VOLTAVISION: A TRANSFER LEARNING MODEL FOR ELECTRONIC COMPONENT CLASSIFICATION

Anas Mohammad Ishfaqul Muktadir Osmani, Taimur Rahman & Salekul Islam
Department of Computer Science and Engineering
United International University
{aosmani203004,trahman221427}@bscse.uiu.ac.bd,salekul@cse.uiu.ac.bd

ABSTRACT

In this paper, we analyze the effectiveness of transfer learning on classifying electronic components. Transfer learning reuses pre-trained models to save time and resources in building a robust classifier rather than learning from scratch. Our work introduces a lightweight CNN, coined as VoltaVision, and compares its performance against more complex models. We test the hypothesis that transferring knowledge from a similar task to our target domain yields better results than state-of-the-art models trained on general datasets. Our dataset and code for this work are available at https://github.com/AnasIshfaque/VoltaVision.

1 INTRODUCTION

Traditional transfer learning uses large pre-trained models on general classification tasks to cut down on the time required for training. However, it may be more valuable to use purpose-built transfer models on focused tasks. Focused tasks such as medical imaging, waste classification and facial recognition have become far more relevant over a one-size-fits-all solution. One such problem is detecting electronic components. Most electronic components are minute and intricate in design which means a rich diverse dataset would be needed to capture their defining features. In order to mitigate the effort of creating such a dataset, transfer learning can be used to train a reliable image classifier needing only a few samples of the target class.

The applications of identifying small electronic components would be significant in recycling e-waste, or automating component transactions in a lending machine as prototyped in Appendix A.1. The benefit of this approach would be to reduce model footprint considerably while maintaining accurate inference.

2 RELATED WORKS

Siamese networks, also known as twin neural networks, have been used for few-shot learning to identify electronic components (Cheng et al., 2022). They use an improved VGG-16 (Simonyan & Zisserman, 2014) model and a visual geometry group with five convolutional layers for feature extraction to improve recognition even with smaller samples. In (Salvador et al., 2018), it is shown that deep convolutional neural networks can be used with transfer learning to identify electronic components. Pre-trained models such as Inception-v3 (Szegedy et al., 2016), GoogLeNet (Szegedy et al., 2015) and ResNet-101 (He et al., 2016b) are used for faster deployment. A faster SqueezeNet network is used in Xu et al. (2020) as the basis for developing a simpler classification algorithm having fewer parameters without compromising the performance.

3 METHODOLOGY

Our target task is to classify three electronic components; humidity sensor (DH11), transistor and Bluetooth module. To achieve this with transfer learning, we have fine-tuned various pre-trained models using a small dataset of our target components, described in Appendix A.2. We have created this dataset adopting the methodology used in (Chand, 2023) by capturing images of the components

from various view angles. The pre-trained models are either trained on a general dataset or a specific dataset that belongs to the same domain as the fine-tune data.

We have experimented with some large well-known image classifiers as the pre-trained model which includes the ResNet-18 (He et al., 2016a), VGG-16 (Simonyan & Zisserman, 2014), Inception-v3 (Szegedy et al., 2016), AlexNet (Krizhevsky et al., 2012) and GoogLeNet (Szegedy et al., 2015). These models contain weights trained on ImageNet (Russakovsky et al., 2015). We train our model, VoltaVision with general CIFAR datasets (Krizhevsky et al., 2009), and also task-relevant datasets (Chand, 2023; Acharya, 2019) with varying class sizes to be used as the pre-trained model (shown in Appendix A.3). Figure 1 outlines our experimental setup, and more technical details are covered in Appendix A.4. Since our fine-tuning dataset is relatively small, we perform 5-fold cross validation to ensure that our reported performance values are accurate.

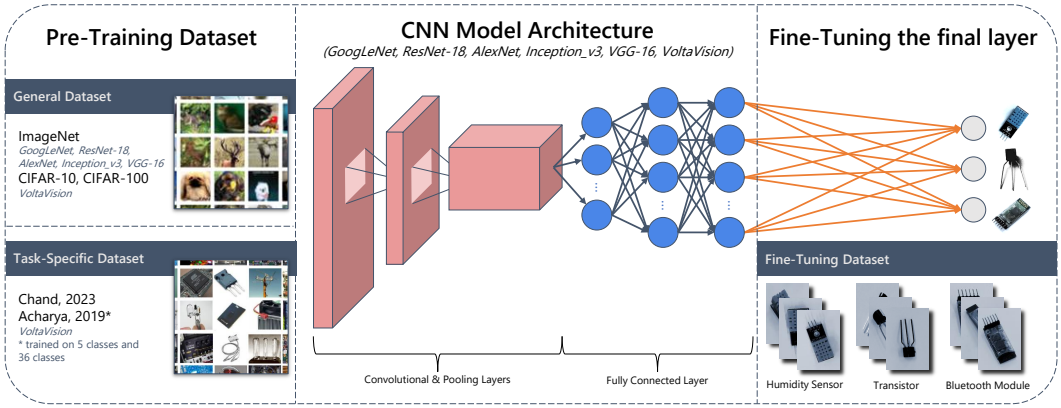

Figure 1: Experimental setups

4 Results and Discussion

Pre-Train Dataset	Model	Accuracy	Precision	Recall	F1-Score	Fine-tuning Time	GPU Usage	Model Size
ImageNet_{1000}	GoogLeNet	78.00	78.65	78.29	77.94	84.1	225.1	49.7Mb
	ResNet-18	80.60	81.20	80.76	80.67	77.9	181.5	44.7Mb
	Inception-v3	82.60	83.31	82.90	82.78	121.1	521.2	104Mb
	AlexNet	96.40	96.38	96.35	96.32	66.9	844.8	233Mb
	VGG-16	**98.80**	**98.78**	**98.85**	**98.81**	117.5	1968	528Mb
CIFAR-10_{10}	VoltaVision	76.40	76.80	76.02	75.51	53.1	31.1	320kb
CIFAR-100_{100}		89.60	89.85	89.67	89.73	48.4	23.6	2.92Mb
Chand (2023)_{3}		95.20	95.21	95.25	95.20	46.7	**18**	**127kb**
$\text{Acharya (2019)}_{5}$		93.60	93.85	93.61	93.67	**45.5**	18.1	185kb
$\text{Acharya (2019)}_{36}$		92.60	92.31	92.64	92.38	46.8	18.9	1.08Mb

Table 1: Average performance metrics for corresponding pre-trained models with different datasets, where subscripts denote the number of classes. Performance metrics are shown in percentages, time is in seconds, GPU usage is in MiB.

Our average results are summarized in Table 1. It is noticeable that our model, VoltaVision (a custom CNN model) shows performance comparable to that of VGG-16 and AlexNet when pre-trained on a task-oriented dataset similar to our target. VoltaVision achieves this while it is smaller in size, requires less computational resources, and is faster in training. Furthermore, it is worth noting that when VoltaVision is pre-trained using general datasets CIFAR-10 and CIFAR-100 the results from a task-specific pre-trained model is still relatively higher. Thus, it shows that a more focused knowledge transfer yields better results in most cases (i.e. excluding complex architectures).

For future work, there is a scope to classify more components, or consider further datasets for pre-training, as well as experiment with other model architectures. It may also be worthwhile to consider finer classifications, such as subcategories within transistors. Finally, it might be interesting to see if fine-tuning layers beyond just the final one yield different results.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

REFERENCES

Sunim Acharya. Electronic components and devices, 2019. URL https://www.kaggle.com/datasets/aryaminus/electronic-components.

Praneel Chand. A low-resolution used electronic parts image dataset for sorting application. *Data*, 8(1):20, 2023.

Yahui Cheng, Aimin Wang, and Long Wu. A classification method for electronic components based on siamese network. *Sensors*, 22(17), 2022. ISSN 1424-8220. doi: 10.3390/s22176478. URL https://www.mdpi.com/1424-8220/22/17/6478.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016b.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Rodolfo C. Salvador, Argel A. Bandala, Irister M. Javel, Rhen Anjerome R. Bedruz, Elmer P. Dadios, and Ryan Rhay P. Vicerra. Deeptronic: An electronic device classification model using deep convolutional neural networks. In *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, pp. 1–5, 2018. doi: 10.1109/HNICEM.2018.8666303.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Yuanyuan Xu, Genke Yang, Jiliang Luo, and Jianan He. n electronic component recognition algorithm based on deep learning with a faster squeezenet. *Mathematical Problems in Engineering*, 2020:1–11, 2020.

A APPENDIX

A.1 ELECTRONIC LENDING MACHINE

Figure 2 shows an overview of the hardware implementation of the prototype we build. There are two major functions of the machine; borrowing and returning components. An authentication mechanism is deployed using an RFID module, where the user needs to scan a registered card on the RFID scanner to get access to the system. For interacting with the machine, a UI module is designed consisting of a 16x2 LCD display and a 3x4 keypad. For automating the component transaction process, several servo motors and stepper motors are used for precise handling of the components. The authentication module, UI module and the actuators (i.e., the motors) are interfaced with an Arduino MEGA microcontroller which is also connected to a Windows laptop. For tracking the lending records, a database server is maintained, which is updated after every transaction.

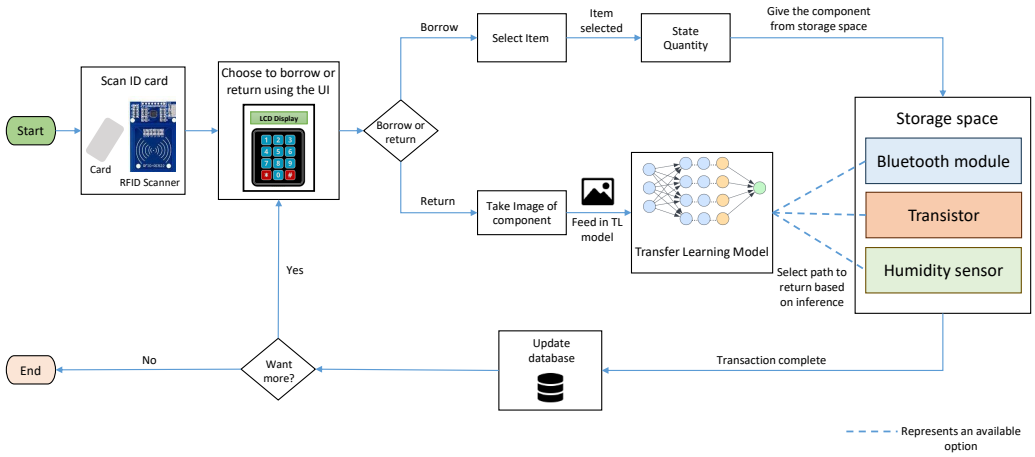

Figure 2: Overview of the electronic lending machine prototype

The transfer learning model is used when returning a component. An image of the component to be returned is captured via a webcam and is fed into the model for inference. In response to this prediction, a sequence of servo motor rotation moves the component from the photo-shooting platform to its designated slot in the storage space.

Figure 3: Lending machine from user view and internal view

A.2 DATASET DESCRIPTION

Classes	Instances
Transistor	110
Humidity Sensor	116
Bluetooth Module	102

(a) Our dataset description

Dataset	Classes	Total Images
ImageNet-1K	1,000	1,281,167
CIFAR-100	100	60,000
CIFAR-10	10	60,000
Chand (2023)	3	1,810
Acharya (2019)	36	10,990

(b) Datasets used for pre-training

Table 2: Description of datasets used in our experiments

For building our fine-tuning dataset, we take pictures of three distinct electronic components from different angles on a white background. There are over 100 images for each of the classes, totalling to 328 images. The dataset is available at our GitHub[1] and it is described in Table 2a. Furthermore, readily available pre-trained models such as AlexNet are trained on ImageNet (Russakovsky et al., 2015). While it is not feasible for us to train VoltaVision on a large dataset such as ImageNet, VoltaVision is pre-configured with weights trained from CIFAR (Krizhevsky et al., 2009), Chand (2023) and Acharya (2019). These pre-training datasets are outlined in Table 2b. Another experiment is carried out with a manually downsized Acharya (2019) dataset to five classes (electrolytic capacitor, electric relay, heat sink, potentiometer and solenoid).

A.3 BUILDING VOLTAVISION BY CUSTOMISING CNN MODEL

The term "Volta" is derived from voltage as it is closely related to every electronic component and "Vision" comes from the fact that our model works with visual data. In terms of the network architecture, there are 3 convolution layers. The first convolution layer gets the image as the input and has 12 filters of size 3, stride 1 and padding 2. Then, batch normalization and ReLU activation are applied. To reduce the dimension of the feature map, a max pooling layer is used next. The second and third convolution layers have 20 and 32 filters of the same size as the first one. Similarly, the outputs of these convolution layers have also gone through batch normalization and ReLU activation. Finally, the output of the last convolutional layer is flattened and fed to a fully connected layer as a vector.

A.4 TRANSFER LEARNING EXPERIMENTS

Model	Parameters	Number of Layers
VGG-16	138 million	16
Inception-v3	24 million	48
AlexNet	62 million	8
ResNet-18	11 million	18
GoogLeNet	7 million	22
VoltaVision	29 thousand	3

Table 3: Complexity of the pre-trained models

For each of the pre-trained models, the last layer is replaced with a new one which matches the number of desired classes (i.e., 3). Then, the model is fine-tuned for 25 epochs to classify the target components: Bluetooth module, humidity sensor and transistor. The fine-tuning dataset contained around 110 images of each class A.2. It is divided into 5 folds to perform 5-fold cross validation where each time 1 fold is selected as the validation set and the rest of the folds form the train set. A step LR scheduler with a step size of 7 and a gamma of 0.1 are used to reduce the learning rate iteratively. As for the optimizer and loss function, SGD and CrossEntropy Loss are used. More complex models like VGG-16 and AlexNet (shown in Table 3), take much longer time and resources (shown in Table 1) than VoltaVision to train and thus, produce relatively better results than VoltaVision.

[1] `https://github.com/AnasIshfaque/VoltaVision`