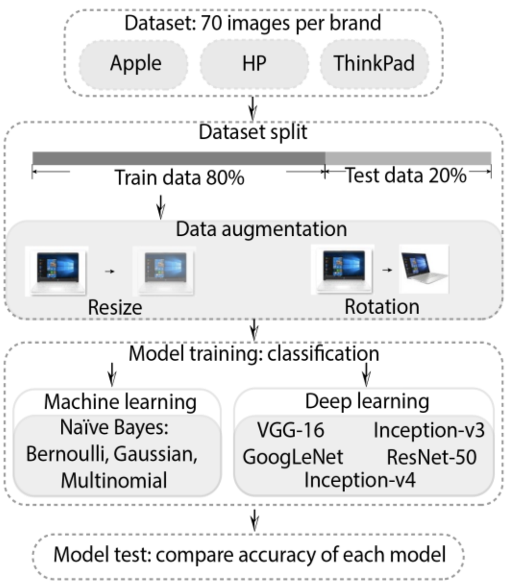



**FIGURE 2:** THE PROPOSED STRUCTURE OF THIS STUDY.

## 3. METHOD

The Edge Histogram Descriptor (EHD) and Scale Invariant Feature Transform (SIFT) are applied to extract image features before training each ML model. The DL architectures are trained by the input images directly. Each method is briefly discussed in the following subsections.

### 3.1 Machine learning

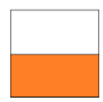
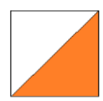
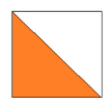
#### 3.1.1 Edge Histogram Descriptor

The EHD is useful tool to extract features for image processing. EHD has scale-invariant benefits by normalizing the extracted features [13] and is a useful tool for retrieving images [14], [15]. EHD has been widely used in the literature. Agarwal et al. (2013) used EHD to retrieve image features to study Content-Based Image Retrieval [16]. Ali et al. (2020) also applied EHD to study Content-Based Image Retrieval for four types of images of car accidents, fire, abnormal objects, and digs [17]. When using EHD, feature types such as vertical, horizontal, degrees, and non-orientation are considered. The output will be a 1x5 sized filter with five matrixes corresponding to each type of feature as shown in Table 1. After extracting features from each image, ML models will be trained using these features. But with the same orientation and size of different laptops, the output of EHD will be identical, which can be addressed by increasing the size of the dataset.

#### 3.1.2 Scale Invariant Feature Transform

SIFT is an image descriptor and feature detection algorithm in computer vision that is invariance to image scale and rotation [18][19]. Five steps are involved in SIFT including scale-space peak selection for locating features, keypoint localization, orientation assignment, keypoint descriptor to describe key points as a high dimensional vector, and keypoint matching. Locality, distinctiveness, quantity, efficiency, and extensibility are advantages of SIFT [20] which makes it a suitable descriptor for object categorization [21], texture classification [22], and image alignment tasks [23].

**TABLE 1:** THE REPRESENTATION OF EACH EDGE TYPE OF MASK [34].

| Edge Type | Representation | Operator Mask |
|---|---|---|
| Vertical Edge |  | $\begin{bmatrix}1 & -1\\1 & -1\end{bmatrix}$ |
| Horizontal Edge |  | $\begin{bmatrix}1 & 1\\-1 & -1\end{bmatrix}$ |
| Diagonal 45 degrees |  | $\begin{bmatrix}\sqrt{2} & 0\\0 & \sqrt{2}\end{bmatrix}$ |
| Diagonal 135 degrees |  | $\begin{bmatrix}0 & \sqrt{2}\\-\sqrt{2} & 0\end{bmatrix}$ |
| Non-Orientation Type Edge |  | $\begin{bmatrix}2 & -2\\-2 & 2\end{bmatrix}$ |

#### 3.1.3 Naïve Bayes

For the ML algorithms, the Naïve Bayes (NB) is used for classification. The NB method considers the probability distribution of the training dataset based on Bayes assumption. The conditional probability can be expressed as:

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)} \tag{1}$$

where $C_k$ denotes class $k$, $X$ is the input dataset, the $P(C_k|X)$ is the posterior, $P(X|C_k)$ is the likelihood multiplied by $P(C_k)$ as the prior, which equals the number of class $k$ divided by the total number of samples, and $P(X)$ is the evidence. After considering different probability distributions for $P(X|C_k)$ such as Bernoulli, Gaussian, and Multinomial, the NB classifiers can be used [24], [25].

### 3.1.4 Support-Vector Machine

SVM [26] is a supervised learning model with considerable accuracy. This algorithm can be used for classification [27], regression [28], and outlier detection [29] tasks. SVM algorithm performs a non-probabilistic binary linear classifier or adopts kernel trick [30][31] to efficiently perform non-linear classification. In this paper, we adopted SVM with Linear, Radial Basis Function (RBF), Polynomial, and Sigmoid kernels and renamed them as SVM-Linear, SVM-RBF, SVM-Poly, and SVM-Sigmoid respectively.

## 3.2 Deep learning

For the DL part, our focus is on Convolutional Neural Network (CNN). CNN is a class of deep neural networks with efficient learning algorithms which have shown exemplary performance in image segmentation, classification, detection, recognition, retrieval, and analysis [35]–[38]. CNN has received attention not only from academia but from the industry as well, where high tech companies such as Google, Microsoft, AT&T, NEC, and Facebook have developed active research groups for exploring CNN's new architectures [35]. There are different well-developed architectures in CNNs like AlexNet, VGG, Inception, ResNet, and Xception. In this paper, we adopted five different architectures to classify different brands of laptops. We used MATLAB software to train GoogLeNet. The MATLAB provides a deep learning toolbox to create new architectures or apply transfer learning, which adopts existing pre-trained models that have been used for other tasks to accelerate the development process for a new task [39]. Also, we applied TensorFlow and Keras on other network architectures such as VGG-16, ResNet-50, etc.

### 3.2.1 VGG-16

Increasing the size of CNNs may enhance its performance [40]. the Visual Geometry Group (VGG) team invented VGG-16 in 2014, consisting of 138 million parameters [41]. With 13 convolutional layers and 3 fully-connected layers, VGG-16 is deeper than AlexNet, but has a smaller filter size of 2×2 and 3×3 and the same ReLU activation function [42]. The weight configuration of the VGG-16 is publicly available. The model adopted in this paper is the pretrained model used by the VGG team in the ILSVRC-2014 competition with a further update, such as reducing the number of parameters by applying all convolutional layers with small 3×3 filters. It is trained with scale jittering [41]. Figure 3 shows VGG-16tarchitecture.

### 3.2.2 GoogLeNet

GoogLeNet is another architecture used in this paper as shown in Figure 4. It was first proposed by Szegedy et al. in 2014 [40]. It is the first version of Inception, namely Inception-v1. Many researchers used GoogLeNet to classify images in different applications. Singla et al. (2016) applied GoogLeNet to identify food images using transfer learning [43]. Lee et al. (2018) used it to improve the performance of recognition on Korean characters [44]. Jahandad et al. (2019) created an offline signature verification system by using GoogLeNet [45]. The GoogLeNet was trained by over a million images with 1000 different objects and has 22 layers (27 layers considering pooling layers. GoogleNet uses the global average pooling at the last inception module.

### 3.2.3. Inception-v3

Inception-v2 [46] and Inception-v3 [47] were put forward as successors to Inception-v1 in 2015. For Inception-v2, the authors recorded some successful tweaks, such as changes to the optimizer, loss function, and adding batch normalization to the auxiliary layers in the auxiliary network [48][46] through extensive experiments. Inception-v2 is not commonly used since it is a prototype of Inception-v3 [48]. We adopted Inception-v3 in this paper. Inception-v3 is the runner-up for image classification in ILSVRC 2015 with 24 million parameters and 48 layers deep [47]. The cost of computation is only 2.5 higher than that of GoogLeNet, and it is more efficient than VGGnet with similar complexity [40], [41], [47], [49]. We utilized the pre-trained model with transfer learning to retrain and test the laptop dataset. Transfer learning is a way to improve learning architectures to transfer the knowledge which has already been learned from a related task [39].

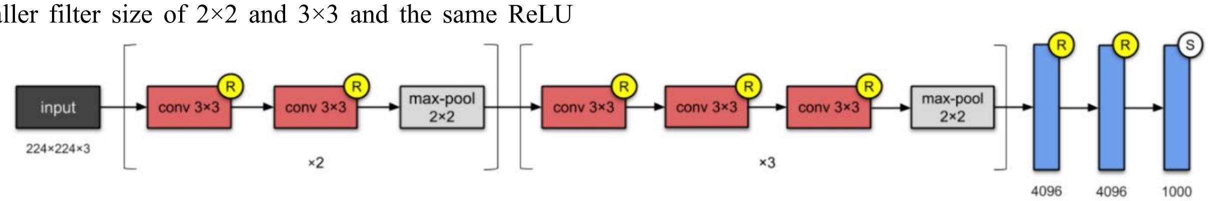



**FIGURE 3:** ARCHITECTURE OF VGG-16 OBTAINED FROM REF [48] BASED ON REF [41].

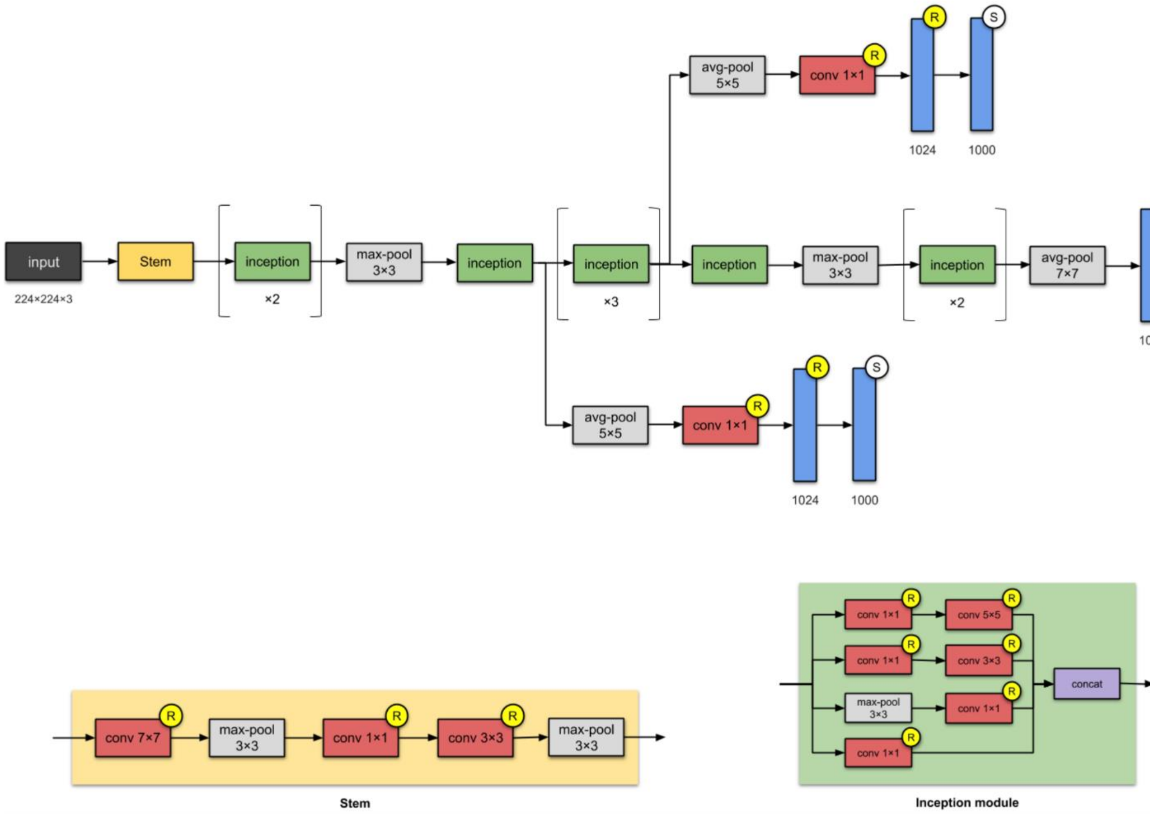



**FIGURE 4:** ARCHITECTURE OF GOOGLENET (INCEPTION-V1) OBTAINED FROM REF [48] BASED ON REF [40].

#### 3.2.4. ResNet-50

Stacking layers and making deeper networks will cause the accuracy to get saturated and degrade rapidly [50]. To address this issue, Microsoft Research launched Residual Neural Network (ResNet) in 2015, as a novel architecture with "shortcut connections" (or skip connections, residuals) [50] that skips one or more layers and has the heavy batch normalization function [46] in building models deeper.ResNet is a deeper trained Neural Network while maintaining lower complexity compared to VGGnet [51]. The original models (ResNet-50, ResNet-101, and ResNet-152) were used in ILSVRC and COCO 2015 competitions, which won the 1st places in: ImageNet classification, ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation [50]. We retrained ResNet-50, which has 50 layers and 26 million parameters.

#### 3.2.5. Inception-v4

Inception-v4 is an improvement from Inception-v3 with 43 million parameters. The inception-v4 architecture is similar to Inception-v3 with a different stem module modified by the Google team. For each grid size, the Google team made uniform choices for Inception blocks [52]. Moreover, a Reduction Blocks was applied to change the dimension of the gird, which has not been used in the earlier versions [40], [47], [52]. There are many applications of Inception-v4. Too et al. (2018) used it to detect plant diseases [53]. Cogan et al. (2019) also applied it on automatic detection of anatomical landmarks and diseases [54]. We applied transfer learning to this model to conduct the classification task.

#### 3.2.6. Comparison of each architecture

VGG-16 with 16 layers and 138 million parameters was invented in 2014 [41]. GooLeNet (Inception-v1) is 22-layer deep with 5 million parameters [40]. As a successor of GooLeNet, Inception-v3 is deeper with 48 layers and 24 million parameters [47]. ResNet-50 has 50 layers and 26 million parameters [50]. Finally, Inception-v4 is a modified version of Inception-v3 with 164-layers and 43 million parameters [52]. Table 2 shows the comparison of them.

**TABLE 2:** COMPARISON OF DEEP LEARNING MODELS.

| Deep Learning Models | Year | Developer | Number of Layers | Number of Parameters |
|---|---|---|---|---|
| VGG-16 | 2014 | VGG | 16 | 138 million |
| GooLeNet | 2014 | Google | 22 | 5 million |
| Inception-v3 | 2015 | Google | 48 | 24 million |
| ResNet-50 | 2015 | Microsoft | 50 | 26 million |
| Inception-v4 | 2016 | Google | 75 | 43 million |

## 4. IDENTIFY LAPTOP BRANDS BY MACHINE LEARNING AND DEEP LEARNING

### 4.1 Performance of deep learning architectures

We trained the five architectures, including VGG-16, GoogLeNet, Inception-v3, Inception-v4, and ResNet-50, and analyzed their performance. As shown in Table 3, GoogLeNet is outperforming other models. Figures 5 and 6 show the normalized confusion matrix of VGG-16 and the GoogLeNet. Both VGG-16 and GoogLeNet have acceptable performance. The VGG-16 can identify Apple, HP, and ThinkPad with 77% accuracy. For example, 100 unlabeled Apple images are fed to VGG-16 model, 77 images were classified correctly. GoogLeNet can identify ThinkPad with 100% correctly if the unlabeled data are full photos. Because we train the models with full images, if the unlabeled data are a partial part of images like the corner part of the keyboard, the performance will be decreased. Both VGG-16 and GoogLeNet have acceptable performance. Inception-v3 is an advanced version of GoogLeNet, but Inception-v3 does not perform better than GoogLeNet in this example. The reason is that Inception-v3 is more complicated than GoogLeNet, and moreover, the training dataset (210 images) is limited. Inception-v3 has more parameters, so it overfits easily. However, its performance is better than VGG-16. Also, as models become more complex such as Inception-v3 with 48 layers, Inception-v4 with 75 layers, and ResNet-50 with 50 layers, the test accuracy is decreased due to overfitting.

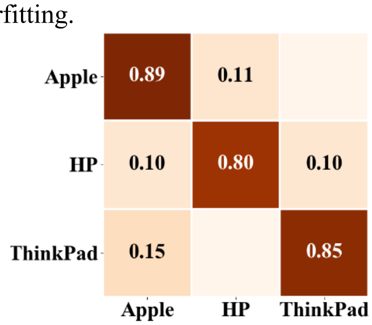



**FIGURE 5:** THE TESTING RESULTS WITH NORMALIZED CONFUSION MATRIX OF VGG-16.

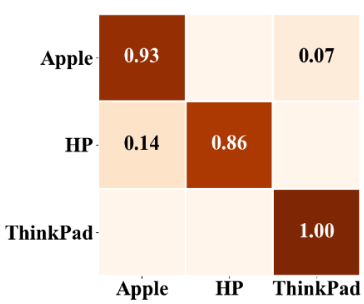



**FIGURE 6:** THE TESTING RESULTS WITH NORMALIZED CONFUSION MATRIX OF GOOGLENET.

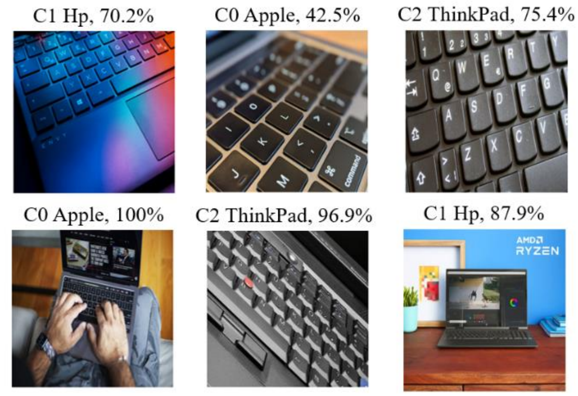



**FIGURE 7:** THE GOOGLENET IDENTIFICATION RESULTS OF SIX ADDITIONAL PARTIAL IMAGES BEYOND 210 FULLY IMAGES DATASET.

Although all models are trained by full images (80% of the dataset), GoogLeNet can identify partial images, as shown in Figure 7. Six additional images are collected, and GoogLeNet recognizes each keyboard brand. Even with other information in the images, such as human hands and messy backgrounds, GoogLeNet can identify them correctly.

### 4.2 Comparison of machine learning and deep learning

Table 3 shows the training and testing results for each algorithm. Among ML models, the best performing model is SVM-Poly (SIFT) with 0.900 training and 0.696 testing accuracy. Among DL models, GoogLeNet is outperforming. The testing accuracy can reach up to 0.929, more efficient than ML. The ML models require extracted features for training, but the DL models can learn directly from training data. Also, during training, DL models will extract features repeatedly by adjusting different convolution layers.

It seems that ML only considers one type of feature by one-time feature extraction, but DL computes various features by using different types of convolution layers and pooling layers.

However, the trained models are slightly overfitting due to the limited dataset (210 images of three laptop brands) and the large numbers of parameters in each model. Although we tried to adjust different parameters to avoid overfitting, the intrinsic parameters of each model still learn to overfit. However, we still achieve acceptable accuracy in identifying each laptop brand.

Although the accuracy of ML is not approximate as DL models, ML models can still identify characteristics of each brand. ML models are fast and easy to train using large datasets in a timely manner compared to DL. Therefore, ML models are still necessary under time-limited conditions.

**TABLE 3:** THE TRAINING AND TESTING RESULTS OF EACH MODEL.

| Model | Training Accuracy | Testing Accuracy |
|---|---|---|
| **Machine learning** | | |
| NB-Bernoulli (EHD) | 0.457 | 0.441 |
| NB-Gaussian (EHD) | 0.607 | 0.595 |
| NB-Multinomial (EHD) | 0.491 | 0.477 |
| SVM-Linear (EHD) | 0.306 | 0.345 |
| SVM-RBF (EHD) | 0.599 | 0.572 |
| SVM-Poly (EHD) | 0.585 | 0.583 |
| SVM-Sigmoid (EHD) | 0.505 | 0.488 |
| NB-Bernoulli (SIFT) | 0.542 | 0.453 |
| NB-Gaussian (SIFT) | 0.594 | 0.530 |
| NB-Multinomial (SIFT) | 0.590 | 0.554 |
| SVM-Linear (SIFT) | 1 | 0.638 |
| SVM-RBF (SIFT) | 0.886 | 0.673 |
| SVM-Poly (SIFT) | 0.900 | 0.696 |
| SVM-Sigmoid (SIFT) | 0.586 | 0.572 |
| **Deep learning** | | |
| VGG-16 | 0.793 | 0.776 |
| GoogLeNet | 0.983 | 0.929 |
| Inception-v3 | 0.868 | 0.845 |
| ResNet-50 | 0.620 | 0.602 |
| Inception-v4 | 0.589 | 0.574 |

## 5. CONCLUSION

In this study, several ML and DL techniques are applied to classify images of different laptop brands. ML algorithms include EHD and SIFT for feature extraction and NB and SVM for classification. DL models include VGG-16, GoogLeNet, Inception-v3 and Inception-v4, and ResNet-50. Among ML models, SVM-Poly (SIFT) has the best performance. Among DL algorithms, GoogLeNet performs better than others. Overall, the results proved that DL models outperform ML algorithms. Meanwhile, overfitting occurred slightly in the training process for both SVM-Poly (SIFT) and GoogLeNetdue to the limited dataset and complexity of model parameters. Although the dataset is limited, the DL models can identify each brand with satisfying accuracy. Also, ML models can still be used in the case of large datasets and the need for a fast response.

This study can be extended in several ways. First, the results can be validated by collecting real-world image data from remanufacturing sites rather than using online images. Second, this research's outcomes can be employed in actual robotic applications in assembly and disassembly tasks. Third, other DL architectures can be developed to recognize and sort different types of products such as computers, refrigerators, and printers. Moreover, algorithms can be extended to other feature and component recognition tasks to harvest parts and critical components in used devices.

## ACKNOWLEDGEMENT

This material is based upon work supported by the National Science Foundation–USA under grants # 2026276, 2026533, and 1928595. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

[1] F. Cucchiella, I. D'Adamo, S. C. Lenny Koh, and P. Rosa, "Recycling of WEEEs: An economic assessment of present and future e-waste streams," *Renewable and Sustainable Energy Reviews*, vol. 51. Elsevier Ltd, pp. 263–272, Jun. 2015, doi: 10.1016/j.rser.2015.06.010.

[2] R. Kahhat, J. Kim, M. Xu, B. Allenby, E. Williams, and P. Zhang, "Exploring e-waste management systems in the United States," *Resources, Conservation and Recycling*, vol. 52, no. 7. Elsevier, pp. 955–964, May 2008, doi: 10.1016/j.resconrec.2008.03.002.

[3] L. Q. Dat, D. T. Truc Linh, S. Y. Chou, and V. F. Yu, "Optimizing reverse logistic costs for recycling end-of-life electrical and electronic products," *Expert Syst. Appl.*, vol. 39, no. 7, pp. 6380–6387, Jun. 2012, doi: 10.1016/j.eswa.2011.12.031.

[4] M. T. Islam and N. Huda, "Reshaping WEEE management in Australia: An investigation on the untapped WEEE products," *J. Clean. Prod.*, vol. 250, p. 119496, Mar. 2020, doi: 10.1016/j.jclepro.2019.119496.

[5] D. Coughlan and C. Fitzpatrick, "Trialling the preparation for reuse of consumer ICT WEEE in Ireland," *J. Clean. Prod.*, vol. 256, p. 120512, May 2020, doi: 10.1016/j.jclepro.2020.120512.

[6] A. Yasar, M. Shoukat, N. Anwar, A. B. Tabinda, M. N. Anwar, and A. S. Nizami, "A comparison of waste recycling facilities for their contribution of heavy metals and trace elements in ambient air," *Environ. Sci. Pollut. Res.*, 2021, doi: 10.1007/s11356-020-12101-3.

[7] Z. Hammache, S. Bensaadi, Y. Berbar, N. Audebrand, A. Szymczyk, and M. Amara, "Recovery of rare earth elements from electronic waste by diffusion dialysis," *Sep. Purif. Technol.*, vol. 254, p. 117641, Jan. 2021, doi: 10.1016/j.seppur.2020.117641.

[8] A. Atasu and L. N. Van Wassenhove, "An Operations Perspective on Product Take-Back Legislation for E-Waste: Theory, Practice, and Research Needs," 2011, doi: 10.1111/j.1937-5956.2011.01291.x.

[9] P. Wang, E. Fan, and P. Wang, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning,"

*Pattern Recognit. Lett.*, vol. 141, pp. 61–67, Jan. 2021, doi: 10.1016/j.patrec.2020.07.042.

[10] R. J. Jankovi´c, "Machine Learning Models for Cultural Heritage Image Classification: Comparison Based on Attribute Selection," doi: 10.3390/info11010012.

[11] P. Nowakowski and T. Pamuła, "Application of deep learning object classifier to improve e-waste collection planning," *Waste Manag.*, vol. 109, pp. 1–9, 2020, doi: 10.1016/j.wasman.2020.04.041.

[12] W. Sterkens, D. Diaz-Romero, T. Goedemé, W. Dewulf, and J. R. Peeters, "Detection and recognition of batteries on X-Ray images of waste electrical and electronic equipment using deep learning," *Resour. Conserv. Recycl.*, p. 105246, Nov. 2020, doi: 10.1016/j.resconrec.2020.105246.

[13] C. S. Won, D. K. Park, and S. J. Park, "Efficient use of MPEG-7 edge histogram descriptor," *ETRI J.*, vol. 24, no. 1, pp. 23–30, 2002, doi: 10.4218/etrij.02.0102.0103.

[14] A. K. Jain and A. Vailaya, "Image retrieval using color and shape," *Pattern Recognit.*, vol. 29, no. 8, pp. 1233–1244, 1996, doi: 10.1016/0031-3203(95)00160-3.

[15] M. . Swain and D. . Ballard, "Color indexing," *iNT. j. Comput. Vis.*, vol. 7–1, 1991.

[16] S. Agarwal, A. K. Verma, and P. Singh, "Content based image retrieval using discrete wavelet transform and edge histogram descriptor," in *Proceedings of the 2013 International Conference on Information Systems and Computer Networks, ISCON 2013*, 2013, pp. 19–23, doi: 10.1109/ICISCON.2013.6524166.

[17] S. K. Ali and S. A. Sadoon, "Image Classification based on CBIR," in *Journal of Physics: Conference Series*, 2020, vol. 1591, no. 1, p. 12022.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[19] T. Lindeberg, "Scale invariant feature transform," 2012.

[20] J. Wu, Z. Cui, V. S. Sheng, P. Zhao, D. Su, and S. Gong, "A Comparative Study of SIFT and its Variants," *Meas. Sci. Rev.*, vol. 13, no. 3, p. 122, 2013.

[21] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[22] L. Sifre and S. Mallat, "Rigid-motion scattering for texture classification," *arXiv Prepr. arXiv1403.1687*, 2014.

[23] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, 2010.

[24] Wikipedia contributors, "Naive Bayes classifier." 2021.

[25] K. P. Murphy, "Naive bayes classifiers," *Univ. Br. Columbia*, vol. 18, no. 60, 2006.

[26] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. their Appl.*, vol. 13, no. 4, pp. 18–28, 1998.

[27] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification." Taipei, 2003.

[28] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Adv. Neural Inf. Process. Syst.*, vol. 9, pp. 155–161, 1997.

[29] D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognit. Lett.*, vol. 20, no. 11–13, pp. 1191–1199, 1999.

[30] V. Jakkula, "Tutorial on support vector machine (svm)," *Sch. EECS, Washingt. State Univ.*, vol. 37, 2006.

[31] M. Hofmann, "Support vector machines-kernels and the kernel trick," *Notes*, vol. 26, no. 3, pp. 1–16, 2006.

[32] M. A. Ebrahimi, M.-H. Khoshtaghaza, S. Minaei, and B. Jamshidi, "Vision-based pest detection based on SVM classification method," *Comput. Electron. Agric.*, vol. 137, pp. 52–58, 2017.

[33] E. Alickovic and A. Subasi, "Ensemble SVM method for automatic sleep stage classification," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 6, pp. 1258–1265, 2018.

[34] L. Piras, "Interactive search techniques for content-based retrieval from archives of images," 2021.

[35] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020.

[36] A. Kamilaris and F. X. Prenafeta-Boldú, "A review of the use of convolutional neural networks in agriculture," *J. Agric. Sci.*, vol. 156, no. 3, pp. 312–322, 2018.

[37] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, "Medical image

analysis using convolutional neural networks: a review," *J. Med. Syst.*, vol. 42, no. 11, pp. 1–13, 2018.

[38] M. T. McCann, K. H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging: A review," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 85–95, 2017.

[39] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, IGI global, 2010, pp. 242–264.

[40] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012.

[43] A. Singla, L. Yuan, and T. Ebrahimi, "Food/non-food image classification and food categorization using pre-trained googlenet model," in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, 2016, pp. 3–11.

[44] S.-G. Lee, Y. Sung, Y.-G. Kim, and E.-Y. Cha, "Variations of AlexNet and GoogLeNet to improve Korean character recognition performance," *J. Inf. Process. Syst.*, vol. 14, no. 1, pp. 205–217, 2018.

[45] S. M. Sam, K. Kamardin, N. N. A. Sjarif, and N. Mohamed, "Offline signature verification using deep learning convolutional neural network (CNN) architectures GoogLeNet inception-v1 and inception-v3," *Procedia Comput. Sci.*, vol. 161, pp. 475–483, 2019.

[46] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.

[47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[48] R. Karim, "Illustrated: 10 CNN Architectures," *Medium*, 2020. .

[49] S.-H. Tsang, "Review: Inception-v3 - 1st Runner Up (Image Classification) in ILSVRC 2015," *Medium*, 2019. .

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[51] S. Das, "CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more," *Medium*, 2019. .

[52] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, vol. 31, no. 1.

[53] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, 2019.

[54] T. Cogan, M. Cogan, and L. Tamil, "MAPGI: Accurate identification of anatomical landmarks and diseased tissue in gastrointestinal tract using deep learning," *Comput. Biol. Med.*, vol. 111, p. 103351, 2019.