Neural Architecture Design for Human Classification

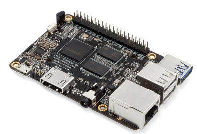
STUDENTS: Logan Krumholz, Mounica Ramreddy, Sharan Ranjit S, Weijing Li, Linyu Yang, Jason Wu, Harsheeta Venkoba Rao

Human Classification

- Amazon Lab 126 tasked our group with curating a bias-free human dataset and developing a deep learning neural network for human classification.
- The neural network makes use of MobileNetV2 as a backbone and is deployable on multiple edge devices.
- Human classification is a useful tool for security and being deployable on edge devices allows for numerous and varied applications.
- Accuracy, size, and frame rate were prioritized and measured to assess the usability and reliability of the various models trained in both PyTorch and TensorFlow.

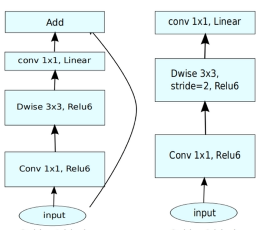
Edge Devices

- Two edge devices used over the course of the project: Libre Computer Board ROC RK3328 and Raspberry Pi 4 Model B.
- Libre was originally used as cheaper alternative, although slower processing speed than Raspberry Pi.
- Edge devices served to pose real world constraints of model size and frame rate.



MobileNetV2 Architecture

- MobileNetV2 architecture is a convolution based neural network (CNN) structured for optimal performance on mobile devices.
- The architecture is based on an inverted residual structure where connections are formed between bottlenecked layers.
- The incorporation of lightweight depth wise convolutions to filter features in the intermediate expansion layers, makes MobileNetV2 generally smaller than other CNNs.
- Compared to other CNN architectures such as VGG or ResNet, MobileNetV2 models typically have a smaller number of parameters and require less memory and computational power.
- These aspects made it an ideal choice for human classification on edge devices, where bulkier models would be less ideal.



Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

t: expansion factor c: output channel
n: repeat time s: stride

IBN and NAS

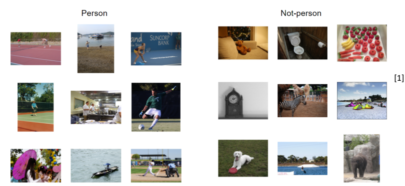
Comparing MobileNetV2 model with other architectures: FuseNet IBN architecture and using NAS methods

- Neural Architecture Search (NAS) refers to the process of automatically discovering optimal neural network architectures for a given task or dataset. Instead of manually designing and tuning the architecture, NAS employs search algorithms or reinforcement learning techniques to explore a vast space of possible architectures and identify the most effective ones.
- Fused IBN-Net: Fused Inverted Bottleneck Net (Fused IBN Net) is an advanced neural network architecture that combines the benefits of both Inverted Bottleneck (IBN) and feature fusion techniques.
- Comparison shows that our MobileNetV2 architecture performs better than the other models.

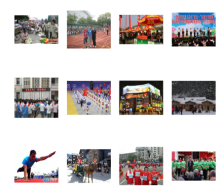
Datasets

- Having multiple bias-free datasets to create and compare different models is crucial to combat any biases that often plague deep learning fields.
- The COCO and WiderPerson datasets were selected based on their diverse image pool.
- The datasets incorporate pictures of entire human frames as well as individual body shots from various distances to ensure varied and accurate human classification.

COCO



WiderPerson



Accuracy and Testing

- We trained the model in three datasets separately: COCO, COCO+WiderPerson, WiderPerson. The final validation accuracies are 93.7%, 98.3%, 93.0%. The model can identify human after training.

Human classification metrics

Dataset	Accuracy	Precision	Recall	F1 score
COCO	0.937	0.912	0.975	0.94
COCO+Wider	0.930	0.907	0.967	0.93
WiderPersons	0.982	0.974	0.991	0.98

- For further testing, we did more experiments about model compression and model size which can be important to edge device deployment. The accuracies and model sizes are shown in left chart(taking COCO-Wider for example).

COCO-Wider

D, Params	Model type	Model size	Accuracy
0.35, 411K	h5	3.2M	88.66
	fp32	1.6M	88.66
	fp16	821K	88.64
	int	611K	88.38
	dyn	538K	88.52
0.5, 707K	h5	5.3M	91.08
	fp32	2.7M	91.08
	fp16	1.4M	91.04
	int	953K	90.78
	dyn	852K	91.34
0.75, 1.38M	h5	10M	92.32
	fp32	5.2M	92.32
	fp16	2.7M	92.36
	int	1.7M	92.72
	dyn	1.6M	92.42
1, 2.26M	h5	17M	92.9
	fp32	8.5M	92.9
	fp16	4.3M	92.94
	int	2.6M	92.84
	dyn	2.4M	92.9

Model Compression and Quantization

- Quantization refers to the process of reducing the precision of numerical values in a neural network model. It involves converting floating-point values, typically 32-bit, to lower-precision fixed-point values, such as 16-bit or even 8-bit integers. This reduction in precision allows for more efficient computation and memory usage, which can lead to improved performance on resource-constrained devices like the Raspberry Pi.
- In our project, we implemented Post Training Quantization and Quantization Aware Training(QAT). Post-training quantization involves applying quantization to a pre-trained model after it has been trained using full precision, while QAT is a technique that incorporates quantization into the training process itself.

QAT

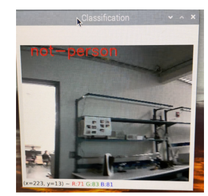
D, Params	Model type	Model size	Accuracy
0.35, 420K	h5	4.0M	90.34
	fp32	3.3M	90.44
	fp16	607K	90.44
	int	607K	90.44
	dyn	607K	90.44
0.5, 719K	h5	6.2M	90.18
	fp32	5.5M	90.54
	fp16	949K	90.42
	int	949K	90.42
	dyn	949K	90.42

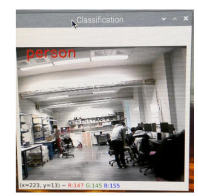
Post Quantization

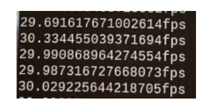
D, Params	Model type	Model size	Accuracy
0.35, 411K	h5	3.2M	88.66
	fp32	1.6M	88.66
	fp16	821K	88.64
	int	611K	88.38
	dyn	538K	88.52
0.5, 707K	h5	5.3M	91.08
	fp32	2.7M	91.08
	fp16	1.4M	91.04
	int	953K	90.78
	dyn	852K	91.34
0.75, 1.38M	h5	10M	92.32
	fp32	5.2M	92.32
	fp16	2.7M	92.36
	int	1.7M	92.72
	dyn	1.6M	92.42
1, 2.26M	h5	17M	92.9
	fp32	8.5M	92.9
	fp16	4.3M	92.94
	int	2.6M	92.84
	dyn	2.4M	92.9

Real Time Detection with Raspberry Pi

- We implemented an int8 quantized model on the Raspberry Pi, boasting an impressive accuracy of 93%.
- This model enables the device to discern the presence of people within the camera's field of view. Moreover, it sustains a detection rate of at least 30fps.







Future Work, References, and Acknowledgments

- Further improvements in model compression and frame rate
- Additional research and implementation of Fused IBN-Net for our models.
- Paper on novelties regarding customized MobileNetV2 architecture usability on edge devices.

Faculty: Payman Arabshahi, Nathan Kutz
Graduate Student: Harsha Vardhan

[1] "Papers with Code - Visual Wake Words Dataset." *Dataset | Papers With Code*, paperswithcode.com/dataset/visual-wake-words. Accessed 23 May 2023.
[2] Howard, Andrew G., et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." *arXiv.Org*, 17 Apr. 2017, arxiv.org/abs/1704.04861.

ELECTRICAL & COMPUTER ENGINEERING
UNIVERSITY of WASHINGTON

INDUSTRY MENTOR: Mansi Manohara

SPONSOR: Amazon Lab 126