Do Convolutional Neural Networks Implicitly Count?

Coby Mulliken

Introduction

Convolutional Neural Networks (CNNs) are known to be capable of counting objects at fine resolutions. [1] [2] These algorithms traditionally use a regression framework; that is, a ResNet-style head paired with a scalar-valued output layer. A reasonable question is whether networks trained on categorical—but still quantity-based—tasks like *discrimination* or *classification* also learn, in some sense, 'to count.' We address this question by training ResNet-18 models to choose the majority object on toy datasets consisting of randomly placed and sized circles and squares. We find that although these models perform well on images with similar densities of objects to their training set, they struggle to classify out-of-distribution images, i.e. those with greater or fewer objects per image. Furthermore, these models typically exhibit significant bias in favor of circles or squares (or, in some cases, both directions depending on the object density), and do significantly *worse* than random choice on small-margin differences, despite solid performance overall.

Method

We began by generating several synthetic datasets. Each dataset consisted of 10,000 256×256 grayscale images. Nine datasets were created, each one with its own mean number of shapes per image N_i, with the means ranging from 10 to 90 at increments of 10. Within dataset i, image j, which we will denote x_j, was generated by choosing a total number of shapes $n_j \sim \mathcal{N}(N_i, 3)$, a proportion of circles $c_j \sim \mathcal{N}(0.5, 0.25)$, and then computing the raw numbers of circles and squares by rounding. Shape positions were drawn from a uniform distribution on $[0, 256] \times [0, 256]$. Circle radii were drawn from a a uniform distribution on $[2, 15]$ and square side lengths from drawn from a uniform distribution on $[2, 20]$.

The datasets were split at a ratio of 80/20 into training and validation sets. Nine ResNet-18 models (modified to account for the single color channel input, as opposed to the traditional three) were initialized, and each was trained on a separate dataset. We denote the model trained on the dataset with $N_i = k$ as $\mathcal{M}_k$. The models were trained for 12 epochs, with a batch size of 64 and a learning rate of 0.0002. (This was more or less to convergence, see the figure below; the models were trained on my rather ancient gaming laptop, so compute was limited.) Each model was then evaluated against every testset.

Based on these results, several new test sets were created, intended to evaluate how well the models could discern small changes in margin (that is, small differences between number of circles and number of squares). We created 21 new datasets in total, each of 2,000 images, consisting of each possible combination of margins in the set $\{-8, -4, -2, 0, 2, 4, 8\}$ and image densities in the set $\{10, 50, 100\}$. (The omission of -6 and 6 was not intentional. Oops!)

Results and Discussion

The models attain remarkable accuracy on their own test sets. Accuracies spanned from 60% (achieved by $\mathcal{M}_{90}$ on the $N_2 = 20$ set) to 92% (achieved by many of the models across training

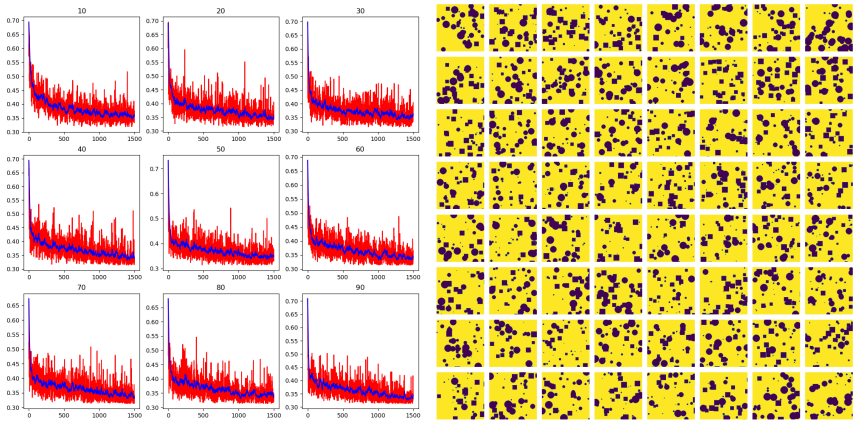

Figure 1: Training loss curves for each of the nine models, and some samples of the training images from the $N_4 = 40$ set. Images are given `viridis` coloring to enhance visibility, but are grayscale in truth.

sets). We note, in general, a pattern whereby models trained on lower densities generalize better to higher densities than vice versa. That said, there is significant variation in model performance across training image density. A reasonable hypothesis (which would require more testing to verify) is that models trained on lower densities must develop more sensitive counting mechanisms since the average margin between the number of circles and the number of squares is smaller, a theory born out by the results shown in Figure 3.

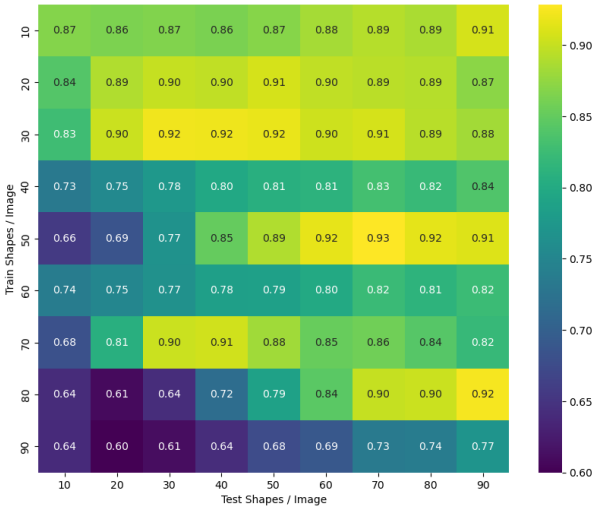

Figure 2: Test accuracies of 9 different models on the 9 different test sets. Models' proper test sets appear on the diagonal.

A peculiar result is that models, in general, place their decision boundaries (i.e. the point at which they reach 50% test accuracy) significantly removed from 50-50 split images; that is to say, for example, that $\mathcal{M}_{50}$ continues to predict that roughly 90% of images have more circles, even when there are 4 more squares than circles. Interestingly, the bias is not unidirectional. $\mathcal{M}_{10}$, for instance, exhibits a similar bias in the opposite direction. Some of this can be ascribed to the fact that ties were coded as a majority for squares. (In all honesty, I made this choice and then realized shortly after beginning the training that you can feed distributions as targets into `nn.CrossEntropyLoss`, which would've been much more reasonable. A task for next time, I suppose.)

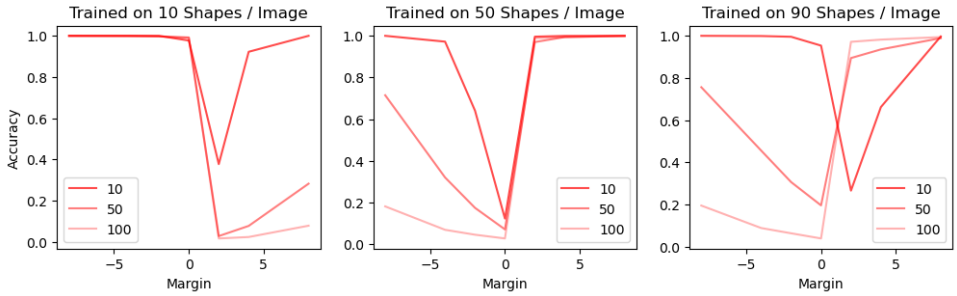

Figure 3: Test accuracies of 3 selected models (trained on $N_i = 10, 50, 90$ respectively) on datasets with various circle-square margins. Negative margins indicate more squares than circles, and vice versa. Lines correspond to datasets with $10, 50$, and 100 shapes per image.

We observe also that, across training densities, models exhibit the least bias on the 10 shape test set. (A rather unsurprising result considering its size.) Correspondingly, the greatest bias occurs on the 100 shape test set, indicating that the *relative* size of the margin is arguably *the* most important factor in determining model capability on a given test set.

Discussion and Further Investigations

This work was conducted over the course of a weekend, and accordingly lead to many more questions than it answered. I'd be most interested in probing the biases observed in Figure 3. Does the categorical framework necessarily encourage these sorts of biases, or could we make tweaks to the architecture and/or training regime to ameliorate them? We might try training a model *solely* on small margin differences. After all, considering that the circle proportion was sampled from a normal distribution with large variance, small margin examples constituted a rather small portion of the training set. Relatedly, what causes single models to develop *different* biases for various densities (i.e. as seen with $\mathcal{M}_{90}$)?

I'd also be interested to see a mechanistic interpretability-style approach to these models. Could we localize counting mechanisms to individual neurons in later layers? And could we identify which filters in convolutional layers detect various shapes at various resolutions? More localized understanding of these *mechanistic* questions might shed light on the bigger picture questions raised above.

References

[1] Jeroen P A Hoekendijk, Benjamin Kellenberger, Geert Aarts, Sophie Brasseur, Suzanne S H Poiesz, and Devis Tuia. Counting using deep learning regression gives value to ecological surveys. *Sci. Rep.*, 11(1):23209, December 2021.

[2] Naveed Ilyas, Ahsan Shahzad, and Kiseon Kim. Convolutional-neural network-based image crowd counting: Review, categorization, analysis, and performance evaluation. *Sensors (Basel)*, 20(1):43, December 2019.