Leveraging AI in Waste Management:

Deploying Variants of Convolutional Neural Network in Garbage Classification

Report prepared by Team 1

April 24th, 2023

Introduction

With the increasing, undivided attention dedicated to environmental sustainability worldwide, countries in the world have introduced measures such as the *compulsory garbage sorting system* introduced by China in 2019 (Huang, 2019) and the *Single-Use Plastic Prohibition Regulation* introduced by Canada in 2022 (Government of Canada, 2023).

Waste management is an essential process that helps in reducing the negative impact of human activities on the environment. The improper disposal of waste can lead to soil, air, and water pollution, which can have devastating consequences for the ecosystem and human health. Waste management involves the collection, transportation, and disposal of waste in a safe and sustainable manner. It plays a crucial role in conserving natural resources, reducing greenhouse gas emissions, and mitigating climate change.

In recent years, machine learning models, such as Convolutional Neural Networks (CNN), have been increasingly employed to automate waste classification. These models can accurately identify different types of waste materials based on their visual features, which can significantly improve waste management efficiency and reduce human error. Hence, the goal of this study is to use different variants of CNN models for waste classification and their potential impact on waste management and environmental conservation.

In this report, we will first introduce the dataset we used to train the models, followed by a discussion of the different variants of CNN we deployed. Then, we ran different experiments on dataset-model combinations to demonstrate the capability of this topic and conclude the report with a discussion on the limitation and future steps.

Background

While manual waste sorting is more efficient than mechanical sorting in developing countries, it can expose workers to hazardous materials and cause various health problems. Using deep learning for garbage classification is an effective solution that has positive environmental effects and benefits worker health. Recent studies have developed visual waste separation systems based on pre-trained neural networks and deep learning approaches using computer vision to identify the waste type automatically.

In 2016, Yang and Thung released the *TrashNet* dataset achieving an accuracy of 63% using *SVM* and 22% using an adapted version of *AlexNet* CNN (Yang & Thung, 2016). Other researchers have since attempted to evaluate their models using *TrashNet*. For instance, Arda Aral and Recep Keskin applied several CNN models, such as *Xception*, *MobileNet*, *Inception-ResNet-V2*, *DenseNet-169* and *DenseNet-120*, to classify the waste material with data augmentation (Liu & Yuan, 2022). Their experiments showed that the *DenseNet-121* transfer learning model achieved the best accuracy of 95%, followed by the *Inception-ResNet-V2* model with 94% accuracy (Liu & Yuan, 2022). Bircanoğlu developed *RecycleNet*, which optimized deep CNN architecture for selected recyclable objects by altering the skip connection patterns (Bircanoglu, Atay, Beser, & Genc Ozgun, 2018). Although it achieved 81% accuracy for the *TrashNet* dataset, it reduced the parameters from 7 million to about 3 million.

Vo et al. designed a *DNN-TC* framework using the *ResNeXt* model that automatically classifies trash in smart waste sorter machinery (VO, SON, VO, & LE, 2019). They modified the standard *ResNeXt-101* model by adding two fully connected layers, achieving the best accuracy of 94 % on the *TrashNet* dataset (Xie, Girshick, Dollár, Tu, & He, 2017). However, the *DNN-TC* method outperformed *ResNeXt-101* only for glass and cardboard classes, and its performance decreased for metal, paper, and plastic classes without explanation. Shi et al. proposed the *M-b Xception* network improvement method based on channel expansion for trash image classification, achieving the best performance results of 94.34% on the *TrashNet* dataset (Shi, Xia, & Wang, 2020). Although their method improved classification accuracy, it also increased computational cost and parameter capacity.

Our project goal is to replicate prior research findings and assess novel computer vision models that have emerged in recent years. Once we have developed a model, we intend to deploy it on small electronic devices such as smartphones equipped with sensors and cameras to aid in the daily task of garbage classification.

Business Problem

How to leverage modern AI techniques to accurately and efficiently classify trash?

Methods

Data

Primary Data for Training

TrashNet, also known as *Garbage Classification Data*, is used as our primary training dataset, as many of the previous garbage classification tasks are also trained on this dataset. Using this dataset to train the model would allow us to compare our results with others to see how we perform. *TrashNet* contains a total of six categories, with a total of 2,527 images. An overview of this dataset is summarized in *Table 1*.

Table 1: Data summary for the primary training dataset

Category	Number of Observation	Sample Image
Cardboard	403	
Glass	501	
Metal	410	
Paper	594	
Plastic	482	
Trash	137	
Total	2,527	

Supplementary Dataset for Experiment

The *Garbage Classification (12 Classes)* dataset obtained from *Kaggle* was used as a supplementary dataset in this study. The dataset consisted of images of garbage from twelve categories, compiled by the dataset author. To gather most of the images, the author used web scraping, as it was infeasible to collect real garbage images at the time. The remaining images were real garbage snapshots from a dataset put together by other contributors on *Kaggle* (Mohamed, 2021). In order to align the dataset with real-life garbage behavior, we further combined the twelve categories into seven categories. For example, old *clothes* and *shoes* were classified as donation items rather than garbage, and *glass* tinted with different colors was classified as just *glass*. In addition, as the supplementary dataset contains more than seven thousand images, and have images from the *battery* and *biological* category which the trained model cannot predict as they are not present in the training set. Hence, to reduce the runtime and also allowing the trained model to be able to predict on the supplementary dataset, we sampled 20% from the supplementary dataset and removed images from the *battery* and *biological* category.

After these manipulations, we obtained our final experimental dataset. An overview of the dataset is shown in *Appendix A Table 4*.

Models

Our entire workflow went as follows: we first divided the primary dataset into 80% training and 20% testing. When training each of our models on the dataset, we resized the images to match input size required by that specific model, then trained the model with ten epochs and recorded the relevant metrics. We would then select the model giving us the best performance and fit that model on the entire primary dataset, then use this model to predict labels on the supplementary dataset to further measure model performance. Next, we will introduce and explain the various structures of the methods we used.

Pre-trained model

Our project relied heavily on pre-trained neural networks to aid our classification task. Pre-trained models are neural networks that have been trained on a large dataset for a specific task, such as image classification. Using a pre-trained model as a starting point for building a deep neural network can be beneficial, as it can save time and computation compared to starting from scratch. Pre-trained models often achieve state-of-the-art performance and can serve as a strong baseline for our models. Some commonly used pre-trained models for image classification include *ResNet*, *MobileNet*, and *EfficientNet*, all of which we explored and deployed in our project.

In practice, we typically load the model with pre-trained weights and freeze some layers. Freezing the weights of some layers means that they will not be updated during training, which allows us to keep the pre-trained weights intact and use them to extract high-level features efficiently. However, in some cases, we may want to unfreeze some or all of the layers of the pre-trained model during training. This is because as we fine-tune our model on our specific task, we may need to adjust the features extracted from the earlier layers to better fit our data. Unfreezing some layers allows us to adjust the features captured by those layers and make the model more tailored to our task. Typically, we would unfreeze only the last few layers of the pre-trained model.

Next, we will move on to introduce the structure, common use cases, and the benefits and drawbacks of the different models we used in this project.

ResNet (2015)

Residual neural network (ResNet) is a convolutional neural network frequently utilized for computer vision applications. *ResNet* is constructed to accommodate numerous convolutional layers, and it initially stacks multiple identity mappings (convolutional layers that do nothing), skips these layers, and reuses the previous layer's activations. Skipping accelerates initial training by compressing the network into fewer layers. *ResNet* uses residual blocks, which contain two convolutional layers with a skip connection that bypasses the convolutional layers, on the remaining parts of the network to explore more of the feature space of the input image. By stacking multiple residual blocks on top of each other with downsampling layers, *ResNet* can decrease the spatial dimensions of the feature maps (*ResNet: The Basics and 3 ResNet Extensions*, n.d.). The final output of the network is usually a global average pooling layer followed by a fully connected layer. *ResNet* is commonly employed for image classification, object detection, and semantic segmentation.

ResNet's capacity to train very deep networks and attain high accuracy on image classification tasks makes it an ideal pre-trained model. The advantage of using *ResNet* is that networks with a large number of layers can be trained easily without increasing the training error percentage. When used as a pre-trained model for garbage classification, *ResNet* can identify important features and patterns that may be difficult for other algorithms to detect based on its ability to learn hierarchical representations of images.

MobileNet (2017)

MobileNet-V2 is a lightweight and efficient deep neural network architecture. It is an improvement over the first version of *MobileNet* and is designed to be used in mobile and embedded vision applications. The network uses depthwise separable convolutions, which factorize a standard convolution into a depthwise convolution followed by a pointwise convolution, and make it faster and more efficient. *MobileNet-V2* is built on an inverted residual structure with linear bottlenecks between the layers. The inverted residual structure increases the input and output dimensions of the layers, while the linear bottlenecks reduce the number of channels in between, resulting in fewer parameters and computations. While *MobileNet-V2* is more efficient than other models, it may not achieve the same level of accuracy,

especially for complex tasks or large-scale datasets. *MobileNet-V2* can be used for classifying images into various categories, such as identifying objects or animals. In addition, *MobileNet-V2* can be used as a feature extractor in transfer learning, where a pre-trained model is fine-tuned on a specific task with a limited amount of data. In *Appendix B. MobileNet Explanation in Detail*, we have included more technical details regarding this model.

Given the benefits *MobileNet-V2* brings us, the reasons that we chose *MobileNet* as one of the pre-trained models can be summarized in two reasons. Most importantly, *MobileNet* has robust feature extraction. Trash classification requires the model to distinguish between various materials and objects, some of which may have subtle differences. *MobileNet* is built using depthwise separable convolutions, which allow it to learn robust and discriminative features for different object classes. Moreover, *MobileNet* is efficient. Considering we might deploy the classification technique on mobile devices, its efficiency helps to reduce power consumption and ensures smooth operation.

EfficientNet (2019)

EfficientNet is a type of convolutional neural network architecture that was designed to achieve higher accuracy while using significantly fewer parameters and computational resources than existing models. The architecture uses a method called compound scaling to scale the depth, width, and resolution of the network simultaneously with a fixed set of scaling coefficients, resulting in better performance and efficiency (Tan & Le, n.d.). However, compared to other image classification models, *EfficientNets* require larger image resolutions to achieve strong performance. As a convolutional neural network architecture applied to various computer vision tasks, *EfficientNet* is also commonly employed for image classification, object detection, and semantic segmentation.

EfficientNet is also a promising pre-trained model for garbage classification, as its ability to achieve high accuracy with fewer parameters and less computational resources makes it an efficient and effective model for this task. Garbage classification involves identifying and categorizing different types of waste materials based on their appearance, which can be complex and subtle. *EfficientNet* consists of seven models labeled B0 to B7, with B0 being the smallest and least computationally expensive, and B7 being the largest and most computationally expensive. For our task, we utilized *EfficientNet* B7 to achieve higher accuracy.

MobileViT (2021)

Convolutional Neural Networks (CNNs) have been proven to be highly effective in various computer vision tasks, from image classification to object detection. However, CNNs have limitations. One major limitation is their limited ability to *capture global context*. For example, in the context of an image, a CNN may struggle to understand the scene, such as whether it is a beach, forest, or city, and the relationship between objects, such as a person holding a leash connected to a dog. Another limitation is *distortion*, which can come in occlusion, distribution shift, or even adversarial patches (Dosovitskiy et al., n.d.). These challenges can cause the CNN to misidentify or fail to identify objects, leading to decreased accuracy in computer vision tasks. Therefore, while CNNs are powerful tools, they must be aware of their limitations and consider alternative approaches when necessary.

As transformer architectures dominate the *Natural Language Processing (NLP)* field, researchers have explored their application in computer vision. In 2021, Google's Brain team introduced the *Vision Transformer (ViT)* model, incorporating transformer encoding blocks (Dosovitskiy et al., n.d.). ViT's architecture is unique because it minimizes CNN usage, focusing on the potential of a predominantly transformer-based model for computer vision tasks (Dosovitskiy et al., n.d.).

Transformers necessitate vector inputs (1-D arrays), so researchers divide images into patches and convert them into vectors (e.g., 10 x 10 to 100 x 1). These vectors are combined with positional embeddings, ensuring that transformers recognize each patch's position. Transformers rely on self-attention mechanisms to model input-element relationships. It's important to note that self-attention is permutation invariant, treating input elements as unordered sets. Therefore, without the addition of positional encoding, two different images (shown below) could yield the same output when using a transformer.

Figure 1: Original image on the left, and the shifted image on the right

In addition to positional embeddings, researchers introduce a learnable empty token before the first patch, serving as a class token for image classification (Tomar, n.d.). This class token encodes information from other patches through attention layers and is eventually fed into a softmax classifier (MLE) to classify the image. For visualization and detailed architecture of the model please refer to *Appendix B.ViT and Mobile ViT Architecture*.

The developers of ViT found that it outperforms state-of-the-art CNNs when pre-trained on extensive datasets (Dosovitskiy et al., n.d.). However, the model has its limitations. A significant drawback is its large number of parameters, necessitating considerable computing power for training and deployment (Mehta & Rastegari, n.d.). The model also lacks spatial inductive bias, making it less sensitive to spatial information than CNNs, which could hinder performance in computer vision tasks where spatial relationships are crucial. ViT requires more data for pre-training than CNNs and relies heavily on L2 regularization and augmentation techniques, making it sensitive to these factors and susceptible to a significant decrease in accuracy without proper augmentation (Mehta & Rastegari, n.d.). Positional encoding in ViT can hinder its transferability to other tasks due to its fixed-length sequence, limiting the model's flexibility. For example, pre-training the model on smaller image sizes like 224 x 224 and testing it on larger ones like 512 x 512 could result in worse performance (Dosovitskiy et al., n.d.). As such, it is vital to carefully weigh ViT's limitations before employing it in computer vision tasks.

To address these limitations, Apple researchers developed *MobileViT*, a new model that combines the strengths of CNNs (inductive bias and spatial information capture) with those of ViT (global context processing) (Mehta & Rastegari, n.d.). *MobileViT* is akin to *MobileNetV2*, and its design involves modifying the standard convolution process, which includes unfolding, local processing (depth-wise convolution), and folding operations (Mehta & Rastegari, n.d.). The *MobileViT* block replaces local processing in convolutions with global processing through transformers (Mehta & Rastegari, n.d.) (as shown below). This fusion endows *MobileViT* with both CNN- and ViT-like properties, enabling it to learn superior representations using fewer parameters and straightforward training methods (e.g., basic augmentation). Experimental results demonstrate that *MobileViT* outperforms other lightweight CNNs when pre-trained on smaller datasets (Mehta & Rastegari, n.d.). For visualization and detailed architecture of the model please check *Appendix B.ViT and Mobile ViT Architecture*.

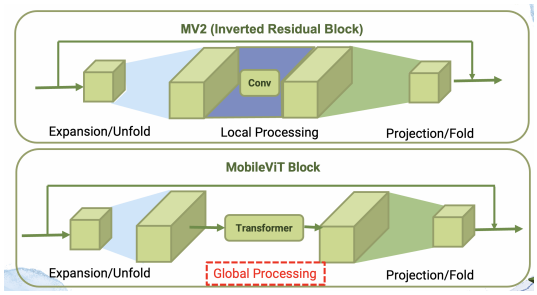


Figure 2: Difference between MV2 and MobileViT Block

Data Augmentation

Despite the many models we tried for our classification task, we further applied a technique called *data augmentation* to add diversity to our existing training data. Data augmentation involves different transformations, such as rotating, flipping, and shifting image positions on the training dataset. The goal is to *augment* the data set by adding synthetic data points for the model to learn more robust and generalizable patterns. In our analysis, we used the augmentation choices summarized in *Table 2*:

Table 2: Transformations we used to augment the training dataset

Transformation	Value	What it does
Rotation	40	Randomly rotate an image by ± 40 degrees
Width shift	0.2	Shift image horizontally by a maximum of 20% of its total width
Height shift	0.2	Shift image vertically by a maximum of 20% of its total height
Shear	0.2	Randomly shear an image by a random angle within ± 20 degrees
Zoom	0.2	Zoom an image in or out by a factor of up to 20
Horizontal flip	True	Flip an image horizontally

Experiments

Setup

To better demonstrate the capability of the garbage classification problem, and to compare the performance among different advanced existing techniques, we ran different model experiments on our primary dataset. Specifically, we want to try an exhaustive combination of backbone models, data augmentation techniques, and (un)frozen layers, with the exact experiments setup listed out in *Table 3*. For example, in *experiment 1*, we used *ResNet* as our pre-trained model with the last set of layers unfrozen to fit the training dataset that has not been augmented.

In addition to the different model structures we experimented with, we used *Top 1 Accuracy* and *Top 3 Accuracy* as our two metrics to measure the model performance for each experiment. *Top 1 Accuracy* measures, in percentage, how many examples in the test set did the model predict correctly as its top class. *Top 3 Accuracy*, on the other hand, measures the percentage of test samples the model predicted correctly in its top 3 predicted classes.

After running all the experiments, we would also like to pick the model with the best performance measured by *Top 1 Accuracy* to predict labels for the supplement dataset to further demonstrate the capability of our classification models.

Results

Experiment Results Comparison

Table 3: Different experiments run on the training dataset and their associated top 1 & top 3 accuracy score

	Model	Frozen Weights	Data Augmentation	Top 1 Accuracy (%)	Top 3 Accuracy (%)
1	ResNet-101	No	Yes	86.98	98.44
2	ResNet-101	No	No	89.32	99.48
3	ResNet-101	Yes	Yes	88.78	98.62
4	ResNet-101	Yes	No	89.17	99.21
5	MobileNet-V2	No	Yes	76.57	94.88
6	MobileNet-V2	No	No	74.41	95.47
7	MobileNet-V2	Yes	Yes	78.54	96.46
8	MobileNet-V2	Yes	No	75.59	97.24
9	EfficientNet-B7	No	Yes	87.99	99.21
10	EfficientNet-B7	No	No	86.22	99.21
11	EfficientNet-B7	Yes	Yes	86.51	98.62
12	EfficientNet-B7	Yes	No	85.63	99.41

Table 3 shows the relevant metrics for each experiment we ran. Among all setups, *ResNet-101* with *unfrozen weights* and *no data augmentation* (Experiment 3) produced the best *Top 1* and *Top 3* Accuracy. We suspect that *ResNet-101*'s superior performance may result from its high capability of being the deepest net among all pre-trained models we tried. However, the fact that having data augmentation weakened the model's performance is a bit astonishing. It may be due to us *over-augmentating* the data. That is, the data augmentation we applied to the images may be excessive resulting in images being distorted and difficult to recognize, which made it difficult for the model to learn the correct patterns.

Misclassified Images

In addition to model accuracy, we further looked into the misclassified images to see which categories they belong to. As *Figure 3* shows, the accuracy is the highest among *paper* (91.6%). While all other categories had a prediction accuracy of over 80%, our model performed relatively poor on classifying *trash*. We further dove into the *trash* category and found the type of *trash* contained in this category was pretty mixed-up, which made it challenging for the model to learn the pattern.

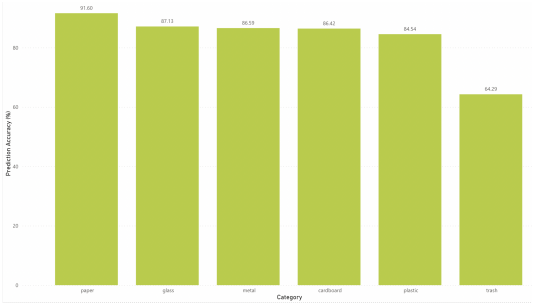


Figure 3: Prediction accuracy (%) by category for the primary dataset

Classification on the Supplementary Dataset

We then used the model from *Experiment 3* to predict the labels on the supplementary dataset. The *Top 1 Accuracy* is around 61.31%, and the *Top 3 Accuracy* is 89.86%. The performance is not that promising compared to the results we got from the primary dataset, which is predictable, as the supplementary dataset differs much in nature from the primary dataset as the supplementary dataset contains images mostly coming from web scraping. The images all have a clear background, with many of the trash being in very good condition, not rotten or damaged, which may potentially cause obstacles for our model to generalize the features learned from the training dataset. Similarly, we visualized the prediction accuracy in percentages for each category as shown in *Figure 4*. Here, the model is terrible in predicting *trash* with an accuracy of only 9.3%. After closely examining the images, we found that most of the *trash* in the supplementary dataset is either *masks* or *diapers*, which differ a lot from the images in the training dataset, making it hard for the model to classify.

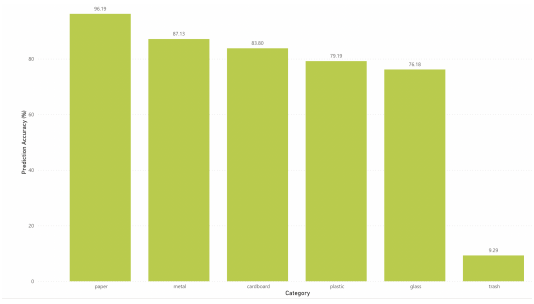


Figure 4: Prediction accuracy (%) by category for the supplementary dataset

Conclusion

Summary

In this report, we first identified our project goal to accurately and efficiently identify the category that different trash belongs to. Then, we introduced the primary and supplementary datasets we used in our model training and experimentation process. After that, we set up different experiments to run on our datasets to evaluate our model performance and demonstrate its capabilities. After showing the results and diagnosing the potential classification challenges our model faced, we next want to identify some limitations in our project and point out some directions for future study.

Limitations & Future Steps

1. The datasets we used in this project both categorized the images into six categories. Neither of these datasets exhaustively included all trash categories, which may result in the model underperforming when it encountered images that did not fit into any of its existing categories. In the future, it would be valuable to train the model on other categories of trash images too. Also, we may be able to boost the model performance by categorizing the training data into more fine-grained categories. For example, the *metal* category in *TrashNet* contains pretty miscellaneous types of metal that could potentially be sorted into finer categories. Sorting them into more specific categories would help the model recognize and generalize patterns more easily and perform better.

2. The data we used in our project all have a clear background. Despite the reason that we want to compare our results with previous work that also used *TrashNet* as the training set, the other reason that we trained only on clear-background images is that we want to mimic the type of indoor environment such as inside the building where people toss the garbage. In the future, if we want to scale our model to have more capacity, we can train the model on other images such as trash found in the wild or trash found underwater. There is a nice repo on *Github* (`https://github.com/AgaMiko/waste-datasets-review`) where the author has put together many trash images with different variants that we can easily find and utilize.

3. The limitation of the *MobileViT* in the application phase is that it does not have a very good pre-trained model available for us to use since it is relatively new. Training from scratch would require a lot of computation power and time, which is limited by the time we have for this project.

With the decent performance achieved by our model studied in this analysis and the potential of expanding the scope of our study to obtain a more advanced and accurate model, we see the future potential of deploying AI in the field of garbage classification. Once we are able to identify garbage with high confidence efficiently, we are able to develop infrastructures around this mechanism that use AI to build a sophisticated waste management system and promote environmental sustainability.

Appendix

A.

Table 4: Summary of the supplementary dataset

Category	Number of Observation	Sample Image
Cardboard	180	
Glass	404	
Metal	770	
Plastic	174	
Paper	221	
Trash	141	
Total	1,890	

B.

ResNet Explanation in Detail

Shortcut connection

Rather than relying on the chance that every few stacked layers will fit the desired underlying mapping, a residual mapping approach is used, in which the layers are explicitly designed to fit the residual mapping. This is achieved through the use of feedforward neural networks with shortcut connections, as shown in this Figure. The shortcut connections, which skip one or more layers, perform identity mapping and add their outputs to the outputs of the stacked layers.

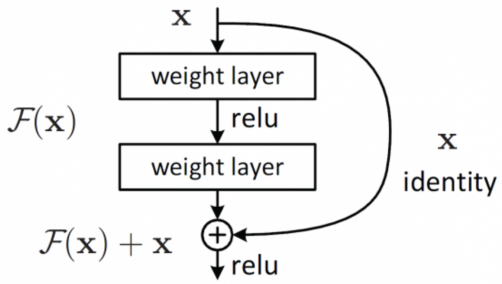


Figure 5

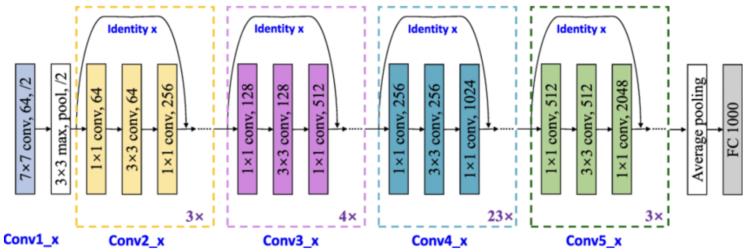


Figure 6: An example of ResNet-101-based deep feature extractor structure

MobileNet Explanation in Detail

Two main invention about *MobileNet*: Inverted residual structure and linear bottlenecks, which allow for increased input and output dimensions in the layers, while the bottlenecks reduce the number of channels in between. This design results in fewer parameters and computations, making the model more efficient and lightweight.

Inverted residual structure

In traditional residual networks, the input is first passed through a bottleneck layer with a smaller number of channels, followed by a convolution layer with a larger number of channels. The result is then added to the original input to form the output. This is known as a residual block. In *MobileNetV2*, the residual structure is inverted. It starts with a convolution layer having a larger number of channels (expansion layer), followed by a depthwise convolution layer with the same number of channels, and then a linear bottleneck layer with a smaller number of channels. This inverted structure helps maintain more information in the intermediate layers and reduces the overall computation cost.

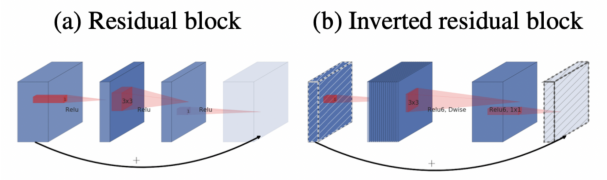


Figure 7

Linear bottlenecks

The linear bottlenecks in MobileNetV2 refer to the layers with a smaller number of channels placed between the depthwise convolution layers and the expansion layers. The purpose of these bottlenecks is to reduce the dimensionality of the feature maps, which in turn reduces the number of parameters and computations involved in the network.

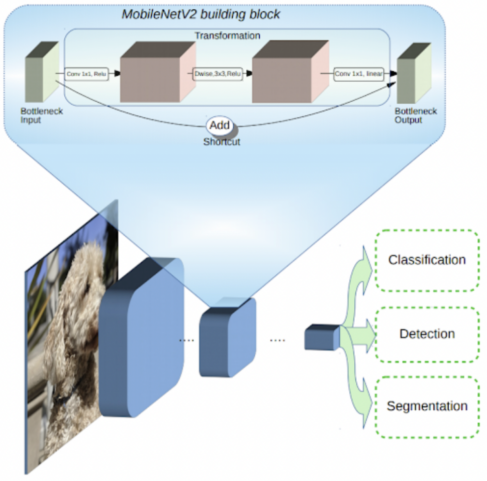


Figure 8

EfficientNet Explanation in Detail

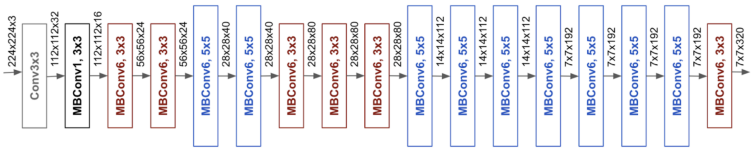


Figure 9: The architecture for baseline network EfficientNet-B0

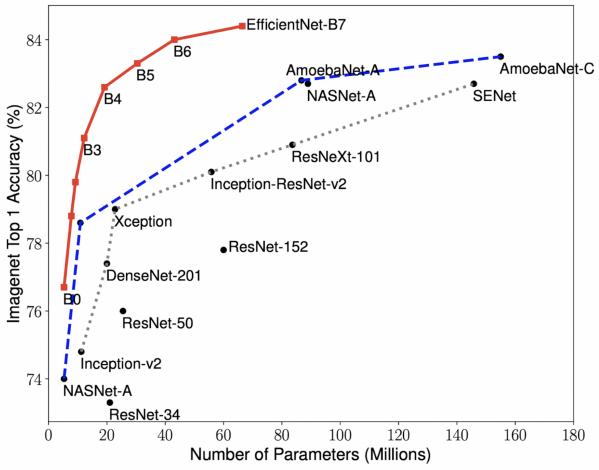


Figure 10: EfficientNet Performance

ViT and MobileViT Architecture

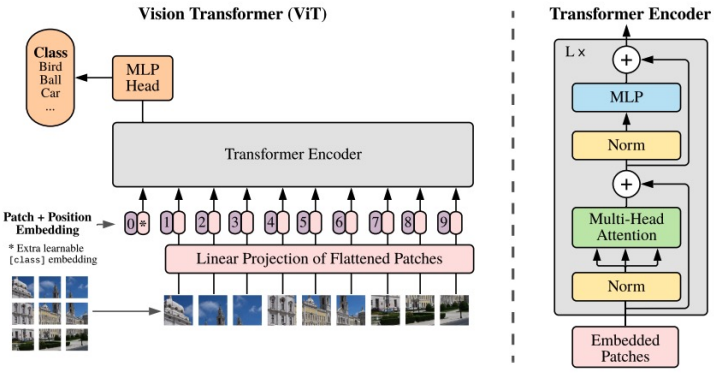
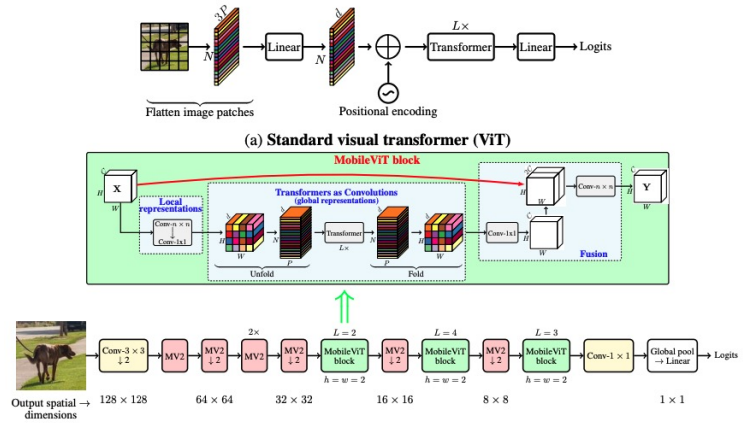


Figure 11: ViT Architecture



(b) **MobileViT**. Here, Conv-$n \times n$ in the MobileViT block represents a standard $n \times n$ convolution and MV2 refers to MobileNetv2 block. Blocks that perform down-sampling are marked with $\downarrow 2$.

Figure 12: MobileViT Architecture

References

Bircanoglu, C., Atay, M., Beser, F., & Genc Ozgun, M. A., Kizrak. (2018). *Recyclenet: Intelligent waste sorting using deep neural networks.* https://www.semanticscholar.org/paper/RecycleNet\%3A-Intelligent-Waste-Sorting-Using-Deep-Bircanoglu-Atay/44e9a393795ce7ccd61b7b1c91e7c83d8e42b94d. (Accessed on: April 21, 2023)

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. (n.d.). *An image is worth 16x16 words: Transformers for image recognition at scale.* https://arxiv.org/abs/2010.11929. (Accessed on: April 23, 2023)

Government of Canada. (2023). *Single-use plastics prohibition regulations - overview.* https://www.canada.ca/en/environment-climate-change/services/managing-reducing-waste/reduce-plastic-waste/single-use-plastic-overview.html. (Accessed on: April 20, 2023)

Huang, H. (2019). *China's radical new rules to recycle rubbish.* https://multimedia.scmp.com/infographics/news/china/article/3038540/china-waste-sorting/index.html. (Accessed on: April 20, 2023)

Liu, J., & Yuan, Z. (2022). *A hybrid deep learning model for trash classification based on deep trasnsfer learning.* https://www.hindawi.com/journals/jece/2022/7608794/?msclkid=a4050768546511f603ff632b554bb80e&utm_source=bing&utm_medium=cpc&utm_campaign=HDW_MRKT_GBL_SUB_BNGA_PAI_DYNA_JOUR_X_PJ_GROUP3&utm_term=\%2Fjournals\%2Fjece\%2F&utm_content=JOUR_X_PJ_GROUP3_JECE. (Accessed on: April 21, 2023)

Mehta, S., & Rastegari, M. (n.d.). *Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer.* https://arxiv.org/abs/2110.02178. (Accessed on: April 23, 2023)

Mohamed, M. (2021). *Garbage classification (12 classes).* https://www.kaggle.com/datasets/mostafaabla/garbage-classification. (Accessed on: April 20, 2023)

Resnet: The basics and 3 resnet extensions. (n.d.). https://datagen.tech/guides/computer-vision/resnet/. (Accessed on: April 22, 2023)

Shi, C., Xia, R., & Wang, L. (2020). *A novel multi-branch channel expansion network for garbage image classification.* https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9165751. (Accessed on: April 21, 2023)

Tan, M., & Le, Q. V. (n.d.). *Efficientnet: Improving accuracy and efficiency through automl and model scaling.* https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html. (Accessed on: April 22, 2023)

Tomar, N. (n.d.). *Vision transformer – an image is worth 16×16 words: Transformers for image recognition at scale.* https://idiotdeveloper.com/vision-transformer-an-image-is-worth-16x16-words-transformers-for-image-recognition-at-scale/. (Accessed on: April 23, 2023)

VO, A. H., SON, L. H., VO, M. T., & LE, T. (2019). *A novel framework for trash classification using deep transfer learning.* https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8930948. (Accessed on: April 21, 2023)

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). *Aggregated residual transformations for deep neural networks.* https://ieeexplore.ieee.org/document/8100117/authors#authors. (Accessed on: April 21, 2023)

Yang, M., & Thung, G. (2016). *Classification of trash for recyclability status.* https://cs229.stanford.edu/proj2016/report/ThungYang-ClassificationOfTrashForRecyclabilityStatus-report.pdf. (Accessed on: April 21, 2023)