Group 3
TrashNet

Jupiter Kang
Long Peter Huynh
Stephen Davanzo

THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC

Introduction

Overall Goal: Improve image classification on the TrashNet dataset to facilitate automated recycling sorting.

- Data Background
  - Original whitepaper
  - Baseline code
- Algorithms
- Experimental Setup
- Results
- Conclusion
- Demo

The Data

- TrashNet dataset created at Stanford Univ. in 2016
  - 2,527 RGB images
  - 512x384 pixels
- No Transfer learning, scratch CNN
- Network Issues
  - Shallow (loosely based on AlexNet)
  - Minimal augmentation
  - No class balancing
  - Minimal hyperparameter tuning

Class Name	Observations	Class Label
Cardboard	403	0
Glass	501	1
Metal	410	2
Paper	594	3
Plastic	482	4
Trash	137	5

The Data

- Baseline Code
  - Kaggle comp
  - Six different models
- Minimal Tuning

Model Name	Year Released	Key Info
MobileNetV2	2018	Fast, lightweight, inverted residuals
ResNet101V2	2016	Deep (101 layers), pre-activation, good feature extraction
ResNet152V2	2016	Very deep (152 layers), great extractions, slow
MobileNet (V1)	2017	Lightweight, fast, depthwise convolutions
MobileNetV3Small	2019	Ultra-efficient, built for small compute power
MobileNetV3Large	2019	Balanced options for speed and accuracy

Algorithms and Networks

- EfficientNetV2S
  - Higher accuracy with fewer parameters
  - Advanced regularization
  - More responsive to training signal
- Other models
  - Shallower, more compressed
  - More parameters
  - Overfit minority classes when weights altered

Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

Experimental Setup

- Initial baseline setup
  - Host data in GCP bucket
  - Use wget to fetch .zip
  - Store in .gitignore-ed folder
- Create dynamic dataloader

Experimental Setup

- Training strategy
  - Phase 1: Initial training
  - Phase 2: Fine-tuning
- Hyperparameters control
  - Initial_epoch, fine_tune_epoch, initial_lr, fine_tune_lr, fine_tune_layers
- Regularization
  - Dropout
  - Earlystopping

Experimental Setup

- Data splitting
  - Test split 20%
- Data augmentation
  - sheer_range, brightness_range
- Optimizer
  - *Adam* - separates initial_lr & fine_tune_lr
  - *AdamW (3)* - decouples weight decay from gradient update process

Experimental Setup

- Class weighting
  - Accuracy suffering due to “Trash” class
  - Automatic class weighting too extreme, scale down impact
- Confusion matrix and learning curves per class

Experimental Setup

- Class Weight:
  $weight_i$ = total / (samples number of classes × samples in $class_i$)

  adjusted $weight_i = 1.0 + \alpha \times (weight_i - 1.0)$

- Dropout Rate 50% (avoid specializing neurons)

- Early Stopping (preserve performance on unseen data)

Results

Model Comparison Results (with Fine-tuning):

	Model	Initial Epochs	Final Train Accuracy	Final Val Accuracy	FineTune Epochs	Test Accuracy	Test Loss
0	EfficientNetV2S	10	0.9062	0.8571	-5	0.8625	0.4549
3	ResNet152V2	10	0.8438	0.7619	-5	0.8062	0.5344
2	ResNet101V2	10	0.8125	0.6667	-1	0.8021	0.5303
1	MobileNetV2	10	0.8438	0.7143	-5	0.7729	0.5804
4	MobileNet	10	0.7812	0.8095	-5	0.7521	0.6457
6	MobileNetV3Large	10	0.2500	0.2381	-1	0.4479	1.4104
5	MobileNetV3Small	10	0.2500	0.2381	-1	0.2604	1.6104

EfficientNetV2S > +9.27% (from previous high)
ResNet101V2 > +3.98%
MobileNetV2 > +2.57%
ResNet152V2 > +2.14%
MobileNetV3Large > +0.58%

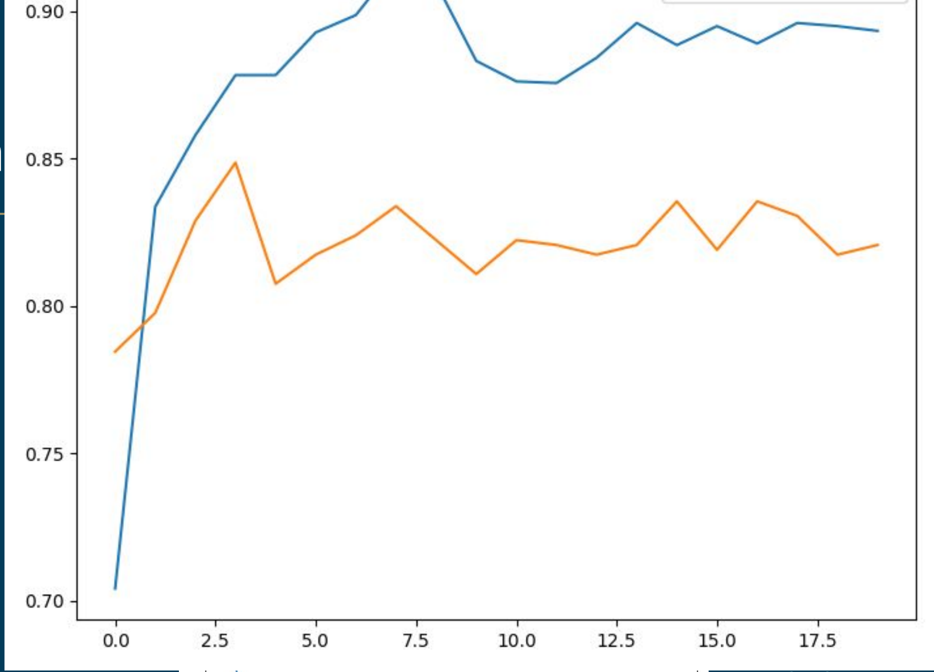
Model Comparison Results:

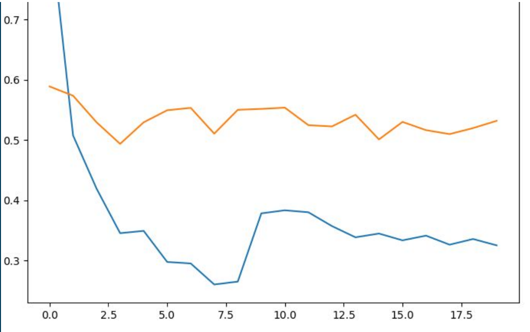
Model	Test Accuracy	Test Loss
ResNet152V2	0.7893	0.6648
ResNet101V2	0.7714	0.731
MobileNetV2	0.7535	0.7557
MobileNet	0.7515	0.6729
MobileNetV3Large	0.4453	1.4076
MobileNetV3Small	0.3559	1.58

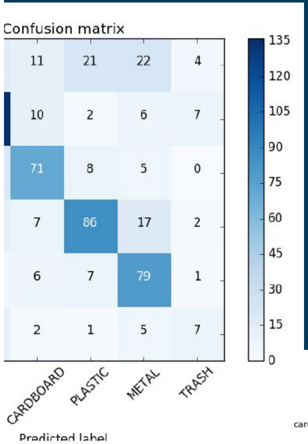
Results

Model	Our Result	Baseline Result	Difference
EfficientNetV2	86.25%	/	/
ResNet152V2	80.62%	78.93%	1.69%
ResNet101V2	80.21%	77.14%	3.07%
MobileNetV2	77.29%	75.35%	1.94%
MobileNet	75.21%	75.15%	0.06%
MobileNetV3Large	44.79%	44.53%	0.26%
MobileNetV3Small	26.04%	35.59%	-9.55%

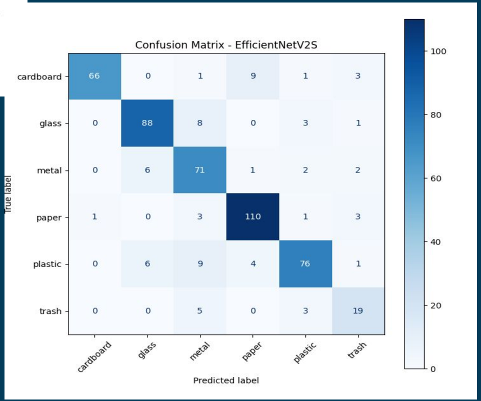
Con







Confusion Matrix from original paper (SVM)
66% accuracy



Confusion Matrix from our top model (EffNet)
86% Accuracy

Demo

References

- Thung, G., & Yang, M. (2016). Classification of trash for recyclability status. Stanford University. Retrieved from https://cs229.stanford.edu/proj2016/report/ThungYang-ClassificationOfTrashForRecyclabilityStatus-report.pdf
- He, J., Zhao, Y., Zaslavskiy, M., Wang, X., Lin, Z., Jin, Q., & Wang, W. Y. (2021). TransFG: A transformer architecture for fine-grained recognition. arXiv preprint arXiv:2104.00298. https://arxiv.org/abs/2104.00298
- Yassin, A. (2024, November 13). Adam vs. AdamW: Understanding Weight Decay and Its Impact on Model Performance. *Medium*. https://yassin01.medium.com/adam-vs-adamw-understanding-weight-decay-and-its-impact-on-model-performance-b7414f0af8a1

Thank you.