High Performance Large Scale Image Recognition Without Normalization

Priyanka Sharma[1], Pratima Sharma[2]

[1]Assistant Professor, Advanced Institute of Technology and Management (AITM), Palwal, Haryana, India
[2]Assistant Professor, Aravali College of Engineering and Management (ACEM), Faridabad, Haryana, India

ABSTRACT
Load standardization is an important aspect of most models, but the usage is focused on a varied range of undesirable features as it depends on the load size and affects the interactions of instances. However, newest profound resent networks are capable of being trained without any standardizing layer, but their precise operation is not the same as standardized batch networks. This will create an adaptive gradient cutting system that fixes these instabilities and develops a much improved Free ResNet class. Our lesser edition is 8.7% faster, and our largest is 87% faster than the top 1.0. The Net-B7 is a versatile Picture Net measuring precision for our smaller versions. As demonstrated in our top models with the accuracy of approximately 90%, Free Normalize models significantly increases the efficiency in fine tuning, compared with large-scale pre-training with 350 million pictures of data-sets.

Keywords: Image Recognition, Normalization**,** ResNet, Tuning, Large-scale Pre-training

1. INTRODUCTION

The majority of recent computer vision models are variations of residual deep networks, with load standardization. Professionals were able to build even deeper networks through the integration of these two architectural innovations that could achieve greater precision in education and science. In terms of quality and regularization, batch standardization often smoothes the losses, allowing for high learning rates and larger batch sizes.
All are single crop numbers in one shape. The simplest one to use with an EffNet-B7 is our NFNet-F1, with the latest Image Net accuracy, the latency of our NFNet-F5 model for training closer to EffNet-B7 is just 87% higher than the top 1. With sharpness-conscious reduction, we increase this further to achieve the top 1 accuracy of 87%.
Yet batch standardization is correlated with three major practical disadvantages. Firstly, the primitive machine is incredibly expensive and requires overall storage and greatly increases the time needed for certain networks to evaluate the gradient. Secondly, this provides a distinction between the model's actions, which allows the balancing of hidden hyper parameters during training. The independence between small-lot training instances is the third and greatest breach of the batch standardization process. For this third property, there are a few negative consequences. For example, practitioners have discovered that standard batch networks on different devices are often difficult to replicate correctly and batch standardization often leads to subtle failures, especially in shared training. Furthermore, batch

standardization should not be used because some of the loss functions can be "cheat" by a network with a single batch interacting between instances. In order to avoid information leakage using such competing learning algorithms, batch standardization, for example, requires specific caution. This is also an important topic in the process of sequence modeling, which culminated in the adoption of language models by alternative officials. The performance of batch-normalized networks can also be decreased when batch statistics vary significantly during training. The performance of batch standardization is inherently sensitive to loading size, and if the load size of the model that is used for finite hardware is too small, standardized batch networks do not perform properly. Thus, while batch standardization in recent years has brought substantial improvements to the learning environment, we expect that progress will be impeded over the long term. We conclude that in a wide range of tasks to achieve good test accuracy, the group should aim to find the simple alternative. While certain other formalizers are recommended, they frequently lead to lower test accuracies, including higher costs of the system and disadvantages to themselves. Luckily, in recent years, two interesting themes have emerged. The first studies the origins of the benefits of batch standardization, while the second studies deep reset for unstandardized competitive accuracy. The measurable magnitude of hidden activations on a residual branch without standardization is one of the key themes in both of these works. Adding a zero-initialized scalar at the end of each branch is the best way to do this. However, to ensure competitive and demanding test accuracy, this trick alone is no longer adequate. Another course of action has shown that ReLU practices implement a 'medium transition' which gradually correlates the occult activation with the growing network depth of different training examples. Recently, during the initialization process, "Normalize Free" was reset to delete the rest of the branch and to apply Scaled Weight Standardization for the minimum adjustment average. With additional regularization, these normalized networks approximate the efficiency of the batch-standard networks of ImageNet but not the most recent efficient networking performance over broad batch sizes. This work is intended to resolve these basic limitations. Our main contributions include the following:

- Gradient adjustment is provided (AGC). It clips graduates by the unit gradient standard ratio to parameters standards and we demonstrate that AGC allows us to develop standard free networks with bigger lot sizes and stronger data gains.
- We develop a family of NFNets with Normaliz-Free Res Networks, which identify a range of latencies with the most recent image net validation accuracy. A new state-of-the-art product without additional 87 percent Top1 data with our NFNet-F1 model of an equivalent precision to the EffectiveNet-B7 model. It is 8.7€ and faster to train our larger model.
- After training on a large private dataset of 350 million labeled frames, we prove that NFNets achieve significantly higher validation accuracy than batch-standardizazed network. Following fine tuning, our best model has reached a top 1 by 90%.

Batch Normalization Understanding

We need to understand the advantages of batch standards throughout training and identify alternative strategies to reclaim those benefits in order to train network for competitive accuracy without standardization. There are descriptions of the four principal advantages identified in past work.

Normalization of batch scales of the branch remaining

With the combination of skis and load connections, networks with tens of thousands of layers can be significantly deeper. Standard batching reduces the activation size of the hidden branch when placed on the other branch (as is typical). This distracts the signal from the path and optimizes well-completed gradients early during the workout.

Normalization of batch eliminates mean-shift

Non-zero median activations such as GELU and ReLU are not symmetrical. Therefore, the product between input functions is close to zero and, after the non-linearity, between activities of separate training examples, the internal product is broad and positive. This complements the network depth by introducing a "mean transformation," in proportion with network depth on all initialization training examples, which can predict deep networks of the same label on a single channel. The standardization of the struggle ensures that each channel has an average activation of zero, removing the mean change.

The regularizing effect of batch normalization

Standardization of lots is also widamente thought to have been regularized by noise in batch statistics computed on a subset of training data and therefore increased test accuracy. This often increases the test accuracy of batch-normalized networks by modifying batch sizes or by standardizing the ghost lot during distributed training.

Standardization batch enables effective comprehensive training

Standardization by batch reduces the error scenario and increases the effective rate of learning. Although the lot is limited, training in wide areas is very important. Although there is no practical benefit to this house. While a major training does not achieve greater test accuracy in a fixed budget, in less parameter changes the test accuracy is achieved and the training speed in comparison with multiple devices is greatly improved.

1.1. Normalization for the elimination of batch

Many authors tried to train deep res networks without standardization in competitive precision by restored one and several of the abovementioned benefits of the batch standardization. For either small constants or learning skalars the activation scale for the rest of the branch is suppressed in the majority of these tasks. With additional power, the efficiency of unused res nets can be improved. However, we can only achieve competitive test accuracy with challenge standards by restore two advantages from batch standardization. In this job, we build and extend a pre-activation course 'Normalize Free ResNets' for competitive training and test precision without layers of standardization. The remaining block of the type is used by NF-Resnet:

$$H_{i+1} = h_{i+\alpha f i} (h_i/\beta_i) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1.1)$$

The remaining block shows the input and the function on the branch of the residual block. The function is to preserve the initialization variances for all Var (fi(z) = Var) (z). The Scalar defines the rate for evaluating a small value like α=0,2 until a residual variance of the block is increased (at first). The standard input difference from its residential blocks can be calculated βi = A complete Var(hi) in which the downscale entry is managed by the skip direction, with the exception of the transformation blocks and the predicted variance can be restored after the transition block hi+1=1+α2. The squeeze-intensive layers have 2 outputs. Empirically, at the end of each remaining branch it was often beneficial to use the learning scale.

1.2. Normalizer-Free Architectures with Improved Accuracy and Training Speed

In the last section, AGC is a gradient selection process that enables us to train efficiently in large quantities and increase the data. We are currently trying to develop standard-free architectures with advanced accuracy and methodological training speed. In the imagery classification, architectural studies based on the inverted Effective Net model family with a backbone strategy usually conduct current state of art. The FLOP and parameters of these models can be optimally evaluated, but their minimal theoretical completeness does not give the modern accelerator higher training rates. EFFNet-B0, although 10 times lower than ResNet-5, has the same latency and efficacy in GPUs or TPUs. In order to select the theoretic FLOP's, subtract latency from an accelerator target system or for training latency, steps should be assessed for different design parameters. We chose to focus on designing models for current latency training accelerators manually. However, we accept that it is important to keep this under consideration (Hooker Europe 2020), as it is easier to speed up science by developing models with improved training speeds on existing hardware so that the most models, including Strong Nets, are likely to make full use of potential accelerators. The accelerators such as TPU and GPU appear to support dense computing, and while variations between the two are obvious, they are sufficiently shared so that a computer model can easily practice. Thus, we analyze the field of model design manually by looking for models that increase parallel to the latency in the top 1. This section deals with the changes that have been so common but not so well described. These changes are controlled and influence overall accuracy. Results occur. Starting with a GELU SE-ResNeXt-D which for standardize-free nets represents a strong cornerstone. These are the changes that we are making. The distance of the group is between 3 to 3 conv and 128, regardless of the block width. Theoretical FLOPS are minimized by smaller group widths, but a decreasing metric density does not allow a large number of modern accelerators. Eight trains at 128 speeds in Group width, for instance TPUv3 SE-ResNeXt-50, except that the lots per unit are 128. This is not due to memory constraints. Second, the backbone model can be upgraded in two different ways. Firstly, the default ResNets pattern for scale-up depths does not consistently increase the second and third phases of layer numbers, while in phases 1 and 4 'stage' maintains three blocks in the same size and resolution activations sequence. This is not the optimal solution, we suppose. Early layers run in higher resolution and require more memory and calculation, and prefer to learn about localized total task properties, while layers operate in later stages at a lower resolution and have the most model parameters. At the early stage, however, it may be too timid, since the model requires a great deal of ability to extract good local features. For deeper versions, a simple scaling rule is also required. With this in mind, we found a range of simple options for the smallest version of the model, F0. F1 is a pattern, for example, and F4 is a profile when a scalar n multiplies the depth of each step. The first point is that the default width pattern in reset networks is reassessed in 256 channels. We considered a number, but with our de-written depth patterns above, only one choice was better than this rule. The breakup of the qualifications is intended in the third stage, as the training rate takes place in the fourth stage. The third stage is normally the best way to monitor the selected resent depth patterns and default settings, which is suppose to be detailed enough for a broad receiving area and a slightly higher resolution hierarchical feature than the second. Also taken into account is the strategy for the remaining block. We have taken many current and new reforms into account, but found that after the first we grouped 3 other alternatives. This further refurbishment does not affect FLOPS or affect the time of training of our accelerators. Finally, for model variants on different calculation budgets a scaling technique is created. The successful strategy of the network scale is to combine the distance, depth and input for the mobile model with slim backbones. However, when the resolution and depth are scaled and the ResNet wide backbones are Bello-compatible with a high degree of efficiency (2021). Through the abovementioned fixed width, scaling depth and scaling resolution of training, we also adopt the later strategies to ensure that any variant can be developed around half the time its predecessor is. The images are measured at approximately 33 percent higher average resolution for every edition. The higher resolution is not quite the same. As this model increases its ability, the intensity of regularization is also helpful. However, the weight or stochastic depth decreases cannot be altered and the rate of decline increases. This is particularly important because our models do not regulate the standardization of batches implied and without accurate regularization are drastically suitable.

2. PROPOSED METHODOLOGY

A updated SE-ResNeXt-D is our NFNet model. The input is a typical HW RGB image with medium/standard variances per channel for most image classification devices from the entire ImageNet training array. The model includes a system 3 GP2 and 16 channels, 2 3 GPS with 32 and 64 channels, and a GPS 3 GPS-2 system with an ultimate 128 channel. A non-linearity between convolutions, but not after the last convolution, is introduced. We use the GELU standard, while most common nonlinearities such as ReLU and SiLU appear to be equally powerful. All our nonlinearity is redesigned to differ roughly with Brock via a fixed scale gain by using our source code. The next step is to complete four residual 'stages' whereby the number of blocks is multiplied by N for our baseline models F0. N=1 is multiplied by N. The other steps begin with a block "transition," followed by the other standard blocks. The switch blocks samples and changes the output channel count in the first step. The use in stages of ski routes convergence of average pooling between 2 and 2 increases efficiency. Found: it is very different from the average phase 2 kernel of Bello (2021) of 3 to3. The resne(x) t bottleneck pattern pre-activates the two blocks with 3 additional blocks grouped into the bottleneck. The main route consists of 1-1 condensed channels with output channels equal to 0.5 per channel block, 2 3 condensed channel 128 and one condensed channel with a final condensed channel of the same number. The final 1:1-cool is followed by a squeeze and ex-cites, which adds a sigmoid to the activation of two linear layers with in linearity, on average world-wide and scales the tensor channel twice that of this sigmoid.
After all the rest, we enforce a transition from 1 to 1 with double channel numbers, close to the final conversion of the growth to the world average productive networks. This layer is especially useful when very thin networks are used because it is typically desirable to surpass or equal the dimensionality of classes to final activation, but we preserve this in a wider network for further work which may aim to construct very slender networks, based on our backbones. After the average pooling, we tried to replace the turret with a fully connected sheet. The last layer is a completely related classification, which results in a 1000-way learning vector. With the standard deviation 0.01 after Goyal, this layer weight is initialized. We found that weight initialization with zero can often lead to multiple-class training instability.
On our residual blocks there are no standardization layers in use. Rather, without standardisation, we use the solution to reduce the gap.
We have also learned a zero-initialized scalar gain, meaning that the remainder is initialized to the identity in order to maximize stability, like in very deep networks. Although it does not inherently follow the predicted variance measured above during initialization, the reduced-down variance is still beneficial for stability.

The scaled Weight Standardization is used to achieve the transition phase in all conversions with learning affinity applied to their normal weight. Standardization is introduced. Crucially, weight decrease or gains of the skip nit will not increase or enhance affinity. S&E layers are not subject to weight nor do they add all related layer weight to their fully attached layers.

3. RESULT DISCUSSION

3.1. ImageNet Experiment Settings

For ImageNet simulations, the ILSVRC2021 training division uses 1281170 images of 1050 classes. Our baseline planning before processing with distorted bordering plants and random horizontal flipping is subject to further increases. We use the 0.1 smoothing mark and optimize our networks with 0.9 energy through the utilization of stochastic gradient descents with categorical soft maximum interentropy losses.

We use standard weight degradation for NFNets with a weight decline coefficient of 2 pp10-5. Critically, the weight loss of SkipInit's normal weight convolution layers or losses does not refer to affinity gains or prejudices. We use a performance rate of 0.25 stochastic deep for all versions for each NFNet version.

The maximum value after Goyal et al is 0.1 to B/256, the batch size. In the first five years, we use a research rate of 0 to its limit (2021). After the learning rate is warmed up, over the rest of the instruction the cosine decay is reduced to zero. AGC shall be used with ̈= 0.01 and = 10-5 for each parameter except the completely connected weight of the linear classifier layer.

Standard training is a common training curriculum for 360 years, with lots size 4096, and the same number of full training courses as training courses for 90 times with lots size 1024. We found that training often improved performance over longer periods but that training was not always consistent between models or settings. We don't stop early.

The average moving average of models with declining rates of 0.9999 is exponential and meets warming schedules when decay is at least equal to (0.9999.1+t/10+t).

To save memory and speed, we are using bfloat16 to train on TPU. The parameters (the momentum buffer) are stored in float 32, but activations and gradients are calculated in float 16 on both sides and back. To improve numerical stability prior to loss measurements, we cast the log into float 32. We cast back to float 32 before summarizing gradients which helps to prevent the build-up of compound errors and ensures the update parameter is calculated in float 32. The usual method of pre-processing one-crop for evaluation, consisting of a 32- pixel row size higher than the target resolution, is adopted. While it is the most common version used, we note that there is an alternate way to take a central padded crop and then adjust the size to the desired resolution. This move is considered marginally worse than the usual pre-harvest option to resize. No time increase, multicrop evaluation or model set has been introduced.

3.2. Measuring Training Latency

The training latency is the wall time observed for the workout on a certain lot size per unit. We finish all 5000 steps and take half the time to complete the preparation. Since the average still requires the first rate rise at the start of the exercise, the average is better immune to these changes and more accurately shows the rate measured during the workout. By running a working loop on already loaded tensors on your computer we can eliminate the data load. This conforms to our NFNets planning, so our data pipeline does not give us any feedback.

We control 32 TPUv3 computers. Each has a batch size of 32 gradients and synthesis gradients in order for our actual pace through distributed training in our training latency to be transmitted. Bfloat16 is used for all models, as described above. In some of the larger models in the 16 GB system memory 32 batch volume per device is not compatible, which means that the compiler is reprocessed automatically. Extra speed can be obtained by careful tuning of manual rematerialisation.

We use float16 to measure the speed of GPU with card tensor cores that significantly accelerate training. Compared with TPUv3 GPU Cross-Device Communication is not considered, as it relies on the hardware for user-accessible connections. Some models like TPUv3 don't have the size of a tone, but use measures to replicate the whole load size. Re-materialization in this case involves manual GPU acceleration but additional engineering work. This appears to be less powerful than reprocessing large models.

We have our own evaluation results and borrowings from Srinivas and others except for SENets, BoTNets and DeIT (2021). Tiny, successful net variants have been also reported with slightly different exercise latencies as we report wall time during the measuring time that neglects interaction between devices. The interface communication costs of very small models can be negligible for the measurement time, in particular for efficient, replicated batch networks. Such hardware costs are typically negligible for major models, such as TPUv3 for fast interconnections. The projected period for BoTNets is also transparent regardless of the reporting process.

3.3. Large Scale Pre-Training Details

The 3500 meters dataset of 350 million marked photographs representative of about 18,100 grades completes our broad pre-training. With the exception of a smaller weight loss with the same optimization as with our ImageNet Evaluation, we prepare all models at resolution 224. We tried to produce more pictures soon and found that increased training costs were not beneficial. We are not using random or exponential increase rates for pre-workouts in basic training other than random crops or flips.

1024 ResNet models load sizes are pre-trained for 10-cycle speeds of 0.4, heated up more than 5,000 steps and reduced to zero for the rest of their testing. ResNet's weight drop and zip depth is balanced for 15 000 strokes on ImageNet with the study rate 0.1 of 2048 batch size. We use the EMA to heat the decrease with a decay of 0.999. We only run a random seed for each model at the cost of this experiment.

However, a substantial loss of weight is not dangerous during pre-training, so that models are no longer limited if they attempt to collect information on a large sequence of data. In this respect, we do not consider Adam as efficient as SGD. We believe that this reflects that, despite similar pretraining and finishing settings, our simple batch-normalized ResNets are well above the reported baselines. The 384-pixel ResNet-50 accident was set to 77.54% with BN-ResNet-50 and 79.9% with NF-ResNet-50. For these ResNet models there are a wide range of accuracies. Instead of beginning with a substantial decrease and attempt to reduce it slightly, potential work on massive pre-paid payments should start with a weight loss of zero.

We schedule a 4096 batch for NFNet versions. NFNet-F4 was developed for 40 epochs and NFNet-F4+ was prepared for 20 years. A wider variant of the F4 model with a channel architecture and associated hyper parameters is included in the F4+ model. The F4 model takes extra time to achieve the same end result as the F4+ model. The F4 model takes about the same latency. This indicates that large models with limited budgets are more efficient than smaller models over a longer period of time in line with the findings (Kaplan et al., 2020).

In 2048 batch size, NFNet models are modified for 15,000 measures, heated by from 0.1 to 5,000, then rinsed null and lowered in further training. A 10-5, 0.25 and stochastic depth rate of 0.1 decrease have increased SAM. For SAM, we are using μ=0.05. We found that with the same regularization as with ResNets we could get similar results, but this mild increase was a little more successful. As with our ResNet tuning, the EMAdecay Warm Parameters are exponentially used and are averaged. In comparison to other models pertained on large datasets, this experiment works.

3.4. Evaluation ImageNet

Starting with our 360-pole 4096 architecture update, now we are studying our ImageNet NFNet modeling. The coefficient of the NESTERVICE engine is 0.9 momentum, the AGC is 0.01 as seen in Section 4, and the training rate of the five epoch's increases from 0 to 1.6 linearly and cosine until it decreases to 0. From the first three lines, we can see that changes in both models only result in minor performance improvements with limited modification of latency. The results of increasing MixUp and R are then calculated. Four layers are using RA and the image resolution is then scaled. This scaling is particularly important to us, since most images are completely blank, as if the image size is too large. For a detailed quantity clarification, please see Appendix A. Data improves efficiency considerably. Finally, we present the results with the normal ResNet stage widths of our complete model, which indicates that during the third and fourth stages our models changed somewhat and resulted in direct comparisons.

We also demonstrate the efficacy of our model architectures while training with NF completeness strategy for batch standardization. In contrast with their NF predecessors, even with highly optimized batch standardize systems without cross-replication synchronization, they are considerably less precisely checked. Moreover, the larger models F4 and F5 were not stable during batch standardization workouts with and without AGC. Big statistics training is required to change these widespread memory models that can lead to numerical imprecision and interfere badly with calculations.

Our models were also supported by the recent sharpness conscious reduction. Since SAM usually just double the workout hours, it is not part of our regular training pipeline. But we have modified a small amount to minimize training costs by 20-40%, using the SAM system for training our two greatest models. The NFNet-F5 ranks in the top 1, 87% and NFNet-F6 in the top 1, 87%. Our efficient net data optimization strategy has also been assessed. While RA significantly improves the performance of the successful networks in relation to the increase of the base lines, it does not increase the production by 2 or the addition of MixUp and CutMix. In addition, instead of RMSProp, SGD degrades performance and not gradual decay. The option of another optimizer not only leads to efficiency changes.

3.5. Evaluating NFNets under Transfer

Unnormalized networks would not share the implicated impact of batch normalization. They seem over compatible with data sets like ImageNet, even if expressly treated. Such regulations cannot, however, be just inappropriate but can harm pre-training output in very large data sets and limit the ability of the model to concentrate entirely on training selection. We feel that learning transmission through a standardization-free network can be enhanced after a broad pre-service process and that 350 million images are monitored in a wide range of data. We prepare a set of standard lots and NF-ResNets in this large 10-phased data array with 15,000 steps for cosine adjustment in all levels with 2048 and 0.1 batch Images Nets. The pictures are simultaneously available. The regular batch counters of the normalization-free networks reach a total margin of about 1 percent. The final implementation of the transfer training method is directly helped by eliminating tones of standardization. Our NFNet versions, NFNet F4 pre-working and a wider range are used in this same experiment called N FNet-F4. Our NFNet-F4+ provides total precision with ImageNet 90 percent in 20 pre-workout cycles. This is the second truest additional schooling and highest accuracy to date since only secondary education can become fresh semi-finished learning.

4. CONCLUSION

First of all, we demonstrate the potential for image recognition models in large, standard free trained data sets of the best organized batch models, and significantly surpass the accuracy of classification. For this function, we use Adaptive Gradient Clipping, a simple clipping algorithm to stabilize long-lasting training and to optimize high-data networks. In this technique and basic architectural principles, we build a family of models which can be trained much faster than competitive approaches at the latest stage of net image development. We also show that standardize free models are suitable for fine tuning after workouts in very large data sets with regard to their usual batch components.

REFERENCES

1. Gitman, I. and Ginsburg, B. (2017). *Comparison of batch normalization and weight normalization algorithms for the large-scale image classification.* arXiv preprint arXiv: 1709.08145.
2. He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep residual learning for image recognition.* In CVPR.
3. De, S. and Smith, S. (2020). *Batch normalization biases residual blocks towards the identity function in deep networks.* Advances in Neural Information Processing Systems, 33.
4. B. Hariharan and R. Girshick. (2016). *Low-shot visual recognition by shrinking and hallucinating features.* arXiv preprint arXiv: 1606.02819.
5. Dosovitskiy, A., et al. (2021). *An image is worth 16x16 words: Transformers for image recognition at scale.* In 9th International Conference on Learning Representations, ICLR.
6. Y. Guo and L. Zhang. (2017). *One-shot face recognition by promoting underrepresented classes.* arXiv preprint arXiv: 1707.05574.
7. Kavukcuoglu, K., Ranzato, M. A., and LeCun, Y. (2010). *Fast inference in sparse coding algorithms with applications to object recognition.* arXiv: 1010.3467.
8. F. Schroff, D. Kalenichenko, and J. Philbin. (2015). *Face net: A unified embedding for face recognition and clustering.* In IEEE Conference on Computer Vision and Pattern Recognition, pages 815–823.
9. Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). *Layer normalization.* arXiv preprint arXiv: 1607.06450.
10. Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). *Image net classification with deep convolution neural networks.* In NIPS.
11. Larochelle, H. and Bengio, Y. (2008). *Classification using discriminative restricted boltzmann machines.* In ICML.
12. He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). *Momentum contrast for unsupervised visual representation learning.* In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738.
13. Simonyan, K. and Zisserman, A. (2015). *Very deep convolution networks for large-scale image recognition.* In ICLR.

14. L. Wolf, T. Hassner, and I. Maoz. (2011). *Face recognition in unconstrained videos with matched background similarity.* In IEEE Conference on Computer Vision and Pattern Recognition, pages 529–534.
15. He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep residual learning for image recognition.* In CVPR.
16. Mahendran, A. and Vedaldi, A. (2015). *Understanding deep image representations by inverting them.* In CVPR.
17. He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., and Li, M. (2019). *Bag of tricks for image classification with convolution neural networks.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 558–567.
18. Ranzato, M. A. and Szummer, M. (2008). *Semi-supervised learning of compact document representations with deep net-works.* In ICML.
19. D. Yi, Z. Lei, S. Liao, and S. Z. Li. (2014). *Learning faces representation from scratch.* arXiv preprint arXiv: 1411.7923, 2014.
20. Bello, I. (2021). *Lambda networks: Modeling long-range interactions without attention.* In International Conference on Learning Representations ICLR, 2021. URL. https://openreview.net/forum?id=xTJEN-ggl1b
21. Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). *A simple framework for contrastive learning of visual representations.* In International conference on machine learning, pp. 1597–1607. PMLR.
22. Valpola, H. (2015). *From neural PCA to deep unsupervised learning.* In Advances in Independent Component Analysis and Learning Machines (Chapter 8), pp. 143 – 171.
23. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). *Image net: A large-scale hierarchical image database.* In CVPR.
24. Torralba, A., Fergus, R., and Freeman, W. (2008). *80 million tiny images: A large data set for nonparametric object and scene recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(11):1958–1970.
25. Hoffer, E., Hubara, I., and Soudry, D. (2017). *Train longer, generalize better: closing the generalization gap in large batch training of neural networks.* In Advances in Neural Information Processing Systems, pp. 1731–1741.
26. Wang, X. and Gupta, A. (2015). *Unsupervised learning of visual representations using videos.* In ICCV.
27. Bjorck, N., Gomes, C. P., Selman, B., and Weinberger, K. Q. (2018). *Understanding batch normalization.* In Advances in Neural Information Processing Systems, pp. 7694–7705.
28. Dosovitskiy, A. and Brox, T. (2016). *Inverting visual representations with convolution networks.* CVPR.
29. Huang, L., Qin, J., Zhou, Y., Zhu, F., Liu, L., and Shao, L. (2020). *Normalization techniques in training dnns: Methodology, analysis and application.* arXiv preprintarXiv: 2009.12836.
30. Arpit, D., Zhou, Y., Kota, B., and Govindaraju, V. (2016). *Normalization propagation: A parametric technique for removing internal covariate shift in deep networks.* In International Conference on Machine Learning, pp. 1168–1176.