**Add A Catchy Title Here**

FirstName LastName

[EmailAddress@utdallas.edu](mailto:EmailAddress@utdallas.edu)

**Abstract**

*Write this last. Summarize each of the below sections using ~ 1 sentence each.*

**1. Introduction**

Write 1 short paragraph on the importance of xNN accuracy vs performance.

Write 1 short paragraph on the contents of the rest of the paper.

**2. Related Work**

The network in this paper was heavily inspired by SENet [1], MobileNetV2 [2], MnasNet [3], CondConv [4], MobileNetV3 [5] and EfficientNet [6].

Summarize the following papers with 1 or 2 sentences on the key ideas in each (I’m not looking for a book). I’ve included some starters to get you going.

MobileNetV2 [2] used inverted residuals …

MnasNet [3] added squeeze and excite operations to the inverted residuals and used neural architecture search to optimize the network depth and width …

MobileNetV3 [5] replaced the swish nonlinearity with a hard swish nonlinearity, improved the design of the final encoder stages and used a new neural architecture search to optimize the depth and width …

EfficientNet [6] refined the mobile baseline network and focused on the combined scaling of network input resolution, depth and width to design larger networks …

A variant of EfficientNet replaced the convolution operations with conditional convolution operations [4] …

**3. Design**

Describe that it’s a relatively standard ImageNet structure in figure 1 and table 1 with the 1st 2 stride by 2 operations change to a stride by 1 operation as the image size has 1/4 the rows and cols of typical ImageNet images. Think encoder (stem and body) and decoder (head).

Describe the operations in the 3 types of building blocks in figures 2 that can be used within this modified ImageNet structure: standard building block, SE enhanced building block (see figure 3 for details on the SE operation) and SE and conditional convolution enhanced building block (see figure 4 for details on the conditional convolution operation).

Note a few key points on differences with this design and EfficientNet: this design only used 3x3 filters in the fully grouped convolution, modified the stem width, modified the depth and width of various blocks, modified the inverted residual expansion ratio, simplified the nonlinearity choice to ReLU (or Sigmoid where appropriate), only included 3 levels of down sampling as the cropped input is 3x56x56, …

For implementations based on the SE enhanced building block, the internal rank reduction ratio R = 4.

For implementations based on SE and conditional convolution enhanced building blocks, the number of experts M = 4.



**Figure 1**: Network structure; the linear layer output dimension and bias dimension is the number of classes



**Figure 2**: [A] Standard building block, [B] SE enhanced building block and [C] SE and conditional convolution enhanced building block



**Figure 3**: Squeeze and excite uses a learned input dependent per channel weighting to re weight input feature maps with internal rank reduction R



**Figure 4**: Conditional convolution uses 3D / 4D convolution weight tensors Wm from M experts combined to a single 3D / 4D weight tensor W via a learned input dependent weighted sum with fully grouped convolution / not fully grouped convolution

|  |  |  |
| --- | --- | --- |
| Re-peat | Input NixWxH | Operation |
| 1 | 3x56x56 | Conv (3x3/1), Batch Norm, ReLU |
| 1 | 16x56x56 | Block (Ne=4Ni, F=3, S=1, ID=True) |
| 1 | 16x56x56 | Block (Ne=4Ni, F=3, S=1, ID=False) |
| 1 | 24x56x56 | Block (Ne=4Ni, F=3, S=1, ID=True) |
| 1 | 24x56x56 | Block (Ne=4Ni, F=3, S=2, ID=False) |
| 2 | 40x28x28 | Block (Ne=4Ni, F=3, S=1, ID=True) |
| 1 | 40x28x28 | Block (Ne=4Ni, F=3, S=2, ID=False) |
| 3 | 80x14x14 | Block (Ne=4Ni, F=3, S=1, ID=True) |
| 1 | 80x14x14 | Block (Ne=4Ni, F=3, S=2, ID=False) |
| 4 | 160x7x7 | Block (Ne=4Ni, F=3, S=1, ID=True) |
| 1 | 160x7x7 | Block (Ne=4Ni, F=3, S=1, ID=False) |
| 1 | 320x7x7 | Conv (1x1/1), Batch Norm, ReLU |
| 1 | 1280x7x7 | Global Avg Pool, Linear, Bias |
| 1 | 1x100 | Output |

**Table 1**: Network specification; block is either a [A] standard building block, [B] SE enhanced building block or [C] conditional convolution enhanced building block

**4. Training**

Table 2 includes a summary of all training hyper parameters. Note that this is a ~ generic ImageNet training routine such as you would find in RegNetX/Y [7]. Training routines that use more complex data augmentation, additional data, different train and test resolutions, more epochs, … can achieve higher accuracies.

Table 3 includes final training results and figure 5 shows a plot of the per epoch accuracy and loss curves.

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Add | Add |
| Add | Add |
| Add | Add |
| … | … |

**Table 2**: Training hyper parameters

|  |  |  |
| --- | --- | --- |
| **Block** | **Training time** | **Accuracy** |
| Standard | Add | Add |
| SE enhanced | Optional | Optional |
| SE and cond conv enhanced | Optional | Optional |

**Table 3**: Training final results

Add a per epoch plot of accuracy and loss here

**Figure 5**: Training per epoch accuracy and loss curves

**5. Implementation**

Table 4 shows per operation MACs and number of filter coefficients for the stem convolution, convolutions in all standard blocks (taking into account repeats) and the head convolution and matrix multiplication, along with their sum for the full network.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operation** | **Rep** | **MAC** | **Filter Cx** |
| Conv (3x3/1), Batch Norm, ReLU | Add | Add | Add |
| Block (Ne=4Ni, F=3, S=1, ID=True) | Add | Add | Add |
| Block (Ne=4Ni, F=3, S=1, ID=False) | Add | Add | Add |
| Block (Ne=4Ni, F=3, S=1, ID=True) | Add | Add | Add |
| Block (Ne=4Ni, F=3, S=2, ID=False) | Add | Add | Add |
| Block (Ne=4Ni, F=3, S=1, ID=True) | Add | Add | Add |
| Block (Ne=4Ni, F=3, S=2, ID=False) | Add | Add | Add |
| Block (Ne=4Ni, F=3, S=1, ID=True) | Add | Add | Add |
| Block (Ne=4Ni, F=3, S=2, ID=False) | Add | Add | Add |
| Block (Ne=4Ni, F=3, S=1, ID=True) | Add | Add | Add |
| Block (Ne=4Ni, F=3, S=1, ID=False) | Add | Add | Add |
| Conv (1x1/1), Batch Norm, ReLU | Add | Add | Add |
| Global Avg Pool, Linear, Bias | Add | Add | Add |
| **Total** | – | Add | Add |

**Table 4**: Per operation and total MAC and filter coefficient counts for all trainable operations

**6. Conclusion**

Write this paragraph next to last. Summarize the introduction, related work, design, training and implementation sections with ~ 1 sentence each.

Then add a final paragraph with some thoughts for future modifications and experiments (e.g., different filter sizes, different widths, depths and repeats of different blocks, different residual expansion factors, different SE rank reduction factors, different numbers of mixtures of experts, different training strategies, …).

**References**

[1] J. Hu et. al., "Squeeze-and-excitation networks," arXiv:1709.01507, 2017.

[2] M. Sandler et. al., "MobileNetV2: inverted residuals and linear bottlenecks," axXiv:1801.04381, 2018.

[3] M. Tan et. al., "MnasNet: platform-aware neural architecture search for mobile," arXiv:1807.11626, 2018.

[4] B. Yang et. al., "CondConv: conditionally parameterized convolutions for efficient inference," arXiv:1904.04971, 2019.

[5] A. Howard et. al., "Searching for MobileNetV3," arXiv:1905.02244, 2019.

[6] M. Tan and Q. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," arXiv:1905.11946, 2019.

[7] I. Radosavovic et. al., "Designing network design spaces," arXiv:2003.13678, 2020.