Benchmarking Energy on Android

Om Shri Prasath
ee17b113@smail.iitm.ac.in

Pruthvi Raj R G
ee17b114@smail.iitm.ac.in

SysDL Project
CS 6886

GITHUB LINK TO PROJECT

Problem definition

Various architecture design parameters have non-trivial interactions and thus lead to different performance on energy and runtime. Moreover, these performances depend drastically on the network under inference. Understanding these patterns enables us to better co-design architectures and networks.

Our aim is to develop an Android App which can benchmark energy usage of Deep Neural Networks on various Android Devices. Also, we want to formulate an opinion piece on energy usage characteristics of Deep Neural Networks on Android Devices.

Related Work

- Research paper on Energy Consumption of ML models
- Android documentation for measuring device power
- Device AI score estimation by ETH Zurich
- ML model inference sample code by Artem Malynovskyi
- Documentation for converting models into TensorFlow Lite

Key technical insight

- Power usage patterns can be unstable; hence we will rely on battery charge level difference before and after the inference. We will subtract the average idle phone battery consumption(method changed later)
- We will explore the tradeoff between model complexity(size), accuracy, energy consumption, inference time on different devices
- We will observe the effect of model quantization on the accuracy, inference time, and energy consumption
- We will also observe the effect of using depthwise convolutions instead of 2D convolutions
- We will require special frameworks like TFLite to use deep learning methods on Android

Milestones - To be achieved by Nov. end

- Write code for converting TensorFlow models into TF Lite.
- Write a function to download the TF model from the given link.
- Write a method for downloading the inference dataset to the device and upload inference results to the cloud.(not adopted)
- Write a function to load the TF Lite model and run the inference on the device in Android studio.
- Finalize a method to access power and energy usage details during the inference.

Milestones - To be achieved by Dec mid

- Build a cloud application for hosting the dataset, TF Lite model, and receiving the submitted inference file, timing, and energy statistics.
- Install the app on different devices and benchmark them.
- Write an opinion piece on the energy consumption of android devices for different models.
- Ideate on breaking the inference further to understand the energy split up between different layers in the network on a given Android device.(not-covered)

MILESTONE 1 - PROGRESS

CODE FOR CONVERTING TENSORFLOW MODELS INTO TF LITE

Tensorflow1 Tensorflow2

- Our code takes in the model link of a ".pb" model and returns a ".tflite" model
- Reference

FUNCTION TO DOWNLOAD TF MODEL FROM THE GIVEN LINK

Link to Gist

- We use the Android library PRDownloader to download the given TFLite Model from the given URL - Repo
- The model is stored in the internal storage of the app allowing us to access it during inference

METHOD FOR DOWNLOADING THE INFERENCE DATASET TO THE DEVICE AND UPLOAD INFERENCE RESULTS TO THE CLOUD

Link to Gist1

Link to Gist2

- We changed the plan of downloading the dataset from the link. Instead, we decided to keep a subset of the dataset stored in the app assets
- We are currently hosting the server code in our local system - were able to successfully receive the correct predictions from the app

WRITE A FUNCTION TO LOAD THE TF LITE MODEL AND RUN THE INFERENCE ON THE DEVICE IN ANDROID STUDIO

Link to Gist

- We used the TFlite library provided by Google to load the ".tflite" model
- We preprocessed the image and predicted classes for the images stored in the assets

FINALIZE A METHOD TO ACCESS POWER AND ENERGY USAGE DETAILS DURING THE INFERENCE

Link to the Gist

- We are using "BatteryManager" in android to log the current battery statistics of the android device

Gist of MainActivity.java of the app

MILESTONE 2 - PROGRESS

FINALIZE THE ENERGY STATISTICS

- We calculate energy consumption by collecting voltage and current values during the run of models
- We have the functionality to calculate the idle energy consumption before running the model. This will be subtracted from the total energy to give the net energy consumption of the run

OPTION TO SHARE THE COLLECTED DATA

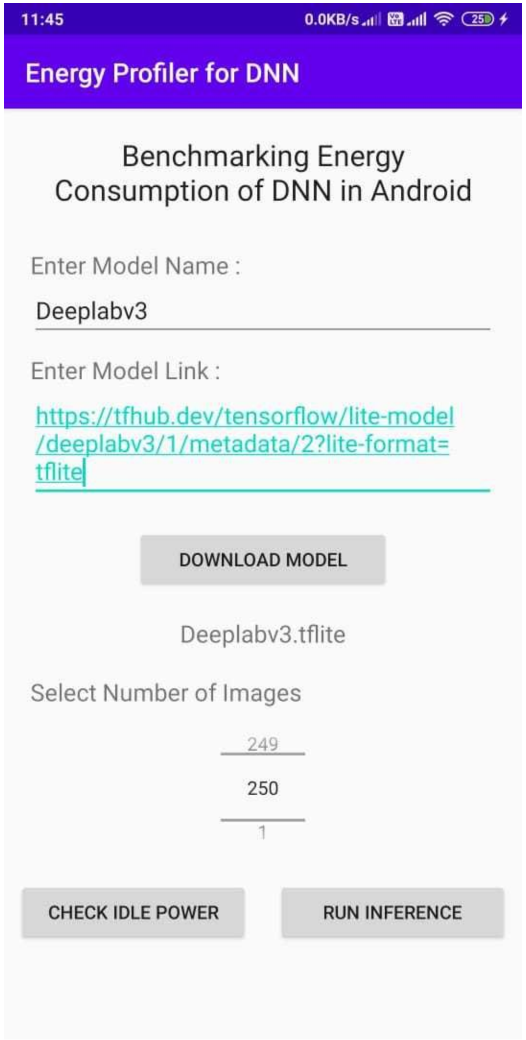
- We create a JSON object containing the statistics after the completion of each run
- Our initial plan was to host a cloud instance and upload the results to the cloud instance. However, we decided to give users the option to share the results directly through WhatsApp to avoid additional costs of running the cloud instance

FINALIZATION OF MODELS

- Our goal is to check the energy comparison between quantized and unquantized models. Hence we picked the quantized model for each unquantized model whenever it was available
- We wanted to evaluate different models, each built for different purposes, including classification, segmentation, and style transfer. More info on the models : link

Model Names	Model Names
mobilenet_v2_1.0_224	magenta/arbitrary-image-stylization-v1-256/fp16/prediction
mobilenet_v2_1.0_224_quantized	deeplabv3
efficientnet/lite4/fp32	custom_conv2d
efficientnet/lite4/int8	custom_depthwise_conv2d

SCREENSHOTS OF THE APP



11:45 0.0KB/s

Energy Profiler for DNN

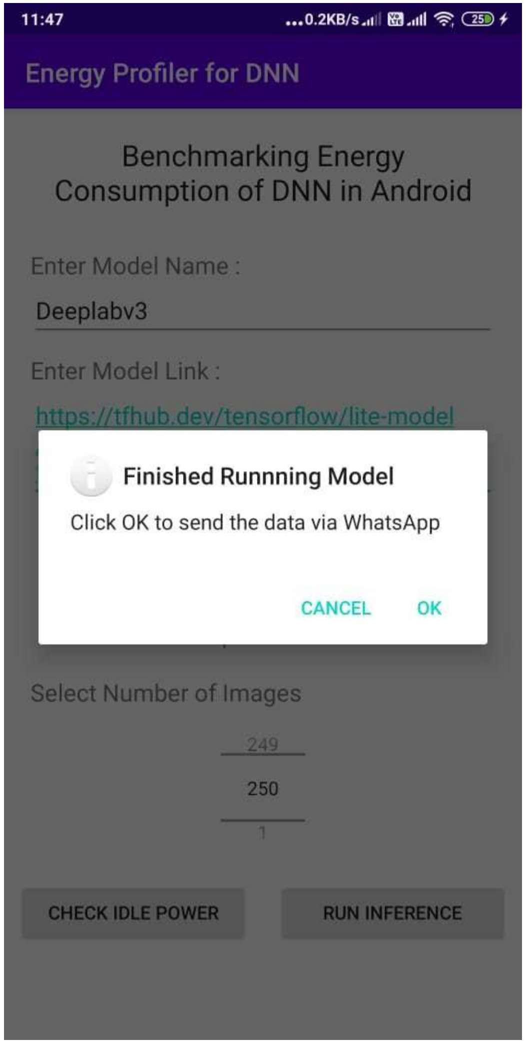
Benchmarking Energy Consumption of DNN in Android

Enter Model Name : Deeplabv3

Enter Model Link : https://tfhub.dev/tensorflow/lite-model/deeplabv3/1/metadata/2?lite-format=tflite

Deeplabv3.tflite

Select Number of Images: 250



11:47 0.2KB/s

Energy Profiler for DNN

Benchmarking Energy Consumption of DNN in Android

Enter Model Name : Deeplabv3

Enter Model Link : https://tfhub.dev/tensorflow/lite-model

Finished Runnning Model

Click OK to send the data via WhatsApp

Link to main activity gist : Link

RESULTS AND OPINION PIECE

TASK AND MODEL SIZE

TASK AND MODEL	MODEL SIZE IN MB
Classification - Mobilenet	13.34
Classification - Efficientnet	49.50
Style transfer(Encoder)-Magenta	4.49
Segmentation(Semantic)-Deeplab	2.65

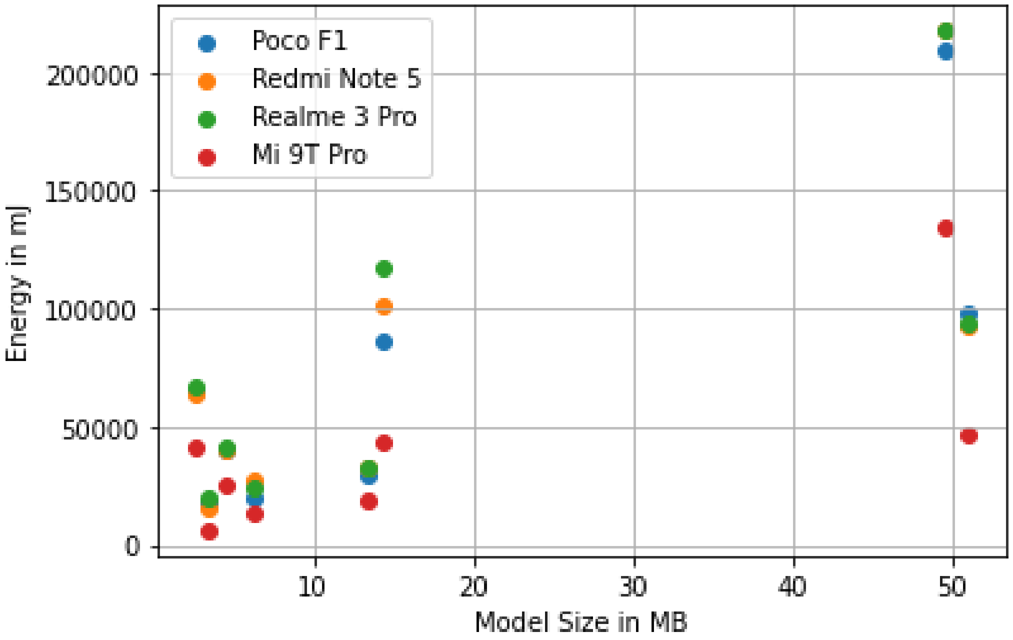
- We can observe that model size varies with the type of the task and and depends heavily on the network architecture

DEVICE SPECIFICATIONS

DEVICE NAME	PROCESSOR	RAM	BATTERY CAPACITY
Redmi Note 5 Pro	Qualcomm Snapdragon 636	4 GB	4000 mAh
POCO F1	Qualcomm Snapdragon 845	6 GB	4000 mAh
RealMe 3 Pro	Qualcomm Snapdragon 710	6 GB	4045 mAh
Mi 9T Pro	Qualcomm Snapdragon 855 processor	6 GB	4000 mAh

RESULTS AND OPINION PIECE

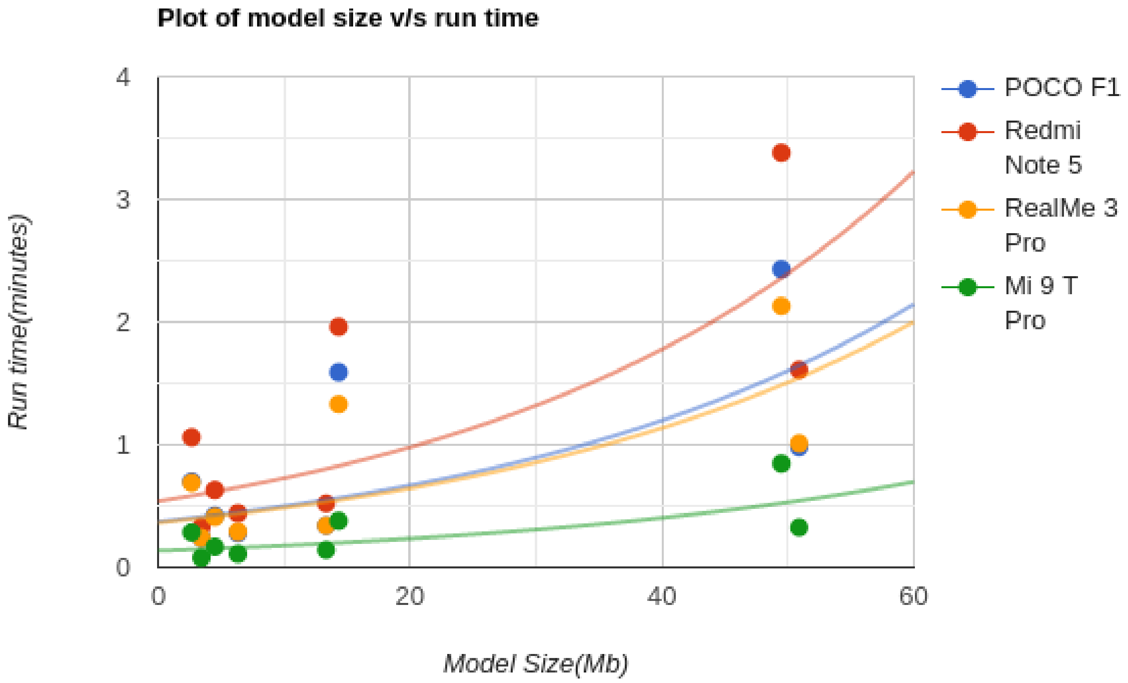
MODEL SIZE V/S ENERGY CONSUMPTION



- We can observe that energy consumption of run increases as model size increases
- It is interesting to note that the energy consumption of runs has very similar values across various devices except for Mi 9T Pro(because of it's on-device AI engine)
- Only EfficientNet quantized gives a significant difference in energy consumption across different devices

RESULTS AND OPINION PIECE

MODEL SIZE V/S INFERENCE TIME



- We can observe that the inference time also increases with model size
- Interesting to note that the inference times are indeed very close between POCO F1 and RealMe 3 Pro (very similar processors and same RAM size)

RESULTS AND OPINION PIECE

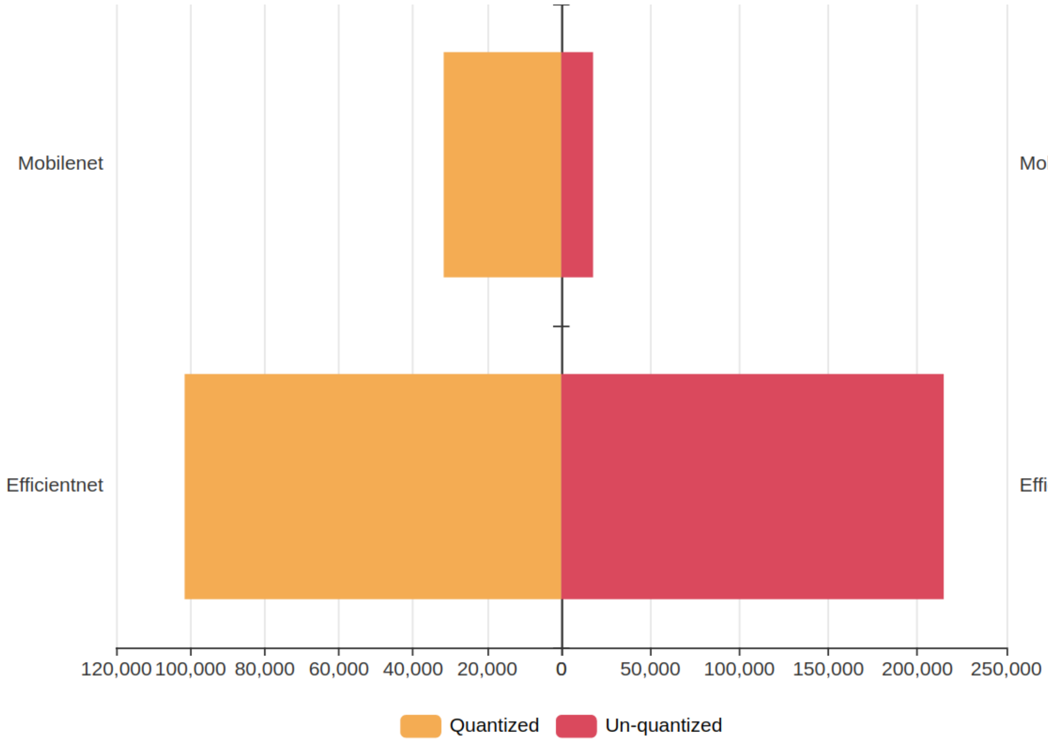
EFFECT OF QUANTIZATION ON MODEL SIZE

MODEL TYPE	MODEL SIZE IN MB
Mobilenet	13.34
Mobilenet quantized	3.42
Efficientnet	49.50
Efficientnet quantized	14.34

- We can observe that model size is drastically decreased by ~4x times when we quantized the model weights, this is consistent as size(float32) = 4*(size(int8))
- Top 1 accuracy only decreased by 1% from 71.8% to 70.8% when we quantize the Mobilenet model
- Top 1 accuracy only decreased by 0.7% from 75.1% to 74.4% when we quantize the Efficientnet model

RESULTS AND OPINION PIECE

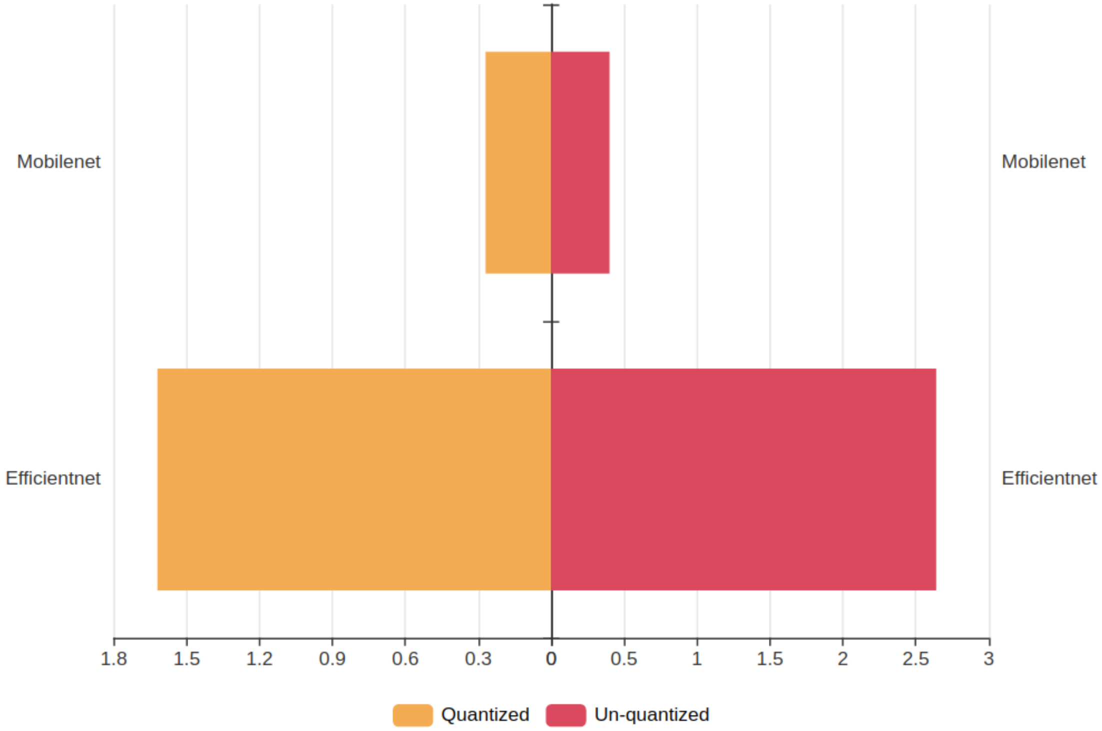
EFFECT OF QUANTIZATION ON ENERGY(MJ)



- We can observe that the energy consumption of quantized models is ~2x less compared to un-quantized models

RESULTS AND OPINION PIECE

EFFECT OF QUANTIZATION ON INFERENCE TIME(MINUTES)



- We can observe that the average inference time is ~1.5x times less for quantized models as compared to un-quantized models

RESULTS AND OPINION PIECE

EFFECT OF DW CONV ON INFERENCE TIME(SECONDS), MODEL SIZE(IN MB) AND ENERGY CONSUMPTION(MJ)

CONVOLUTION TYPE	MODEL SIZE	INFERENCE TIME	ENERGY
CONV2D	6.6	72.47	94907.93
DEPTHWISE SEPERABLE	53.4	20.30	23933.17

- We can observe that the average inference time is ~3.5x less when we use depthwise convolution as compared to the conv2d model
- The energy consumption of the depthwise convolution model is ~4x less when compared to the conv2d model

CONCLUSION

- Model sizes vary with different tasks and architecture used in the design of the network
- Model quantization helps to reduce inference time and energy consumption with a small decrease in accuracy
- Models using depthwise convolution has better inference times and energy consumption stats
- Deep learning runs consumed comparable energy to other task intensive apps

The world as we have created it is a process of our thinking. It cannot be changed without changing our thinking

— Albert Einstein

Om Shri Prasath
ee17b113@smail.iitm.ac.in

Pruthvi Raj R G
ee17b114@smail.iitm.ac.in

SysDL Project
CS 6886