Self Supervised learning in image classification: Introducing Barlow Twins

Outliers
Nilabja Ghosh, Samrat Ghosh, Udvas Das
nilabjaghosh2001@gmail.com
samratghosh080@gmail.com
udvasdas7399@gmail.com

June 30, 2022

Abstract

In the last few years, we have seen tremendous progress in the AI field. With the help of massive amounts of labelled data, we can train our AI models to do the task they are trained to. But there is obviously a limit to which the AI field can go with supervised learning alone. Sometimes, the data we have in our hands may not have the actual labels or the labels it has, cannot be fully trusted. Then supervised models become totally useless. This is where self-supervised comes into existence. This is a method of machine learning which learns from unlabeled data. It can be considered as a middle ground between supervised and unsupervised learning. A successful approach to SSL(Self Supervised learning) is to learn embeddings that are invariant to distortions of the input sample. However, a common problem faced in this approach is the existence of trivial constant or collapsed solutions. There are a number of self-supervised models in AI which are already available in the market, like BYOL, SimSiam, SimCLR, SwAV which help us solve many real-life problems and use complicated methods like gradient stopping, Predictor network or moving average on the weight updates etc. Also, they use two asymmetric networks to process the two distortions of the input sample. The name Barlow Twins came from the 'redundancy principle' coined by the famous neuroscientist David Horace Barlow. It avoids this collapse by finding the cross-correlation matrix between the outputs of two identical networks fed with distorted versions of an input sample, making it as close to the identity matrix as possible. This results in the embedding vectors being similar, reducing the redundancy between them. Barlow twins do not require large batches of data and they generally benefit from high-dimension output data. In this project work titled "Self Supervised learning in image classification: Introducing Barlow Twins" we mainly focus on how our model and other state-of-art supervised models handle a dataset with fewer data in each class.

1 Introduction

The very concept of Self Supervised learning emerged when annotations or ground truth of a dataset is missing or cannot be trusted, for example, medical images per se. So, it primarily aims to learn representations of the input data without relying on human annotations. Another importance of this type of learning is that unlike supervised learning it can know meaningful representations even if the size of input data is not very large. It can be explained by the fact that the model tries to learn the representations from the distorted inputs, which helps us to get good generalization error and the learning becomes invariant to distortion of inputs. But this type of learning also has some major setbacks. One of them is the collapse problem or getting a constant representation of the input data. Many state-of-art SSL models which are readily available have different approaches and strategies to tackle this collapse problem. Even contrastive and non-contrastive methods handle this problem in different ways. For example, contrastive methods like SIMCLR define its positive and negative sample pairs and treat them in different ways in the loss function, followed by an alternate optimization scheme like k-means in DEEPCLUSTER or non-differentiable operators in SWAV and SELA; clustering methods use on distorted input to compute targets and another distorted input is used to predict this target. There is also a standard practice that surfaced in recent years to introduce asymmetry in the predictor network by training the parameters using only one distorted version of the input and representations learned from the other distorted input are used as a fixed target. Also, it has been shown in recent work that a 'stop-gradient' is necessary for avoiding the trivial solution. As opposed to the previous approach of asymmetric networks and alternate optimization schemes, in this project work, we use the method BARLOW TWINS, a non-contrastive and fairly simpler self supervised model which works on the principle of redundancy reduction. This model uses a loss function which incorporates the useful information as well as the information which may seem redundant. This can be logically explained by the principle of human vision. The human brain needs way less visual information about an object than a machine to distinguish it from other objects or predict it. This is simply because usually, a machine extracts only the most useful information(Standard practice in bottleneck encoders) of features, while the human brain processes all the features including the features which are considered redundant by a machine. The model takes two distorted or augmented versions of input and processes them via similar network architecture, that is how we get the two embeddings, popularly called "Twin embeddings". Then, we calculate the cross-correlation matrix using these two embeddings, which is nothing but the cosine similarity and we try to make it as close as possible to an identity matrix using our loss function. It Learns useful representation even with a small amount of data and is very much benefitted by higher dimensional embeddings, without the help of any asymmetry, gradient stopping, momentum encoders, non-differentiable operators etc.

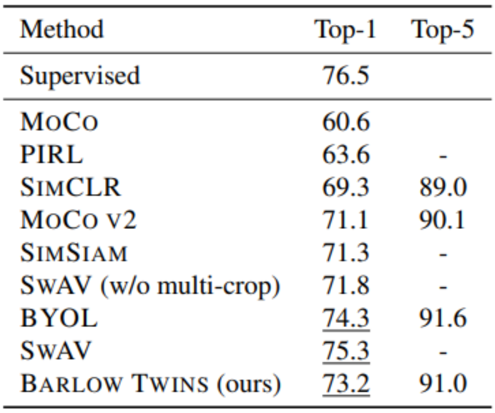
2 Literature review

In the code, the image augmentation pipeline consists of random flipping, resizing, horizontal flipping, colour jittering, gray-scale conversion, Gaussian blur and colourisation. The first two transforms are always applied while the rest are applied randomly with some probability.

The encoder used is a ResNet-50 followed by a projected network. The projector network has three linear layers, each with 8192 output units. The first two layers of the projector are followed by a batch normalization layer and rectified linear units. For optimising the parameters, the model is trained for 1000 epochs with a batch size of 2048 and a learning rate warm-up period of 10 epochs is used.

The network in the original paper is pre-trained using self-supervised learning on the training set of the Image-Net ILSVRC-2012 data-set. A linear evaluation is then used on Image-Net on top of the representations of the ResNet-50 pre-trained within the Barlow Twin method. It outperforms the previous methods on Image-net in semi-supervised classification in the low-data regime. Taking the whole data, the model gives us an accuracy of 73.2% compared to 76.5% in the case of supervised learning classification **Figure 1**.

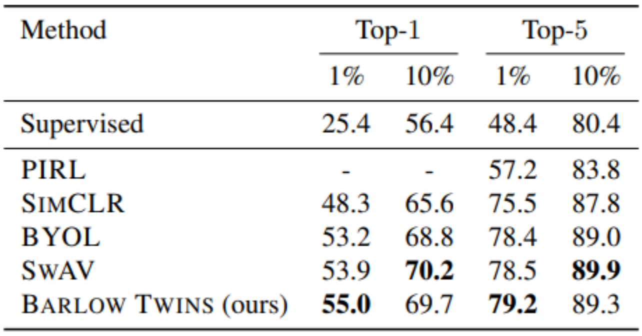
The pre-trained ResNet-50 is finetuned with a subset of images from ImageNet. For the subsets, 1% and 10%, the split used is the same as SimCLR.For 1% data, BarLow Twin has the highest accuracy among all the competing models and for 10% data, it is on par with state-of-the-art model SwAV.



Method	Top-1	Top-5
Supervised	76.5	
MoCo	60.6	
PIRL	63.6	-
SimCLR	69.3	89.0
MoCo v2	71.1	90.1
SimSiam	71.3	-
SwAV (w/o multi-crop)	71.8	-
BYOL	74.3	91.6
SwAV	75.3	-
Barlow Twins (ours)	73.2	91.0

Figure 1: Top-1 and top-5 accuracies (in %) under linear evaluation on ImageNet. All models use a ResNet-50 encoder. Top-3 best self-supervised methods are underlined. [7]

In our project work instead of Imagenet we have incorporated the CIFAR-10 dataset and worked with 10% and 20% fractions of the whole set sampled using balanced sampling. We use the similar architecture for training, i.e ResNet-50 without the last classifier layer but in case of the projection head we choose a much simpler network with hidden and output dimension of 8192. Also, we have used some different augmenta-



Method	Top-1		Top-5	
	1%	10%	1%	10%
Supervised	25.4	56.4	48.4	80.4
PIRL	-	-	57.2	83.8
SimCLR	48.3	65.6	75.5	87.8
BYOL	53.2	68.8	78.4	89.0
SwAV	53.9	**70.2**	78.5	**89.9**
Barlow Twins (ours)	**55.0**	69.7	**79.2**	89.3

Figure 2: Semi-supervised learning on ImageNet using 1% and 10% training examples. Best results are in bold. [7]

tions from the previous work on imagenet which are shown to perform better in case of CIFAR-10. Also, we have used ResNet-50 in a supervised framework and other state-of-art supervised machine learning models to compare our model's performance. Our main goal here would be to see whether our model works as fine without human annotation setup as with annotation setup in case of supervised framework.

3 Proposed methodology

In our project, we are trying to see the effectiveness of Barlow Twins against other supervised methods when fewer data points are used to train the model.

UNDERSTANDING BARLOW TWINS ALGORITHM: Barlow Twin operates on joint embeddings of distorted images. It produces two distorted views of all images in batch X. This distortion is obtained by applying an ensemble of data augmentations. Then the distorted inputs are processed through two symmetric architecture f_θ (Here it is ResNet-50 without the last classifier layer) and we get the twin embeddings Z^A and Z^B.

After we got the embeddings, we fed those into the BARLOW TWINS loss function and what makes BARLOW TWINS different from the other models is the loss function-

$$\mathcal{L}_{\mathcal{BT}} \triangleq \sum_i (1 - \mathcal{C}_{ii})^2 + \lambda \sum_i \sum_{j \neq i} \mathcal{C}_{ij}^2$$

where λ is the positive constant and C_{ij} is the cross-correlation matrix between the two output of the two networks and it is calculated by-

$$\mathcal{C}_{ij} \triangleq \frac{\sum_b Z^A_{b,i} Z^B_{b,j}}{\sqrt{\sum_b \left(Z^A_{b,i}\right)^2} \sqrt{\sum_b \left(Z^B_{b,j}\right)^2}}$$

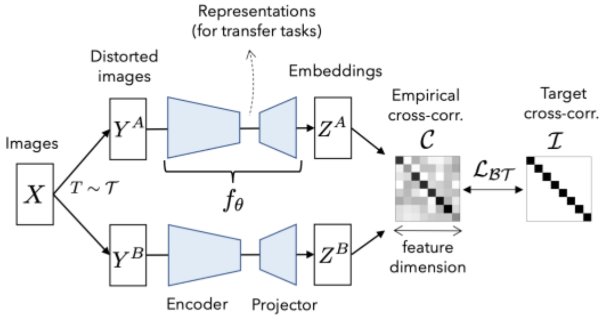


Figure 3: Barlow Twin Model [7]

where b indexes batch samples and i, j index the vector dimension of the networks' outputs. C is a square matrix with the dimension of the network's output, and with values between -1 (i.e. perfect anti-correlation) and 1 (i.e. perfect correlation). If we look closely at the loss function we notice that the first term involves the diagonal elements which represents similarity measure and the second term consists of off-diagonal terms which represents dissimilarity measure and does the redundancy reduction task for us. It tells the machine that we need to not only draw the similar representations close, but also we need to make sure the dissimilarity between two different representations is not close at all.
The pseudo-code for the BARLOW TWINS model is given in figure 4.

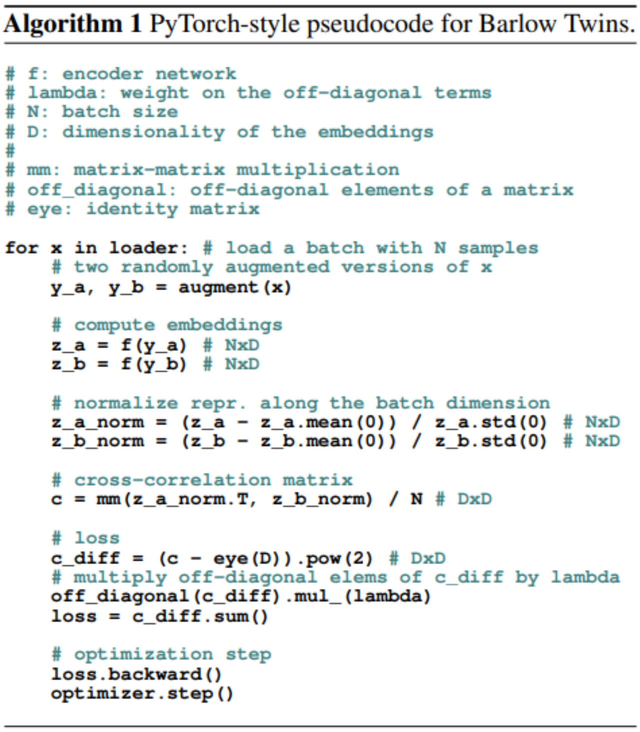
4 Implementation details

You should evaluate your proposed method with standard/state-of-the-art data-sets. In this section you should describe the following:

- **Dataset**: We consider a state-of-art dataset CIFAR-10, which consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. We will use 10% and 20% of this dataset to test our results.

- **Sampling**: In this project, balanced sampling for choosing the fractions of the dataset so that there is no class imbalance problem. Therefore our 10% fraction contains 6000 images of the same dimension in 10 classes, with 600 images per class and 20% contains 12000 images consisting of 1200 images from each class.

- **Image Augmentation**: Two distorted versions of the same image input are made and fed into the symmetric network architecture in our model. The different types of augmentations used sequentially in our model are random cropping, left to right random horizontal flip, colour jittering or random grayscaling, Gaussian blurring.



Algorithm 1 PyTorch-style pseudocode for Barlow Twins.

```
# f: encoder network
# lambda: weight on the off-diagonal terms
# N: batch size
# D: dimensionality of the embeddings
#
# mm: matrix-matrix multiplication
# off_diagonal: off-diagonal elements of a matrix
# eye: identity matrix

for x in loader: # load a batch with N samples
    # two randomly augmented versions of x
    y_a, y_b = augment(x)

    # compute embeddings
    z_a = f(y_a) # NxD
    z_b = f(y_b) # NxD

    # normalize repr. along the batch dimension
    z_a_norm = (z_a - z_a.mean(0)) / z_a.std(0) # NxD
    z_b_norm = (z_b - z_b.mean(0)) / z_b.std(0) # NxD

    # cross-correlation matrix
    c = mm(z_a_norm.T, z_b_norm) / N # DxD

    # loss
    c_diff = (c - eye(D)).pow(2) # DxD
    # multiply off-diagonal elems of c_diff by lambda
    off_diagonal(c_diff).mul_(lambda)
    loss = c_diff.sum()

    # optimization step
    loss.backward()
    optimizer.step()
```

Figure 4: Barlow Twin Pseudo-code. [7]

The first and last transformations are always used, whereas the other three are applied with respective probabilities 0.5, 0.8 and 0.2.

- **Network Architecture**: We use the same architecture which consists of ResNet-50 network without the last classification layer. It gives a fixed output dimension of 2048. It is followed by a projection head which has a hidden dimension and output dimension of 8192. We use higher dimensional embeddings for our model because it highly benefits from it. Though in the previous work, it is shown that a deeper projection head gives better results, we are going for a much simpler head keeping in mind the size of the data and resource constraints. The outputs from the projector head are basically the embeddings which are fed to the loss function of BARLOW TWINS.

- **Training and Optimization (Hyper-parameters)**: After we have got the embeddings out from the projector network, using the loss function mentioned earlier we train the model for 1000 epochs for 10% of CIFAR-10 and 450 epochs for 20% of CIFAR-10. We have chosen a batch size of 64, much smaller than the batch size used in the case of Imagenet because the amount of the training data is way less here. We have used CosineAnnealingWarmRestarts as our learning rate scheduler with a warm rate after 30 epochs.

5 Results

	Test Accuracy (%)			
Model	10 %(Micro)	20 % (Micro)	10%(Macro)	20 % (Macro)
KNN	28(k=5)	28.52(k=5)	24.1(k=5)	27.01(k=5)
Logistic	33.25	37	32.67	36.66
Random forest	38.72	41.04	36.86	40.45
Resnet-50	13.68	16.39	7.03	10.59
Barlow Twins	**66.44**	**69.14**	**66.31**	**69.02**

Table 1: Summary of Testing Accuracy

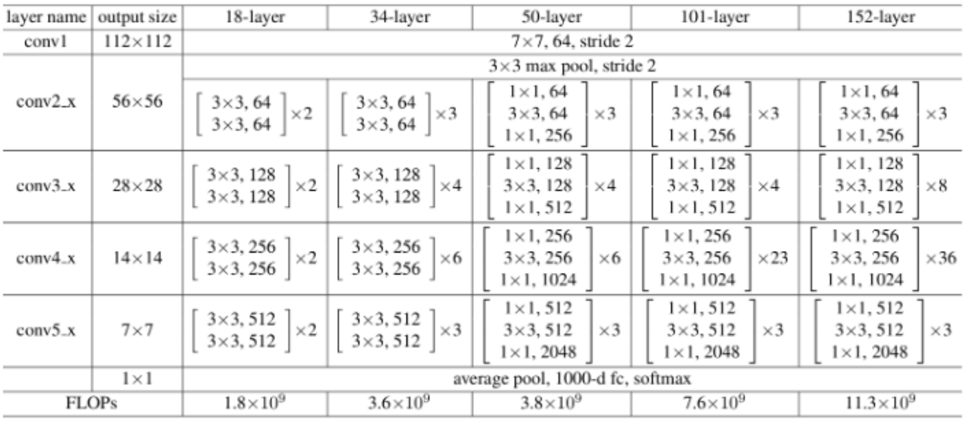
6 Pre-requisites

Here are some concepts which can be considered as prerequisites to read this report,

1. **Contrastive and Non-Contrastive Learning**: **Contrastive learning** is a machine learning approach to finding similar and dissimilar information from a dataset for an algorithm. It is also a classification algorithm where the data is classified based on similarity and dissimilarity. Contrastive methods learn representations by contrasting positive and negative examples. Past research has proved a great empirical success in computer vision tasks using contrastive pre-training. The contrastive method learns representations by minimising the distance between two views of the same data point and maximising views from different data points. Essentially, it minimises the distance between positive data to a minimum and maximises the distance between negative data to a maximum. **The Non-Contrastive approach** uses an extra predictor and a stop-gradient operation. Two popular non-contrastive methods, BYOL and SimSiam, have proved the need for the predictor and stop-gradient in preventing a representational collapse in the model. Unlike contrastive, the non-contrastive approach is simpler, based on optimising a CNN to extract similar feature vectors for similar images. They learn representations by minimising the distance between two views of the same image.

2. **ResNet-50 Architecture**: ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has $3.8 * 10^9$ Floating points operations. It is a widely used ResNet model. ResNet architecture is shown in figure 5.

3. **Cross-Correlation Matrix**: The cross-correlation matrix of two random vectors is a matrix containing as elements the cross-correlations of all pairs of elements of the random vectors. The cross-correlation matrix is used in various digital signal processing algorithms.



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 5: ResNet50 architecture. [6]

4. **Image Augmentation**: Image augmentation artificially creates training images through different ways of processing or combination of multiple processing, such as random rotation, shifts, shear and flips, etc.

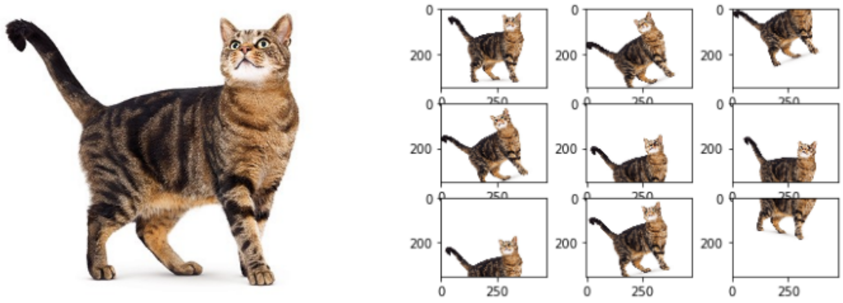


Figure 6: Example of Image Augmentation. [2]

7 Summary

Throughout this project work, we have derived some interesting results.

- Whether we are working with 10% or 20% of the data, BARLOW TWINS consistently outperforms all the state-of-art supervised frameworks we have used.
- In case of using 10% fraction, though the validation accuracy on 1000 samples gives around 45%accuracy for ResNet-50, it drastically reduces when we test the model in the whole test set of size 10000. It can be explained by the fact that when the training sample is fairly small compared to the testing set, a complex architecture

like ResNet-50 learns the representation quite well based on the training set only. But when it is tested on the whole testing set the sample deviates largely from the training set and the learned representations do not seem to work properly anymore.

- In case of using 20% fraction also, validation accuracy is 56.01% and it again reduces drastically. This can be explained by the very reason mentioned earlier.
- Another interesting thing that we noticed is that if we feed augmented image inputs into the ResNet-50 shows around 61% accuracy in case of 10% samples and shows around 69% accuracy. This phenomenon can be explained by the fact that when we are perturbing the input images and then feeding those into the network, the representation that it learns shows a lot better generalization error and performs well on the whole test dataset of size 10000.

8 Future work

This project work opens up a spectrum of options for future work. The idea of Barlow Twins is very new and can be incorporated in many other fields. We suggest a few possible works that can be done further,

1. Since the concept is very new it has not been incorporated in the field of NLP. This field is very famous for having datasets with very little data and also they often show class imbalance problems. For example in case of prescription data, there is no annotation and class imbalance is evident.
2. Datasets in the medical domain also have the same problem mentioned above. So incorporating the idea of Barlow Twins can be a good strategy.
3. Barlow twins' performance can be on par with state-of-art semi-supervised and unsupervised models. So, it will be interesting to see how the other unsupervised and semi-supervised models perform when the amount of data is little.

9 Suggestions during presentation

1. Since the working principle is somewhat similar for both Barlow Twins model and Autoencoders, we were suggested by our instructor to fit an autoencoder using ResNet-50 as encoder and observe how it performs.
2. We were happy to see interest from our batchmates on the problem of dimensional collapse often faced in self-supervised framework. So, we wish to delve deeper into the theoretical background of this well known problem. We are citing a paper in the reference [5] on the very topic for those who are interested can go through.

References

[1] Github link of our code. https://github.com/udvasdas/Self-Supervised-learning-in-image-classification-Introducing-Barlow-Twins, 2022.

[2] Rohit Dwivedi. Image augmentation. https://analyticsindiamag.com/image-data-augmentation-impacts-performance-of-image-classification-with-codes/, 2020.

[3] Facebook. Barlowtwins. https://github.com/facebookresearch/barlowtwins, 2021.

[4] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019.

[5] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[6] Aakash Kaushik. Resnet-50. https://iq.opengenus.org/resnet50-architecture/.

[7] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.