Annotated Bibliography
Efficient Deep Learning Architecture on Mobile (Tugii)

Cheon, Y., Hong, S., Kim, K.-H., Park, M., & Roh, B. (2016). PVANET: Deep but Lightweight Neural Networks for Real-time Object Detection. arXiv:1608.08021 [Cs]. Retrieved from http://arxiv.org/abs/1608.08021

The authors, researchers at the Intel Imaging and Camera Technology in Seoul, demonstrate how a state of the art model for object detection can have minimal computational cost while still retaining stellar accuracy. The main philosophy of minimizing the model is to reduce the number channels and increase the layers. They conclude the research indicating that the minimized model requires only 12.3% of the normal model. Main contributions of the research include usage of concatenated rectified linear units of early stages, adoption of the inception for later layers and concatenation of multi-scale intermediate outputs to make the model thin. They conclude that current neural network architectures are highly redundant and that light networks could be designed for complex tasks such as image recognition. They also note that their work is independent of recent network compression and quantization methods that have been developed and that the optimized can be further decreased using those techniques.

Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M. A., & Dally, W. J. (2016). EIE: Efficient Inference Engine on Compressed Deep Neural Network. arXiv:1602.01528 [Cs]. Retrieved from http://arxiv.org/abs/1602.01528

The authors, researchers at Stanford University and NVIDIA, proposal builds on top of the previous research on Deep Compression, which made it possible to fit large DNNs fully in on-chip SRAM. They propose an energy efficient inference engine that performs the inference on this compressed network model. According to their benchmarks, EIE has 2.9x, 19x and 3x better throughput, energy efficiency and area efficiency. They achieve these results by exploiting dynamic sparsity activations and parallelizing to save computation, which results in energy savings. They have performed experiments running both the compressed and uncompressed models in order to quantify the improvement, as well as, running the experiments on a mobile GPU. In addition, they have tested out different deep learning models such as CNN and LSTM. I think this paper greatly aids my research by pointing out that we can perform inferences even while the model is still compressed. It also introduces another dimension, energy and battery, which can be an important aspect in terms of deploying deep learning models on mobile systems.

Han, S., Mao, H., & Dally, W. J. (2015). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv:1510.00149 [Cs]*. Retrieved from http://arxiv.org/abs/1510.00149

The authors, Starford and Tsinghua researchers, propose a three step pipeline for compressing a deep neural network: pruning, trained quantization and huffman coding **without affecting accuracy**.

Essentially, pruning leaves allows only the important connections, which means that the model size will reduce as a result. Quantizing allows shared weights, which reduces number of bits it takes to represent the weight. Finally, Huffman coding is commonly used for lossless data compression. They indicate that the final compressed model is 35-49 times smaller than the uncompressed one. The paper also makes an important point between SRAM and DRAM, emphasizing how exactly they affect energy consumption of the device. They state that their goal is to also reduce storage and energy requirements requirements to run the model. The paper about EIE adds on to this idea and allows efficient execution of the inference. This is one of the original papers published last year and has already been cited 56 times, which indicates the originality and novelty of the contribution to the field.

Hu, H., Peng, R., Tai, Y.-W., & Tang, C.-K. (2016). Network Trimming: A Data-Driven Neuron Pruning Approach towards Efficient Deep Architectures. *arXiv:1607.03250 [Cs]*. Retrieved from http://arxiv.org/abs/1607.03250

The authors, researchers at Hong Kong University, propose a data-driven approach for the pruning process. The research is based on the observation/claim that there are redundant weak neurons that don't contribute a great to the overall model and can be removed without affecting the accuracy significantly. First, the network is trained using a conventional method and then validated afterwards to determine the neurons that contribute the least. Those neurons are trimmed from the network and the rest of the model is retrained so that it still works without the removed neurons. They have used the highly performant Caffe library and tested the method on LeNet and VGG-16 models resulting in 2-3x less parameters while retaining the original accuracy. This paper relates to the deep compression paper to further build on the pruning subsection and try to optimize it based on data analytics of running the model on a validation set. The benefit of this approach is that it does this process automatically without need for manual tuning as they did on the Google Translate project.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *arXiv:1602.07360 [Cs]*. Retrieved from http://arxiv.org/abs/1602.07360

The authors, researchers at Stanford and Berkeley, propose an architecture called SqueezeNet that aims to decrease the number of parameters while retaining original accuracy of the model. Their main strategy is to use smaller filters and downsample in later training stages. They also introduce a Fire Module, which is made up of squeeze convolution layer that has 1x1 and 3x3 filters. Each Fire Module comprises of three dimensional hyperparameters. They have squeezed AlexNet and reduced its number of parameters 50x. This paper contributes by creating an overarching architecture for the methods that have already been developed - deep compression, network pruning and EIE, as well as, introducing general principles to follow when building the

model, such as using smaller filters. I am not exactly sure how transferable their approach is since it seems pretty opinionated about the details of the architecture.