The Current State of the Art in Deep Learning for Image Classification: A Review

Adam Byerly[1,2], Tatiana Kalganova[1], and Richard Ott[3]

[1] Brunel University London
Department of Electronic and Electrical Engineering
Brunel University London
Uxbridge, UB8 3PH UK
abyerly@fsmail.bradley.edu
[2] Bradley University
Department of Computer Science and Information Systems
Peoria, Il, 61615 USA
[3] Air Force Research Laboratory
Sensors Directorate
2241 Avionics Cir, WPAFB, OH, 45433

Abstract. We present a review of the methods behind the top 40 highest accuracies achieved on the ILSVRC 2012 Imagenet validation set as ranked on Papers with Code. A significant proportion of these methods involve using transformer based architectures, but it should be noted that none of the methods are naïve self-attention transformers, which would be unmanageably large if the tokens were derived on a per-pixel basis. Rather, the works we review here all toil with different methods of combining the global nature of self-attention with local nature of fine-grained image features, which have historically been the strength of convolutional neural networks. However, it should be noted that 9 out of 22 works reviewed did NOT use transformers.

Keywords: State of the Art, Imagenet, Papers with Code, Transfomers, Convolutaionl Neural Networks

1 Introduction

In [1], the authors pose the question: "Are we done with ImageNet?". They ask if the recent progress on the Imagenet-1K [2] evaluation benchmark is continuing improvement on generalization or the result of us (the deep learning for image classification community) learning some latent properties of the labeling procedure. The latter possibility is interesting, and in their work they do some good analysis and provide a better set of labels, which we should all consider using going forward. However, for now, the original labels remain the standard benchmark and the means by which comparisons among the best models are made. Papers with Code [3] has become the best known record of the state-of-the-art methods for all types of deep learning tasks, including image classification. On Papers

with Code, in the case of Imagenet, the performance is ranked by top-1 accuracy achieved. In this review, we will examine the technologies behind the top 40 best ranked accuracies, which are reported in 22 papers (some papers present multiple models which rank multiple times).

2 Transformer-Based Networks

Since [4] and later [5], transformer networks have been dominating NLP deep learning tasks. As such, computer vision researchers have been looking into ways to take that success and transfer it to their domain. They have done so with a fair amount of success, with the caveat that such success in most cases has required unprecedentedly large networks with unprecedentedly large sets of additional training data. The fact that in this review we will encounter non-transformer based networks trained without additional training data that are competitive with these networks suggest that it remains an open question whether or not transformer-based networks will entirely supplant convolutional neural networks in computer vision tasks. See Table 1 for a comparison of the transformer-based models reviewed here.

In [10], the authors introduce ViT. ViT is currently the vision transformer network that most recent transformer networks compare themselves to or use as a basis for their designs. Inspired by the success of transformers applied to the NLP domain, the authors endeavored to create a network for the vision domain out of transformers *sans* convolutions entirely, and in their own words "with the fewest possible modifications" to existing transformer designs. The authors note that applying self-attention naively to entire images means attending every pixel to every other pixel and thus represents a quadratic complexity relative to the image's size, which would not scale well to usable input sizes. The insight they leveraged was that 16×16 patches of an image could be treated much like words are treated in NLP applications. Prior attempts at fully transformer based networks [18] failed to achieve competitive results on ImageNet-1k evaluation accuracies due to having not attempted to scale up the networks parameters and additional training data. Again, in their own words, the authors discovered that "large scale training trumps inductive bias"—the inductive bias being that which is introduced by convolutions.

In [6], the authors conducted a systematic study of the relationships between data size, compute budget, and achieved accuracy across a spectrum of ViT models [10]. Unsurprisingly, they discovered that bigger models with larger compute budgets result in higher accuracies, with the caveat that there exists sufficient data to train the model. In the largest models they studied, even 300M samples was insufficient to saturate the models' achievable accuracy. Additionally, they found that the larger models were more *sample efficient*, meaning they achieve the same accuracy as smaller models after training for fewer steps. Another important observation that the authors made was that for more than two orders of magnitude, compute budget and accuracy followed a power-law, and at the high end of the compute budget, the largest models were not tending toward

Table 1. Transformer-Based Networks

Rank	top-1 Acc.	# of Params.	Addt'l. Training Samples	Paper Title
2	90.45%	1843M	3B	Scaling Vision Transformers [6]
4	90.35%	14700M	3B	Scaling Vision with Sparse Mixture of Experts [7]
8	88.87%	460M	300M	TokenLearner: What Can 8 Learned Tokens Do for Images and Videos? [8]
9	88.64%	480M	1.8B	Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision [9]
11	88.55%	632M	300M	An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale [10]
14	88.36%	7200M	300M	Scaling Vision with Sparse Mixture of Experts [7]
15	88.23%	2700M	300M	Scaling Vision with Sparse Mixture of Experts [7]
16	88.08%	656M	300M	Scaling Vision with Sparse Mixture of Experts [7]
18	87.76%	307M	300M	An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale [10]
20	87.54%	928M	14M	Big Transfer (BiT): General Visual Representation Learning [11]
21	87.5%	173M	14M	CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows [12]
22	87.41%	3400M	300M	Scaling Vision with Sparse Mixture of Experts [7]
24	87.3%	197M	14M	Swin Transformer: Hierarchical Vision Transformer using Shifted Windows [13]
26	87.1%	296M	0	VOLO: Vision Outlooker for Visual Recognition [14]
27	86.8%	193M	0	VOLO: Vision Outlooker for Visual Recognition [14]
32	86.5%	356M	0	Going deeper with Image Transformers [15]
35	86.4%	150M	0	All Tokens Matter: Token Labeling for Training Better Vision Transformers [16]
37	86.3%	271M	0	Going deeper with Image Transformers [15]
38	86.3%	307M	0	BEiT: BERT Pre-Training of Image Transformers [17]
39	86.3%	86M	0	VOLO: Vision Outlooker for Visual Recognition [14]
40	86.1%	271M	0	Going deeper with Image Transformers [15]

perfect accuracy, suggesting that a model with infinite capacity would achieve less than perfect accuracy. The authors noted that similar effects have been observed in generative models and the authors of that work referred to this phenomenon as the "irreducible entropy" of the task [19]. This further supports the hypothesis that there is a ceiling on the achievable accuracy for the ILSVRC 2012 Imagenet validation set [1]. They observed a similar saturation at the lower end of the compute budget scale, where smaller models achieved better accuracy than the power-law would predict.

Mixture of Experts (MOE) is a method of combining the outputs of multiple sub-models called *experts* using a *router* mechanism. Generally, these have been studied since the early 1990s [20][21][22]. More recently, they have been applied to computer vision tasks [23]. In [7], the authors endeavored to combine MOEs with transformers. They designed a network that, while containing a large number of parameters, not all parameters get used during inference and they demonstrate the network's ability to achieve competitive results while using as little as half of the computational power available in the network on any sample. Interestingly, the router mechanism they designed doesn't route entire images, but rather individual patches of the images, so that different transformers in the network operate on different patches, possibly of a single image. Additionally, they created a fixed buffer size per expert in their mixture to encourage load balancing among the experts which encourages the overall model not to end up favoring only a small subset of the experts.

The network designed by the authors of [8] is another design based on transformers. Transformers for visual tasks work by splitting the input into patches. The authors noted that in most cases, of the 200–500 patches produced for images of typical training sizes, about 8 or 16 of them were the most informative. They propose a mechanism that the call "TokenLearner" which, prior to the transformer block, learns which patches are significant and passes only those to the transformer. In so doing, they were able to reduce the total number of FLOPs by half and maintain classification accuracy. In addition to the TokenLearner module that precedes the transformer block, they devised a "TokenFuser" module that follows the transformer block which maps the result of the transformer operation back to the input's original spatial resolution, which allows the input and output of the set of operations to maintain the same tensor shape, making them easier to fit into a model's overall architecture.

In [12], the authors grapple with the fact that in transformer based architectures for vision tasks, global self-attention is an extremely expensive operation (quadratic in complexity) compared to local self-attention, which limits interactions between tokens. Their attempt to find a middle option is to introduce what they term a "Cross-Shaped Window" (CSWin), which is an attention mechanism that involves computing self-attention for vertical and horizontal stripes of the input image in parallel. In addition, they introduce a new positional encoding scheme they call "Locally-enhanced Positional Encoding" (LePE), which they claim "handles the local positional information better than existing encoding schemes", "naturally supports arbitrary input resolutions", and is "especially

effective and friendly for downstream tasks". LePE differs from other positional encoding schemes by, rather than being concatenated into the input before the transformer block as with absolute positional encoding (APE) [4] and conditional positional encoding (CPE) [24], moving the encoding inside the encoding block as with relative positional encoding (RPE) [25][13]. But rather than happening inside the SoftMax operation that uses the queries, keys, and values, LePE is applied directly to the values only.

[13] precedes and is cited by [12] and the two papers share an author. The approaches are also quite similar, though the leap from a network of transformers like are present in ViT to what the authors propose in this work is a little more apparent. In this work, the authors note that the spatial position of the patches of the images (the tokens) being used by all layers in ViT are the same. The authors argue that it is better to think of how the patches are divided up as being subject to a window that shifts across the image in subsequent layers. This allows for connections between overlapping regions in the image to be learned by combinations of transformers. This network is trained entirely on publicly available data, using the 14M image ImageNet-22k dataset for additional training data.

The authors of [15] start with a network similar to ViT, consisting of a series of transformer blocks with residual connections between them. They then altered this design in two specific ways. They posit that a problem with the ViT architecture is that the class token being passed along with the image patches through every transformer layer is asking the optimizer to optimize two contradictory objectives. Those objectives being learning the self-attention for the patches and learning the information that leads to the correct classification. In order to combat this, they propose using two different processing stages, the first of which is not passed the class token so that the transformers in this stage can focus solely on learning the self-attention, and only in this stage does the self-attention get updated. In the second stage, the class token is added and the transformers begin learning the classification. Additionally, they added a learnable diagonal matrix they call the "LayerScale" which they multiply the output of a transformer block by before concatenating together with the path that skipped over that transformer block. They refer to this architecture as CaiT (Class-Attention in Image Transformers). This network is trained without using any additional training data.

In [16], the authors propose a method they call "token labelling". The idea behind it is to have each token coming out of a transformer block learn a K-dimensional vector for the classification for that specific patch, where K is the number of classes and the vector components represent the probabilities of that patch belonging to each class. And then for the final classification, these are averaged together across the patches and then combined with the overall image class to form a final prediction. A drawback to this method is that before doing this, each patch's probability for each image must be generated and stored. This network is trained without using any additional training data.

The authors of [17] attempt to take the methods of BERT [5], which are applied to the natural language processing (NLP) domain, and apply them to the vision domain. They call their attempt BEiT. To do this requires a pre-pre-training step that creates discrete token values for each patch of each image via an autoencoder. Then, during pre-training, a transformer-based BEiTEncoder is trained to encode image patches into their corresponding tokens, with the caveat that some of the image patches fed into the network are masked out. Then for the final task of image classification, the pre-trained model has an additional classifier network appended. This network is trained without using any additional training data.

The authors of [14] took note of the fact that all of the best performing transformer based vision models were using large amounts of additional training data to achieve their results. This motivated them to study the use of transformers while training on only the actual Imagenet 1k training data. Their findings were that a major factor in this is the larger parch sizes (typically 16×16 or 14×14) that most transformer architectures use due to their quadratic complexity. The authors posit that this fails to encode sufficiently fine-grained information. Their solution, which at first seems counter-intuitive, is to increase the patch size to 28×28, which for images of size 224×224 means an 8×8 embedding. Then, within each of those patches, use a sliding window attention mechanism to relate the fine-grained information within those patches together. A series of these transformer blocks make up the first stage of their design. The second stage of their design is to split each of those embeddings into 2×2 embeddings of size 14×14 and again apply the sliding window attention mechanism. This network is trained without using any additional training data, and is the highest ranked network to do so.

In their own words, the authors of [11] "aim not to introduce a new component or complexity, but to provide a recipe that uses the minimal number of tricks yet attains excellent performance on many tasks". They refer to this "recipe" as Big Transfer (BiT). In their work, they show that BiT can be pre-trained once and then fine-tuned quite cheaply on the task it is transferred to using a simple heuristic for choosing the hyperparameters for the fine-tuning training. They call this heuristic the "Bit-HyperRule". In their study they found that they could limit the hyperparameters that need fine-tuned to the learning rate schedule and whether or not to use MixUp [26] after transferring. The first step in their heuristic is to categorize the size of the dataset they are transferring to. They class datasets with fewer than 20k labeled examples as *small*, datasets with more than that, but less than 500k labeled examples as *medium*, and everything else as *large*. Then after transfer, for small datasets, they train for 500 steps, for medium, 10,000 steps, and for large 20,000 steps, decaying the learning rate by a factor of 10 after 30%, 60%, and 90% of the training steps. They use MixUp with $\alpha = 0.1$ for medium and large datasets. The network they designed is based on ResNet-v2 [27], but instead of using Batch Normalization (BN), they use Group Normalization (GN) [28] and add Weight Standardization (WS) [29] to all of the convolutions. The authors argue that batch normalization is a poor choice for

transfer learning due to the requirement to update running statistics and show that the combination of GN and WS has a significant positive impact on transfer learning tasks. This network is trained entirely on publicly available data, using the 14M image ImageNet-22k dataset for additional training data.

3 Transformer/Convolution Hybrid Networks

Two of the works we reviewed, including the top ranking design, endeavored to use a combination of transformers and convolutions in their designs. See Table 2 for a comparison of the transformer/convolution hybrid networks reviewed here.

Table 2. Transformer/Convolution Hybrid Networks

Rank	top-1 Acc.	# of Params.	Addt'l. Training Samples	Paper Title
1	90.88%	2440M	3B	CoAtNet: Marrying Convolution and Attention for All Data Sizes [30]
3	90.45%	1470M	3B	CoAtNet: Marrying Convolution and Attention for All Data Sizes [30]
19	87.7%	277M	14M	CvT: Introducing Convolutions to Vision Transformers [31]

The authors of [30] note that convolutional neural networks perform well due to their natural locality bias and tend to generalize well and converge relatively quickly, whereas networks employing transformers perform well because of their ability to find global interactions more easily than CNNs but have been shown to require much more data and many more parameters. In their work, the authors endeavored to combine the benefits of both convolution and attention by summing a global static convolution kernel with the attention matrix prior to the Softmax normalization inside the transformer block's attention heads. They refer to this as *relative attention*. Because the global context required for relative attention has a quadratic complexity with respect to the spatial size of the input, the direct application of relative attention to the raw image is not computationally tractable. Thus, their overall network architecture begins with an initial stem of traditional convolutional operations, which they refer to as stage 0, that down-samples the input image to feature maps half of the original image's size. Then, stage 1 and 2 are Squeeze and Excitation [32] blocks that each further reduce the size of the filter maps by half. It is at this point the filter maps have attained a size that relative attention is able to cope with. As such, stages 3 and 4 are made up of a series of relative attention transformer blocks before the network goes on to a final global pooling and fully connected layer that leads to the output classification probabilities. Residual connections are made between each stage and before the feed-forward network of each transformer block. The authors

pre-trained their networks on Google's internal JFT-3B dataset [6], which as the name implies, consists of 3 billion images. It is worthy of note that training their best performing network took 20.1K TPUv3-core days.

The authors of [31] start with ViT as a basis for their design and then introduce 3 changes. First, at the beginning of each transformer, they introduce what they call a *convolutional token embedding*, which involves reshaping the token sequence going into the transformer back into their 2D spatial positions and performing an overlapping, striding convolution. Then, they replace the linear projection before each self-attention block with what they call "convolutional projection", which uses depth-wise separable convolutions [33] on the 2D-reshaped token map. This replaces the linear projection used by ViT that is applied to the query, key, and value embeddings. Finally, they remove the positional encoding that is usually present in the first stage of a transformer block. The question regarding the necessity of positional encoding in transformers used for vision tasks had been previously raised and studied [24]. Notably, this is the highest rank achieved using less than 300M additional training samples, as well as being the highest ranking design to use a public dataset (Imnagenet-22k) for its additional 14M samples of training data.

4 EfficientNet Networks

4.1 EfficientNetV2: Smaller Models and Faster Training

EfficientNet [34] is a model family that consists of progressively larger models which have been optimized for computation and parameter efficiency using Neural Architecture Search [35], which is a reinforcement learning method that learns the best neural network architecture to use for a given task. See Table 3 for a comparison of the EfficientNet networks reviewed here.

Table 3. EfficientNet Networks

Rank	top-1 Acc.	# of Params.	Addt'l. Training Samples	Paper Title
12	88.5%	480M	300M	Fixing the train-test resolution discrepancy: FixEfficient-Net [36]
23	87.3%	208M	14M	EfficientNetV2: Smaller Models and Faster Training [37]
25	87.1%	66M	300M	Fixing the train-test resolution discrepancy: FixEfficient-Net [36]
28	86.8%	120M	14M	EfficientNetV2: Smaller Models and Faster Training [37]
30	86.7%	43M	300M	Fixing the train-test resolution discrepancy: FixEfficient-Net [36]
34	86.4%	30M	300M	Fixing the train-test resolution discrepancy: FixEfficient-Net [36]

In [37], the authors of the original EfficientNet paper continue their work by introducing EfficientNetV2. In their study, they argue that the scale of regularization needs to be proportional to the original image size of the dataset's images. This includes varying the regularization on a single network design based on the original image size of the dataset it is being trained with. Networks that work with smaller images, should use less regularization, and networks that work with larger images should use more regularization. In their prior work, the authors scaled up the number of layers in every stage of their network by the same factor. In this study, they show that gradually adding additional layers in the later stages is superior. Their prior work achieved the then state-of-the-art top-1 accuracy of 84.4%. This extension to that work achieved 87.3% top-1 accuracy—nearly a 4% absolute improvement.

The work in [36] can appropriately be seen as an extension of their earlier work [38]. In both papers, the authors note that there exists a discrepancy between the prevalent data pre-processing operations during training vs. evaluation. It is common to extract random rectangles from training images and scale them to a certain size each epoch as a form of data augmentation, but during evaluation, the common practice is to choose a central crop of equivalent size. This differing approach during training and evaluation results in varying typical scales of the objects trained on compared to objects of the same class during evaluation, and crucially, unlike with the case of translation, CNNs do not respond to scale differences in a predictable manner. In both works, the authors combat the scale discrepancy by allowing the network to learn how to resize the images during both training and evaluation. The details of the method by which they accomplish this are quite involved and beyond the scope of this paper. The interested reader is referred to the original works. In the first paper, the authors applied their method to ResNet networks and trained only with the 1.2M training images that are a part of the standard Imagenet-1k training set. In the second paper, they applied their method to EfficientNet [34] networks and used the standard Imagenet-1k training set with an additional 300M images for training.

5 Using Neither Transformers Nor Convolutions

A single network using neither transformers nor convolutions ranks among the top 40 state-of-the-art networks we reviewed (see Table 4).

Table 4. Networks Using Neither Transformers Nor Convolutions

Rank	top-1 Acc.	# of Params.	Addt'l. Training Samples	Paper Title
17	87.94%	431M	300M	MLP-Mixer: An all-MLP Architecture for Vision [39]

The authors of [39] begin their introduction with the observation that "As the history of computer vision demonstrates, the availability of larger datasets coupled with increased computational capacity often leads to a paradigm shift". Ironically, their architecture involves avoiding the usage of the canonical paradigm shifting methods of convolutions and transformers and instead is made up entirely of simple multi-layered perceptrons (MLPs). Their architecture uses exclusively matrix multiplication, reshaping and transposition, and scalar nonlinearities. They use two different types of MLP layers. One which works independently on image patches, which "mix" the per-location features, and one which works across patches, which "mix" spatial information. They build their architecture from a series of "Mixer" layers, each of which is made up of each of the two types of "mixer" MLPs, each of which is two fully-connected layers and a GELU [40] nonlinearity. Mixer layers also include residual connections around the mixing sub-layers.

6 Teacher-Student Networks

Using teacher and student networks is arguably more of a training method than a network design. The overarching idea is that the two networks (a) have closely-related but nonetheless different goals or information and (b) either feed from the teacher to the student in a directed manner or to each other in a cyclic manner. See Table 5 for a comparison of the teacher-student networks reviewed here.

Table 5. Teacher-Student Networks

Rank	top-1 Acc.	# of Params.	Addt'l. Training Samples	Paper Title
5	90.2%	480M	300M	Meta Pseudo Labels [41]
6	90%	390M	300M	Meta Pseudo Labels [41]
13	88.4%	480M	300M	Self-training with Noisy Student improves ImageNet classification [42]

Pseudo-labeling [43] involves using a teacher network that generates pseudo-labels on unlabeled data that is fed into a student network in tandem with labeled data. Eventually, the student outperforms the teacher. In [41], the authors extended on this idea, by allowing the teacher to receive feedback from the student and then to adapt. Specifically, how well the student performs on the labeled data is fed back to the teacher as a reward signal for the quality of the pseudo-labels it generated. This surprisingly simple idea leads to the highest ranked design that we reviewed that does not use transformers.

The work presented in [42] is clearly the prior stepping stone that led to [41] reviewed above, as the methods described are quite similar, and the papers share 3 authors. The first key difference in this paper is the attention they pay to the

role of noise in the teacher-student training process, thus the name NoisyStudent. They never inject noise in the teacher model so that when it generates pseudo labels, those labels are as accurate as possible. However, when training the student, they inject considerable noise using RandAugment [44], dropout [45], and stochastic depth [46]. The second key difference is that in this paper rather than having the single student feedback to the single teacher, in this work, the authors follow a self-training framework [47] consisting of three steps. The first step is training the teacher with labeled data. The second step is to generate pseudo labels for unlabeled data with the teacher. The third step is to train the student with a mixture of labeled and pseudo-labeled data. These steps are repeated several times, each time promoting the prior student to be the new teacher and creating a new student model. The authors compare their method to Knowledge Distillation [27], but note that in that work the student was often smaller so that it could infer faster and did not inject noise so aggressively. They say that their method could be thought of as Knowledge Expansion in that the student is larger, with greater capacity and taught in a difficult environment made up of more noise.

7 Innovations Related to Training Procedures

In the remaining works we review, the authors credit their achievement of state-of-the-art results not on the design of the network they used, but rather on other innovations related to the training of the networks (see Table 6).

Table 6. Innovations Related to Training Procedures

Rank	top-1 Acc.	# of Params.	Addt'l. Training Samples	Paper Title
7	89.2%	527M	300M	High-Performance Large-Scale Image Recognition Without Normalization [48]
9	88.64%	480M	1.8B	Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision [9]
10	88.61%	480M	300M	Sharpness-Aware Minimization for Efficiently Improving Generalization [49]
29	86.78%	377M	0	Drawing Multiple Augmentation Samples Per Image During Training Efficiently Decreases Test Error [50]
31	86.5%	438M	0	High-Performance Large-Scale Image Recognition Without Normalization [48]
33	86.45%	255M	0	Drawing Multiple Augmentation Samples Per Image During Training Efficiently Decreases Test Error [50]
36	86.3%	377M	0	High-Performance Large-Scale Image Recognition Without Normalization [48]

In [9] the authors note that there are no publicly available labeled datasets of the order being used by many of the state-of-the-art network designs (e.g. JFT-300M and JFT-3B). This is due in large part to how costly and labor intensive it is to curate such datasets. In their work, they describe a process of downloading 1.8B images accompanied with alt-text from the internet, and rather than doing labor intensive curation, instead opt to only perform a small amount of filtering to the alt-text. Although they don't give a detailed explanation of their filtering process, it would stand to reason that they would filter out words that occurred very infrequently or extremely frequently. After the filtering process, they then have multiple noisy "labels", one per word in the alt-text, per image. For the purposes of this review, we will focus on their efforts to use this data as supplementary data to train a network for validating on the Imagenet-1k validation data. As such, we only briefly mention that prior to doing training for the image classification task, they trained a different model to embed the image and alt-text pairs of their 1.8B image dataset into a shared embedding space where matched pairs were pushed together and unmatched pairs were pushed apart. They then used this embedding to give each of the images' associated alt-text words different weights as labels.

The majority of networks, especially very deep networks like ResNets [51], employ Batch Normalization (BN) [52]. BN has the effect of smoothing the loss landscape which allows for larger learning rates and larger batch sizes. However, BN is a costly operation, behaves differently during training than it does evaluation, and breaks the independence among the training examples in each batch. Furthermore, BN results in a tight coupling of batch size to network performance such that when the batch size is too small, the network performs poorly. The authors of [48] believe that in the long term, reliance on BN will impede progress in neural network research. They noted that by suppressing the scale of the activations on residual branches in ResNets, networks can be trained effectively without BN. Specifically, they propose *Adaptive Gradient Clipping* (AGC) which works by clipping the gradients based on the ratio of gradient norms to parameter norms. The authors note that their work is closely related to recent work studying "normalized optimizers" [53][54][55] which ignore gradient scale and instead choose adaptive learning rates inversely proportional to the gradient norms. They state that "AGC can be interpreted as a relaxation of normalized optimizers, which imposes a maximum update size based on the parameter norm but does not simultaneously impose a lower-bound on the update size or ignore the gradient magnitude".

The authors of [49] point out that with the heavily overparameterized models that are commonly in use, minimizing the training loss, which is the usual goal when training neural networks, can easily result in a suboptimal model. They propose a simple, yet effective approach of not only minimizing the training loss but while doing so, simultaneously minimizing the curvature of the loss landscape in the neighborhood of the loss. Among their other results, notably, they show that when using Sharpness-Aware Minimization (SAM), they achieve robustness to noisy labels "on par with that provided by state-of-the-art procedures that

specifically target learning with noisy labels". In their related work section, they note that similar superior generalization had previously been observed by achieving wider minima, not by explicitly searching for such, but by arriving at it by evaluating on a moving average of the prior training weights [56].

The usual approach to online data augmentation is to draw n samples from the training data, augment each of them with whatever augmentation procedure is being followed and then submit that batch of n augmented images to the training procedure. In [50], similar to earlier work as in [57] and [58], the authors perform a study of the consequences of drawing n samples, augmenting each of them c times and submitting a batch of size cn to be trained. One of their key findings is that for integer values of c greater than 1, higher accuracies were achieved, even in the presence of fixed batch sizes, which means the number of unique images in each batch was fewer. The authors noted that this was especially true in the cases of large batch sizes. The authors state of such models that "despite their superior performance on the test set, large augmentation multiplicities achieve slower convergence on the training set." It is our opinion that it is not "despite" this but at least in part *because* of this. The authors also note that prior work has found that observations of the regularizing effect of large learning rates was proportional to the batch size used [59][60][61]. An interesting hypothesize put forward by the authors is that this observation is a specific case where c is held at one of the more general principle that the regularizing effect of large learning rates is proportional to the number of unique training samples in the batch.

8 Conclusion

In this work we reviewed the 22 papers that elucidate the methods that result in the top 40 accuracies on the ILSVRC 2012 Imagenet validation set as ranked on Papers with Code. An obvious trend is the interest in transformer-based architectures. Additionally, the general trend towards larger and larger model capacities as measured in the number of trainable parameters is readily apparent (see Figure 1).

One thing that could be overlooked, though, is that along with the trend towards increased model capacities there exists the trend toward using more and more additional training data (see Figure 2), the two largest sets of which are not publicly available.

These trends present problems for independent researchers, researchers who are University faculty, and smaller labs. The first such problem is simply the availability of data. The creation of Imagenet represents a turning point in the history of computer vision. Up to that point, dataset sizes were most commonly measured in the 10s of thousands of samples. Imagenet gave us a mega scale dataset and has become the de-facto measure of state of the art as a result. The second problem is the compute power required to train giga scale models on giga scale data. For example, the highest ranked model had to be trained for 20,100 TPUv3-core days. The published price for this much compute is over $300,000

and would take 10 days using the largest TPUv3 pod that exists. On a consumer GPU like an NVIDIA GeForce RTX 3090, it would take approximately 18 years to train this model. As such, state-of-the-art research is now dominated by large corporations like Google, Microsoft, and Facebook.

What can we (the deep learning computer vision community) do to re-democratize the research of state-of-the-art methods? We are certainly not saying that the directions we have been going should be abandoned nor should such research be ceded to large corporations. Instead, we are saying that we should consider scaling up our efforts along an additional vector. That vector being the data we train these models with. We should consider prioritizing the collection of new standard benchmark datasets that fill the gaps between CIFAR-100 and Imagenet and between Imagenet and the internal giga scale datasets of large corporations. And furthermore, we should start researching the quality of the data and develop analytical methods of measuring sufficiency in data quantity as opposed to simply assuming more will always be better.

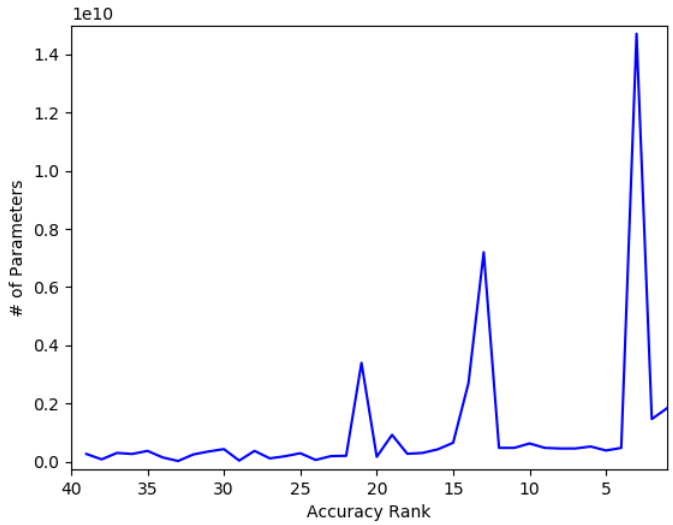

Fig. 1. The number of model parameters used to achieve the top 40 best accuracies on the ILSVRC 2012 Imagenet validation set.

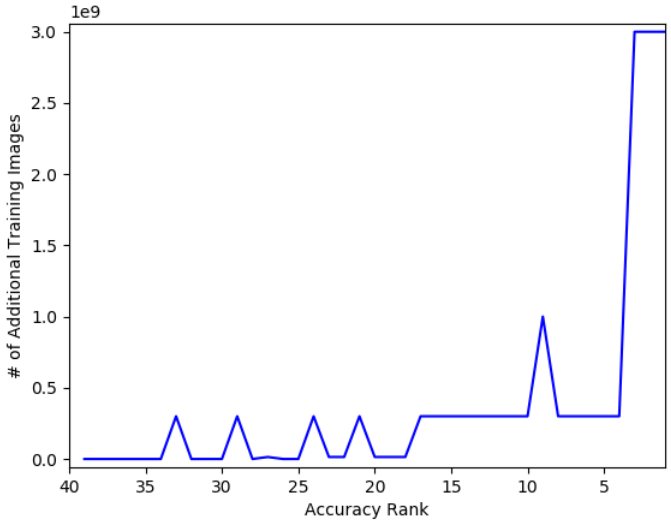

Fig. 2. The amount of extra training data used to achieve the top 40 best accuracies on the ILSVRC 2012 Imagenet validation set.

References

1. Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X. & van den Oord, A. Are we done with ImageNet? http://arxiv.org/abs/2006.07159 (2020).
2. Deng, J. *et al. ImageNet: A large-scale hierarchical image database* in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2010), 248–255. ISBN: 9781424439911.
3. Kardas, M. *et al. AxCell: Automatic Extraction of Results from Machine Learning Papers* in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing* (2020), 8580–8594.
4. Vaswani, A. *et al. Attention is all you need* in *Advances in Neural Information Processing Systems* (2017), 5999–6009.
5. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (2019), 4171–4186.
6. Zhai, X., Kolesnikov, A., Houlsby, N. & Beyer, L. Scaling Vision Transformers. http://arxiv.org/abs/2106.04560 (2021).
7. Riquelme, C. *et al.* Scaling Vision with Sparse Mixture of Experts. http://arxiv.org/abs/2106.05974 (2021).

8. Ryoo, M. S., Piergiovanni, A., Arnab, A., Dehghani, M. & Angelova, A. TokenLearner: What Can 8 Learned Tokens Do for Images and Videos? http://arxiv.org/abs/2106.11297 (2021).
9. Jia, C. *et al. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision* in *Proceedings of the 38th International Conference on Machine Learning (PMLR)* (2021).
10. Dosovitskiy, A. *et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* in *Ninth International Conference on Learning Representations (ICLR)* (2020).
11. Kolesnikov, A. *et al. Big Transfer (BiT): General Visual Representation Learning* in *16th European Conference on Computer Vision* (2020).
12. Dong, X. *et al.* CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows. http://arxiv.org/abs/2107.00652 (2021).
13. Liu, Z. *et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows* in *The International Conference on Computer Vision (ICCV)* (2021).
14. Yuan, L., Hou, Q., Jiang, Z., Feng, J. & Yan, S. VOLO: Vision Outlooker for Visual Recognition. http://arxiv.org/abs/2106.13112 (2021).
15. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G. & Jégou, H. *Going deeper with Image Transformers* in *The International Conference on Computer Vision (ICCV)* (2021).
16. Jiang, Z. *et al.* All Tokens Matter: Token Labeling for Training Better Vision Transformers. http://arxiv.org/abs/2104.10858 (2021).
17. Bao, H., Dong, L. & Wei, F. BEiT: BERT Pre-Training of Image Transformers. http://arxiv.org/abs/2106.08254 (2021).
18. Ramachandran, P. *et al. Stand-alone self-attention in vision models* in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)* (2019).
19. Henighan, T. *et al.* Scaling Laws for Autoregressive Generative Modeling. http://arxiv.org/abs/2010.14701 (2020).
20. Jacobs, R. A., Jordan, M. I., Nowlan, S. J. & Hinton, G. E. Adaptive Mixtures of Local Experts. *Neural Computation* **3,** 79–87. ISSN: 0899-7667 (1991).
21. Jordan, M. & Jacobs, R. *Hierarchical mixtures of experts and the EM algorithm* in *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)* **2** (1993), 1339–1344.
22. Chen, K., Xu, L. & Chi, H. Improved learning algorithms for mixture of experts in multiclass classification. *Neural networks : the official journal of the International Neural Network Society* **12,** 1229–1252. ISSN: 0893-6080 (Nov. 1999).
23. Ahmed, K., Baig, M. H. & Torresani, L. *Network of Experts for Large-Scale Image Categorization* in *Computer Vision – ECCV 2016* (Springer International Publishing, 2016), 516–532. ISBN: 978-3-319-46478-7.
24. Chu, X. *et al.* Conditional Positional Encodings for Vision Transformers. http://arxiv.org/abs/2102.10882 (2021).

25. Shaw, P., Uszkoreit, J. & Vaswani, A. Self-attention with relative position representations. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* **2,** 464–468 (2018).
26. Zhang, H., Cisse, M., Dauphin, Y. N. & Lopez-Paz, D. *Mixup* in *International Conference on Learning Representations (ICLR)* (2018).
27. Hinton, G., Vinyals, O. & Dean, J. Distilling the Knowledge in a Neural Network. http://arxiv.org/abs/1503.02531 (2015).
28. Wu, Y. & He, K. *Group Normalization* in *Proceedings of the European Conference on Computer Vision (ECCV)* (2018). https://research.fb.com/publications/group-normalization/.
29. Qiao, S., Wang, H., Liu, C., Shen, W. & Yuille, A. Micro-Batch Training with Batch-Channel Normalization and Weight Standardization. http://arxiv.org/abs/1903.10520 (2019).
30. Dai, Z., Liu, H., Le, Q. V. & Tan, M. CoAtNet: Marrying Convolution and Attention for All Data Sizes. http://arxiv.org/abs/2106.04803 (2021).
31. Wu, H. *et al. CvT: Introducing Convolutions to Vision Transformers* in *The International Conference on Computer Vision (ICCV)* (2021).
32. Hu, J., Shen, L. & Sun, G. Squeeze-and-Excitation Networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 7132–7141. ISSN: 10636919 (2018).
33. Kaiser, L., Gomez, A. N. & Chollet, F. *Depthwise Separable Convolutions for Neural Machine Translation* in *ICLR 2018 - International Conference on Learning Representations* (2018).
34. Tan, M. & Le, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. http://arxiv.org/abs/1905.11946 (2019).
35. Zoph, B. & Le, Q. V. *Neural architecture search with reinforcement learning* in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (2017).
36. Touvron, H., Vedaldi, A., Douze, M. & Jégou, H. Fixing the train-test resolution discrepancy: FixEfficientNet. *Advances in Neural Information Processing Systems* **32** (2019).
37. Tan, M. & Le, Q. V. EfficientNetV2: Smaller Models and Faster Training. http://arxiv.org/abs/2104.00298 (2021).
38. Touvron, H., Vedaldi, A., Douze, M. & Jégou, H. *Fixing the train-test resolution discrepancy* in *Advances in Neural Information Processing Systems* (2019).
39. Tolstikhin, I. *et al.* MLP-Mixer: An all-MLP Architecture for Vision. http://arxiv.org/abs/2105.01601 (2021).
40. Hendrycks, D. & Gimpel, K. Gaussian Error Linear Units (GELUs). http://arxiv.org/abs/1606.08415 (2016).
41. Pham, H., Dai, Z., Xie, Q., Luong, M.-T. & Le, Q. V. Meta Pseudo Labels. http://arxiv.org/abs/2003.10580 (2020).
42. Xie, Q., Luong, M. T., Hovy, E. & Le, Q. V. *Self-training with noisy student improves imagenet classification* in *Proceedings of the IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition (2020), 10684–10695.
43. Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop: Challenges in Representation Learning* (2013).
44. Cubuk, E. D., Zoph, B., Shlens, J. & Le, Q. V. *Randaugment: Practical automated data augmentation with a reduced search space* in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* (2020), 3008–3017. ISBN: 9781728193601.
45. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* **15,** 1929–1958 (2014).
46. Huang, G., Sun, Y., Liu, Z., Sedra, D. & Weinberger, K. Q. *Deep networks with stochastic depth* in *European Conference on Computer Vision (ECCV)* (2016), 646–661.
47. Scudder, H. J. Probability of Error of Some Adaptive Pattern-Recognition Machines. *IEEE Transactions on Information Theory* **11,** 363–371. ISSN: 15579654 (1965).
48. Brock, A., De, S., Smith, S. L. & Simonyan, K. High-Performance Large-Scale Image Recognition Without Normalization. http://arxiv.org/abs/2102.06171 (2021).
49. Foret, P., Kleiner, A., Mobahi, H. & Neyshabur, B. *Sharpness-Aware Minimization for Efficiently Improving Generalization* in *Ninth International Conference on Learning Representations (ICLR)* (2020).
50. Fort, S., Brock, A., Pascanu, R., De, S. & Smith, S. L. Drawing Multiple Augmentation Samples Per Image During Training Efficiently Decreases Test Error. http://arxiv.org/abs/2105.13343 (2021).
51. He, K., Zhang, X., Ren, S. & Sun, J. *Deep Residual Learning for Image Recognition* in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778.
52. Ioffe, S. & Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* in *Proceedings of the 32nd International Conference on International Conference on Machine Learning* (2015), 448–456. ISBN: 9780874216561.
53. You, Y., Gitman, I. & Ginsburg, B. Large Batch Training of Convolutional Networks. arXiv: 1708.03888. https://arxiv.org/abs/1708.03888 (2017).
54. You, Y. *et al. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes* in *Eighth International Conference on Learning Representations (ICLR)* (2019).
55. Bernstein, J., Vahdat, A., Yue, Y. & Liu, M.-Y. *On the distance between two neural networks and the stability of learning* in *Advances in Neural Information Processing Systems* **33** (2020), 21370–21381.
56. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. & Wilson, A. G. Averaging Weights Leads to Wider Optima and Better Generalization. http://arxiv.org/abs/1803.05407 (2018).

57. Hoffer, E. *et al.* Augment your batch: better training with larger batches. http://arxiv.org/abs/1901.09335 (2019).
58. Choi, D., Passos, A., Shallue, C. J. & Dahl, G. E. Faster Neural Network Training with Data Echoing. http://arxiv.org/abs/1907.05550 (2019).
59. Jastrz bski, S. *et al. Three Factors Influencing Minima in SGD* in *International Conference on Artificial Neural Networks and Machine Learning (ICANN)* (2018).
60. Li, Y., Wei, C. & Ma, T. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in Neural Information Processing Systems* (2019).
61. Smith, S. L., Dherin, B., Barrett, D. G. T. & De, S. *On the Origin of Implicit Regularization in Stochastic Gradient Descent* in *International Conference on Learning Representations (ICLR)* (2021).