CS182 Project Commentary

EfficientNet: Understanding the Compound Scaling Method

Siqi Zhu, Ethan Qiu, Yuexi Shen, Charlie Cheng

Part 0: Abstract

EfficientNet introduces a novel model architecture scaling approach, optimizing for FLOPs and parameter efficiency. In general, modeling tasks are dependent upon resources available. EfficientNet proposes a systematic scaling method with which to systematically resize network depth, width and resolution to achieve resource efficiency.

We want to emphasize the effect of compound model scaling: how EfficientNet is efficient, what each convolution is calculating and how many convolutions are involved when processing each image. To further prospective student's understanding of the paper, we produced some analytical, and then several implementation tasks relating to the paper.

Part I: Copy of all the review comments

Content and Correctness

Reviewer 1:

For the most part, I think that the code notebook was good from a student's perspective. However, I ran into an issue with the training. The notebook mentions that you should switch the processor to GPU, which makes sense, but in the code (not in a section where the student is supposed to code), the authors manually set it to use the CPU. This made the network take forever to train (45 min didn't even pass Epoch 1). I realized this and changed it back to 'cuda' after which it was a lot faster. After training using the solution, though, the validation accuracy was only 0.0307, which was pretty low in my opinion and makes me think that there is either an issue with the model or the parameters.

The written assignment was good too, but I wish it could be a bit longer to cover other topics other than the compound scaling method (maybe about the efficiency methods the authors used in the paper?).

Medium improvement needed

Reviewer 2:

The coding questions seem to be at an appropriate level and are well made. A potential improvement is that there can be more conceptual questions on the pdf file. The homework itself seems to be doable in an hour so an addition of conceptual questions would increase its workload.

Small improvement needed

Reviewer 3:

Both the written and coding questions' answers are correct.

The content of the written questions could be better. The assignment is only two questions which I would recommend adding another question as well to give the person working on your assignment more practice and familiarity with the topic. Make sure to add the commentary file.

Medium improvement needed

Scaffolding

Reviewer 1:

I ran into a few hardships regarding implementing EfficientNet. I feel like the authors could give a little more background/hints on implementing the model, as it was tricky to complete with the information given. I think the rest was really interesting and explained like the dataset, the optimizations, and the visualizations at the end (although maybe create some more for the results itself?).

Small improvement needed

Reviewer 2:

More scaffolding is definitely needed. A little more introduction of concepts would be helpful on the pdf file. For example, an explanation of what EfficientNet or the Compound Scaling Method is. It seems like the pdf file is jumping into questions without giving context. The notebook has more context explanations but hints on the coding part would be more helpful.

Medium improvement needed

Reviewer 3:

I would just make the solutions more in depth, with more details on the answers so that they're more comprehensive.
Also make the second question more clear of where it starts and where the coding portion starts(You can just make it problem 2-4 if you want.) It would also be helpful to explain in the Homework what FLOPs are before asking them to compute them.

Medium improvement needed

Readability/Clarity

Reviewer 1:
Everything that was written was very clear and organized. It was very easy to follow- no complaints. Excellent work, no actions needed.

Reviewer 2:

Some mathematical parts are not formatted correctly on the notebook. Also, (not a big issue) there is a small box next to every solution of question. But overall, everything is clear except for a small formatting issue.

Small improvement needed

Reviewer 3:
The readability and clarity is down well, It would just be helpful to explain in the Homework what FLOPs are before asking them to compute them. And make the written questions more comprehensive and detailed.

Small improvement needed

Commentary on HW

Reviewer 1:
The homework covered the results of dimension alteration and FLOPS very well. I do think that it could have gone a little bit more in depth into some of the other topics covered in the notebook to give the student a better understanding of the paper.

Small improvement needed

Reviewer 2:
Commentary does not exist.

Medium improvement needed

Reviewer 3:
Make the grammatical problem of "What will doubling the resolution do the total FLOPS?" fixed by making it to "What will doubling the resolution do to the total FLOPS?". (fix this for all the questions about FLOPs).

Small improvement needed

Going above and beyond

Reviewer 1:
I really like the work done at the end with the graphviz library. I thought that it was really cool to visualize the Compound Scaling Method. I think for maybe a little more "going above and beyond" the authors could do a little bit more with the actual results of the model itself, with both visualizations and applications of it working on specific images, especially compared to other standard networks.

Small improvement needed

Reviewer 2:

The code sequentially explains EfficientNet, which simplifies the paper well. Visualizing the dataset is also helpful for the flow of the notebook. More extension could improve the quality of the paper though (as I mentioned, the homework is doable in an hour-ish). An idea is to compare EfficientNet with another model or doing an ablation study, to showcase the performance of the model.

Small improvement needed

Reviewer 3:

I believe they spent a lot of time on the code to make it very good.

Small improvement needed

Part II: Point-to-point response to the review comments
We format our response to our reviewers' comments by the category the question they are responding to.

Content and Correctness
In response to reviewer 1's concerns regarding setting to CPU rather than GPU, we have corrected this error for training. We have also changed to CIFAR-10, which should be easier to dealt

In response to reviewer 1's concerns about model performance, we realized that training on imagenet is perhaps too ambitious for such an assignment, and shifted over to CIFAR-10. This makes training much more realistic on a colab GPU, and fits the homework setting of the project much better.

In response to reviewer 1, 2, and 3's comments about the conceptual foundations of our submission, we have extended the written assignment with more conceptual foundations. And moreover to reinforce the general efficiency of the compound scaling method. We have also increased the length of the coding assignment slightly with a training portion, which should extend the overall project to appropriate length. We have also ensured to include this submission file.

Regarding reviewer 1's comments about additional conceptual questions about the leveraged efficiency methods, we have considered this, however, the nature of these methods (squeeze and excitation blocks etc.) makes analytical problems somewhat difficult to pose. Moreover, the overall assignment is about efficentNet, not these specific methods. We still believe that actual construction and application of these blocks to be the most beneficial way of enlightening students about these designs in the context of EfficientNet.

Scaffolding
In response to reviewer 1's comments regarding the difficulty of the coding questions, we have added a suite of tests to ensure that the student is able to check that each portion of their code is working before proceeding. We have also added more lead up wording and hints for these questions. This should make the implementation process smoother and easier for students.

In response to reviewer 1's comments about visualizations about the performance of the model, we have included a section focusing on model performance, including comparisons with pytorch shipped models. This section includes some performance graphs, and we believe this to be a nice way of engaging with the results of the efficientNet architecture.

In response to reviewer 2's comments, we have added some more conceptual lead up questions for the intuition of the compound scaling method.

Regarding reviewer 2's comments about lead up questions for the implementation portion, we added tests and hints to reduce the difficulty of these tasks.

In response to reviewer 2's comments about the lack of clarity regarding the definition of some essential terms, we have stated and labeled the compound scaling method and efficientNet more explicitly.

In response to reviewer 3's comments about FLOPS, we have also restated the definitions for FLOPs and parameters in the written homework for sake of clarity.

In response to reviewer 3's comments about homework lack of clarity for solutions, we added another introductory section to the homework, and some additional led up wording in the questions, which should hopefully make things clearer to students.

In response to reviewer 3's comment about labeling clarity for the questions, we have also added the conceptual questions to the latex file. And we have added some more sectional labels on the notebook. Though we do note that the code will not be able to run without the blocks, and should alert a student to the fact that there are portions of the homework to complete. We also have coding sections marked with:
#TODO: your code here#.
Which should make things clear.

Readability/Clarity

In response to reviewer 3's comments, we have also restated the definitions for FLOPs and parameters in the written homework for sake of clarity.

Commentary on HW

In response to Reviewer 1's comment, we improved the wording of the notebook. However, we do note that some of the tools that EfficientNet leveraged are actually topics of separate papers, and it is a bit beyond scope. We feel like our diagrams and coding questions provide sufficient context for these items regarding the topic of efficientNet.

We have produced a commentary separate from our homework/colab notebooks (this document) in response to Reviewer 2's comment.

We have reviewed and fixed the grammar issues within the written/analytical portion of the homework.

Going Above and Beyond
We appreciate reviewer 1 and 2's comments about our visualizations, and added a single conceptual question to that portion as to ensure students understand the advantages of the method.

Furthermore, regarding Reviewers 1 and 2's comments, we have extended the notebook with a model training and testing portion against several pytorch shipped models. We trained our model along with several others on Cifar, and assessed their performance. We also provided a function to show the number of model parameters (e.g. model efficiency). We believe this will stress the advantages of the efficientNet approach and architectural advantage, whilst also nicely extending the homework's length.

We appreciate reviewer 3's comment on our code quality.

Part III: Final submission

Analytical Problems
From our reading of the EfficientNet paper, we believe that the core intuition of the paper surrounds the intuition that architecture dimensions (depth, width, resolution) should not be independent of each other, and indeed, should be balanced with each other. This, along with the compound scaling method, we believe to be the core contribution of the paper. Given the importance of these core concepts, we prepared several problems which were designed to communicate this intuition.

We first walk students through sketching out the problem of designing a ConvNet in the form of an optimization problem to better prepare our presentation of the compound scaling method. We try to reinforce the conceptual difficulty of scaling up a ConvNet arbitrarily.

Then we focused on the compound scaling method in its optimization problem form. We posed probing questions with regards to how the architecture might change given shifts in the compound scaling factor, and then, how changes in the architecture would affect FLOPs of the model. This is done to show the underlying logic and intuition of the proposed scaling method.

These should be fairly simple questions, but we aimed for them to more or less act as a guided walkthrough of these concepts in lieu of having students actually read the paper.

Architecture Visualization
During our reading of the paper, we noticed that the authors prepared some visualizations illustrating the changes of the network dimensions in certain ways. We found these to be fairly useful. However, we felt that it may be possible to better show the intended effect of the

Compound Scaling Method: namely, allowing the scaling of the entire model by just tuning the scaling factor.

To this end, we created a visualization generator using the python graphviz library, such that we produced a function that would generate a visualization just by the input of the compound scaling factor. We believe, by allowing the students to play with this function directly, we can hammer home the core concepts of the EfficientNet paper.

Building EfficientNet

For the next step in our paper, we are focusing on the actual EfficientNet. The authors designed EfficientNet to be a state-of-the-art ConvNet architecture that leveraged the compound scaling method that they have proposed. We believe the best method of ensuring understanding for these concepts and this proposed architecture is for the student to implement the architecture. EfficientNet implements several efficiency saving methods: squeeze and excitation blocks, stochastic depth and depthwise separable convolution, which we will have students implement as to ensure their understanding of these methods.

We believe walking students through these relatively modern methods leveraged by the paper such as the Squeeze-and-excitation methods by Hu et al. will equip them with a better understanding of the actual model's architecture, and as well as giving them a good understanding of these modern techniques.

We have some basic model training tasks set up questions (dataloaders etc.) as a warm up to the actual implementation.

The first task is to implement squeeze and excitation blocks. These blocks are designed to act similarly as attention mechanisms: for the model to focus on certain features of the data. We produce a squeeze and excitation block nn.Module class for students to implement. We added diagrams to ensure that the students have a smooth experience during the implementation.

The second task is implementing the stochastic depth technique. The technique is to allow the training of short networks and testing with deep networks by randomly bypassing subsets of layers during training time. We also produced a nn.Module class for students to implement, as well as accompanying diagrams.

The third task is implementing Depthwise Separable Convolution, which can intuitively be thought of as depthwise convolution and pointwise convolution at the same time. Again, we do so via a nn.Module block.

Following the implementation of these techniques, we have the students actually build EfficientNet with their assembled 'ingredients', then also introducing back the Compound Scaling Factor ideas in the final construction. To better reinforce the point regarding the ability to scale these models easily, we also provided configs for EfficientNet b0-b7 for the students to

play around with, as to show the agility of scaling up and scaling down EfficientNet to different resource constraints.

Training EfficientNet
We trained EfficientNet on Cifar10 packaged by pytorch. Although originally, the paper used imagenet for assessing capacity, we had also tried to experiment with a chopped down version of imagenet. We realized that properly training and assessing the model's performance on imagenet on a colab environment/for homework purposes is quite impractical.

As a comparison, we pulled down several models from torchvision: densenet121, mobilenetv2, resNet50 and convNext_small. We trained all these models with identical settings (epochs, etc.) and then compared final validation accuracy, and number of model parameters. In our solutions, EfficientNet performed excellently whilst having a very low number of total parameters compared to the other models. This in general shows that EfficientNet is, well, efficient.

We believe having the student actually train their own implemented model, in direct competition to torch shipped models provides a good view of the advantages of EfficientNet.

Part IV: Team member contribution
Ethan Qiu:
Analytical Problems/Conceptual Problems in the LaTeX File
Initial dataloader/cleaning/pipelining (For the original use of ImageNet)
Graphviz Visualization Generation and conceptual questions
Model comparison study
Final Submission Document
Charlie Cheng
Implemented/Refractured the utility functions, training architecture of EfficientNets
Added unit tests for training code
Visualizations of the training data
Documentations for notebook, Implementation markdown guide, hints, etc.
Siqi Zhu
The main architecture of efficient net
Added Unit tests
Yuexi Shen
The initial code of dataloader and training part
Added CIFAR-10 dataset

Part V: Link to code/dataset/supplementary materials

EfficientNet https://arxiv.org/abs/1905.11946
CIFAR10- Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.
https://pytorch.org/vision/main/generated/torchvision.datasets.CIFAR10.html