Self-Supervised Contrastive Learning with In-Domain and Out-of-Domain Image Classification

Murali M K Dandu
MS in MLDS, ECE
A59004607

Srinivas Rao Daru
MS in SIP, ECE
A59003596

Abstract

The problem with supervised pre-training is the amount of labelled data they need which consumes a lot of manual time. These models also may not fully learn generalized visual representations for all downstream tasks. In this project, we have implemented a Self-Supervised Learning (SSL) technique called SimCLR, a simple contrastive visual representations learning framework. We showed better convergence under a newly created data augmentation strategy. Few-label scenario experiments show that SimCLR can be used an effective semi-supervised learning technique. We tested the transfer learning robustness using couple of in-domain and out-of-domain datasets. The results indicate that SimCLR can be used as common feature extractor for downstream tasks and works on par with supervised pre-trained models in standard fine-tuning scenarios.

1 Introduction

Nueral networks have proven to perform very good in many classification tasks and other computer vision tasks like Image Segmentation, Feature Extraction and High Level Image Processing. In many of these approaches Super Vised learning approaches have been followed, where the model has been trained on datasets with labels. But the important problem with such approach is that they need labelled data to train which might not always be possible. There could be issues like wrongly labelled samples which have a effect on the performance of the model. Many models not only need labelled data, they need them in huge quantities. Specifically the larger models with more number of parameters need large size datasets in order to train well and give good results. The other problem with such approach is that generally the trained models only perform on the datasets on which they have been trained on. So the models have to be trained again, in order to apply them into different datasets. New approaches have been developed which fine tune the already trained model, but these can only be followed when the datasets are related. In our work we discuss about a technqiue which can be followed to solve these problem.

New approaches in machine learning have evolved to tackle these problems which are self-supervised and semi-supervised techaniques. They have been designed to specifically tackle the problem of datasets with no labels at all or those with only few labels. In the next section **2**, we discuss few such approaches using contrastive learning and those in semi-supervised learning. In our work we first implement the SimCLR framework as defined in the paper [1], which has been discussed in the section**3**. We have used the ResNET50 architecture introduced in the paper[6] as our model for all the tasks. We have primarily used Cifar-10 dataset to train the model using contrastives learning technqiue. We have also used other datasets for different experiments which has been discussed later. All the training in the SimCLR approach has been performed with no labels, solving the problem of requiring labelled data during training time. This would give us an approach which can be followed when we come across datasets with no explicit labels. We later test our model in different settings like Linear Evaluation, Semi-Supervised evaluation and Fine Tuning performance, which has been

described in the later section **3**. To summarise, there are two main parts of work, the first one is training the model using the SimCLR framework, later we fine tune this model in three different ways. This final fine tuned model has been used to experiment in different experiments to discuss the performance of the contrastive learning approach. We show the performances of the trained models and analyse how these models perform in different dataset settings by comparing results as shown in the section **4**. We also describe a new data augmentation technique which has been used along with the existing once during the contrastive learning based training. We show few results in the same experiments which have been done using SimCLR framework to comment on the benefit of using this data augmentation .

In our experiments section **4**, we perform various analysis in different settings like linear evaluation, fine-tuning and semi-supervised settings. These have been done to compare the performance of SimCLR visual representations and how introducing a new augmentation into existing once make them even better. All the mentioned accuracies are on the Cifar-10, whcih is discussed later.In linear evaluation, the supervised model has reached an accuracy of 82.1% whereas the SimCLR version 1 has achieved an 59.7%. These higher supervised model accuracy can be attributed to it's benefit of looking at labels during tarining. Our new version 2.2 SimCLR has reached 59.7% by adding the Jigsaw augmentation. In the semi-supervised settings, the supervised model has reached accuracues if 29.7% , whereas the SimCLR version has reached 52.4%. We also have observed that the accuracies are similar in transfer learning on different in-domain and out of domain datasets. These show that the visual representations obtained using SimCLR are very good at capturing the the data information without explicit label requirements.

2 Related Work

We have gone through some existing works majorly focusing on semi-supervised and self-supervised learning approaches. We have come across some contrastive learning approaches for this applcation. In this section we go through some existing works in this domain.

2.1 Contrastive approaches for visual representation learning

In this specific approach a learning is designed by creating positive and negtaive pairs and then a constrastive learning is performed by supoorting positive once and differentiating negative once. The major idea is to create a loss function and sample creations and to combine them to create a final contrastive learning task. In the paper [4] titled Discriminative Unsupervised Feature Learning with Convolutional Neural Networks, the author talks about an approach of creating surrogate classes by different data augmentation techniques. They also talk about hose these visual representations obtained following contrastive learning compared to existing once. In other work Multi-task Self-Supervised Visual Learning[3], they suggest a method to create in-batch images to create negative samples. They combine multiple self-supervised tasks to create one self-supervised tasks and train the model, they also show these approaches could reach traditional training approaches on few datatsets.This work is a similar approach mentioned in the work [1] in approach towards cerating samples, which we have followed.

2.2 Self-Supervised learning using pretexts tasks

Many recent works have come up to tackle the problem of small size labeled data as self-supervised learning techniques. Many of these approaches have been designed by creating a pre text task such as relative patch prediction [2], solving jigsaw puzzles [9] and rotation prediction [5]. In relative patch prediction random patches of images has been taken and the model is trained to predict the order of the patches. In jigsaw puzzle an image is breaken into equal patches of images and randomly suffled to create a new image and the model is trained to predict the actual image/order in which it has been suffled. In a simpler task of image rotation prediction the model is trained to predict the rotation of the final image from the start. In all these approaches we create an intermediate task of psuedo labels by making the pretexts tasks. We then train our model to get the feature representations of the images. Later, we fine tune our visual representations withe the available few labels. This approach solves the problem of requiring large labelled datasets. Infact this is the reason we have followed a similar approach of using Jisaw for data augmentation. We have also made experiments to test our

representations in semi-supervised approaches. Although very good representations can be learnt from large networks and more training these lack generalisation into multiple tasks.

3 Method

Most of our work is primarily focused in the SimCLR framework for contrastive learning as discussed in the [1]. In this section we discuss how we perform training is such framework. We also discuss about the data augmentation techniques which we follow to create images in SimCLR framework, later we continue to discuss a new augmentation technique called as Jigsaw which we have used along with discussed augmentations. We also discuss about the loss function we generally use for this technique and how we evaluate the performance of the model.

3.1 The Contrastive Learning Framework

The framework is based on the idea of training the model to learn data representatioin of the data by maximizing the agreement between positive samples and minimizing between the negative once. The samples are created by combining images augmented using different techniques from same and different images. As shown in the figure 1 there are four main components in the framework:

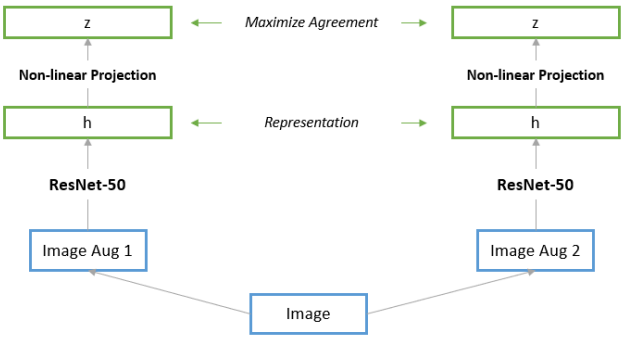
- **Data Augmentation** This module is responsible for creating 2 new images form the actual data with different data augmentation techniques. In addition to the augmentations used in the paper we have also used Jigsaw augmentation technique. In the section **3.2** we give examples of the images after using the augmentations. This module is shown in the Figure 1 as Image Aug 1 and Image Aug 2 blocks respectively. In total we have used 3 versions of data augmentation technique.
- **Base Encoder** A nueral base econder $f(.)$ which extracts the visual representation of the data $h(.)$. We have used ResNet50 architecture introduced in the [6] in all our experiments. The equation is given by $h_i = f(\tilde{x}_i) = ReseNet50(\tilde{x}_i)$, Where $\tilde{x}_i$ is the augmented data from the actual image x_i.
- **Projection Head** This module is shown in the figure 1 as $g(.)$. This layer is responsible for converting the data representations h_i to a different space which are similar to probabilities of some classes which aren't specifically defined. We only use this layer during the training and calculate the loss after the final projection heads. So, the actual data representation module is only till h_i. Here we use a simple one layer MLP to get the final projections. The formulation of this layer is given by $z_i = g(h_i) = W^{(2)}\sigma(W^{(1)}h_i)$ where σ is a ReLU nonlinearity. The contrastive loss described in the next item is taken on z_i than h_i as it has been shown in [1] that the final hidden layer is responsible for projection into probability space and doesn't have any information of actual data representation.
- **Contrastive Loss** The final module is the loss function layer which calculates the loss betwen the final projected h_i. FIr a given set if smaples with augmentations x_iWe use the **NCE** loss defined by the following function described below for all the positive smaples.

$$l_{i,j} = -log\left(\frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1 and k\neq i}^{2N} exp(sim(z_i, z_k)/\tau)}\right)$$

  where τ is the temperature(which is 0.7 for all our experiments) and N is the batch size.

The flow of the training algorithm is as follow. First using randomized data augmentation we create two augmented images for each image in a batch $\tilde{x}_i, \tilde{x}_j$. This way we create 2N samples in a batch for N samples in a batch. The set of two samples from same images is treated as a **positive** sample set and all other sets as **negative**. Then all the 2N samples are passed through the model Base Encoder $f(.)$. Later these image representations h_i is passed through the final projection head layer to get the z_i. After getting these we calculate the NCE loss - $l_{i,j}$, as described earlier between only positive samples and add it to get the final loss value, which is defined in the below equation. We then perform backward on this loss function. After the training we remove the projection head $g(.)$ layer and treat as final trained model.

$$\mathcal{L} = \frac{1}{2N} \sum_{i=1}^{N} \left(l_{i,j} + l_{j,i} \right), \text{ where j is the positive pair of i}$$



	Augmentation 1	Augmentation 2
V1	Standard	Standard
V2.1	Jigsaw	Standard
V2.2	Jigsaw	Standard + Jigsaw

Figure 1: SimCLR framework for contrastive learning along with different augmentation combinations.

3.2 Data Augmentation and Version of SimCLR framework

We have used the following data augmentation techniques used as used in the paper [1]. The link to the pytorch documentation for each tranform function has alos been linked.

- **RandomResizedCrop** - Randomly crops and resizes the image to 32*32(Cifar-10 size).
- **RandomHorizontalFlip** - Randomly flips the image horizontally.
- **ColorJItter** - randomly applies a color jitter across all cchannels with probability 0.8, with input (0.8 , 0.8 , 0.8 , 0.2)
- **RandomGrayscale** - Randomly converts the image into graysacel with Probability 0.2.

All the above augmentations have been combined together and applied randomly to both left and right size image samples as our primarily model version. We call this combination of the augmentation as **Standard** augmentation in future references. We have also used A Jigsaw as a data augmentaion method which is described below:

- **Jigsaw** - We break the Image into patches of 4 16*16 by breaking the image horizontally and vertically in the center.

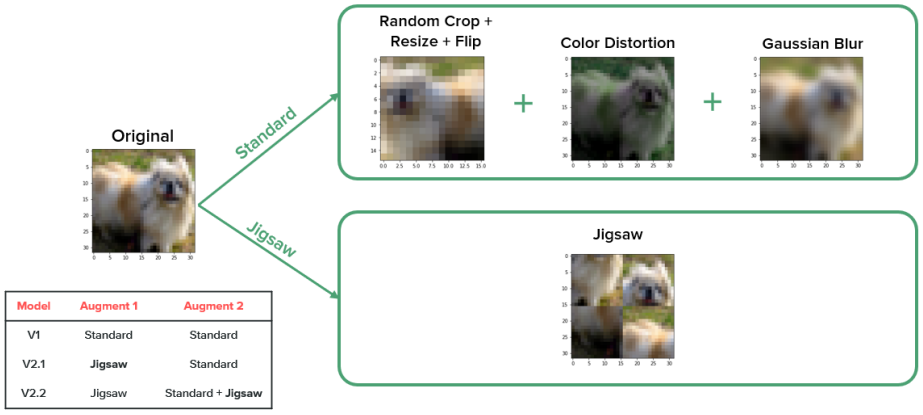
We believe that all the augmentations defined earlier change orientation and position but never changes pixel to pixel relation in terms of nerigbor hood relation. CNN which calculate by convolving with pixels neighbours might suffer form bias by using only this method. By introducing this Jigsaw augmentation we are removing that relation. Many other approaches could be followed to remove this bias like self attention, but given it simplicity we have followed this approach. In the next section we show the loss functions which show us that this approach could give us better image representation.

3.2.1 Versions of SimCLR

We have created three version of SimCLR framework by incorporating our new augmentation method Jigsaw as discussed eralier. 1.

- **V1** - both the augmentations are the Standard augmentation.
- **V2.1** - one of the augmentation is Standard only and other is Jigsaw.
- **V2.2** - one of the augmentation is Standard+Jigsaw and other Jigsaw Only.

The images after performing the augmentations can be seen in the Figure 2. In the next Section **4**, we perform different set of experiments to shown the effect of using our augmentation method on the final image representations.



Model	Augment 1	Augment 2
V1	Standard	Standard
V2.1	**Jigsaw**	Standard
V2.2	Jigsaw	Standard + **Jigsaw**

Figure 2: Sample image and processed image after different data augmentation.

4 Experiments

4.1 Datasets

We have used multiple datasets for the purposes of pre-training and transfer learning (both in-domain and out of domain). CIFAR-10 is used for pre-training while others are used for downstream transfer learning.

- **CIFAR-10** [7]: The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. These are general domain natural images and is used for pre-training our both the ResNet models - supervised learning and self-supervised learning.
- **Caltech101** [8]: Caltech101 consists of images from 101 general domain classes. Each class consists of rouhgly 40-800 images with a total of 9k images. Image sizes are around 200-300 pixels with an average image quality.
- **Oxford Pets**: The Oxford-IIIT pet dataset is a 37 category pet image dataset with roughly 200 images for each class. In the SimCLR paper which is trained on ImageNet, it is considered as natural images dataset, but given that we pretrain on CIFAR-10, we can't easily classify this as in-domain.
- **Satellite Land-Use**: Land-Use Scene Classification dataset contains images (of size 256x256) of 21 classes with 100 images per class. Some of the classes include buildings, baseball fields, parkings, freeways etc.
- **Brain Tumor MRI**: This dataset contains brain MRI scans classified into either one of the three tumor classes or non-tumor class. It contains a total of 7k images with almost balanced distribution.

4.2 Evaluation

Self-Supervised Learning can have multiple downstream objectives and hence we evaluate such model in multiple ways.

- **In linear evaluation,** we freeze the entire model and train the final non-linear layer. This is equivalent to doing a logistic regression in the SSL representation space and shows the generalization ability of SimCLR model
- **In few-label evaluation,** we fine-tune the entire model but with few % of sampled images. Here the SSL is used as a semi-supervised learning technique.
- **In transfer learning,** we fine-tune the model on different downstream dataset to understand the ability of SSL to generalize/transfer to different domains. Here, we also perform linear evaluation.

We want to highlight some differences compared to the original paper mainly due to the compute and time constraints. We have pre-trained our Supervised and SSL models for 100 epochs (compared to 500-1000 epochs in the paper). For transfer learning, we have trained our models for 100 epochs (200 in case of random initialization) compared to few hundred in the paper. In terms of hyper-parameter tuning, we have considered a small grid of 2-3 learning rates, 2-3 weight decays. We have used SGD optimizer with momentum of 0.9, exponential learning rate scheduler (rate that multiplies by factor of 0.9 every 10 epochs) with a batch size of either 62 or 128. We have also used dropout regularizer for few-label learning experiments and transfer learning fine-tuning experiments to avoid overfitting. Given all that, for fair comparison, for a given dataset and experiment, same hyper parameters are used to compare supervised vs SSL settings.

4.3 CIFAR-10 Linear Evaluation

To understand the representations learnt, we have performed linear evaluation of Supervised and SimCLR ResNet models. Table 1 shows the top-1 accuracy of these linear classifiers. While the standard supervised pretrained model gave 82% accuracy, the self-supervised technique (v1) gave an accuracy of 58.3% without utilization of any labels and with same architecture. This is inline with the results that are presented in the paper and shows the correctness in implementation.

Method	Pretrained Data	Architecture	Top-1 Accuracy
Supervised	CIFAR-10	ResNet-50	82.1%
SimCLR v1	CIFAR-10	ResNet-50	58.3%
SimCLR v2.1	CIFAR-10	ResNet-50	59.7%
SimCLR v2.2	CIFAR-10	ResNet-50	58.8%

Table 1: CIFAR-10 accuracies of linear classifiers trained on representations learned with different methods.

4.4 Jigsaw Data Augmentation for SimCLR

The standard SimCLR model (v1) uses the combination of random crop, random flip, random color distortion, and random Gaussian blur with certain probabilities. Taking inspiration from other pre-text tasks like Jigsaw puzzle representation learning [10], we have designed a new jigsaw augmentation where 2x2 image patches are randomly shuffled and used for one side of the data augmentation, keeping the other side same as the standard one. The data augmentations subsection in the Method section describes the related variants. The idea behind this is that the model will try to learn spatial connections by looking at different patches (similar to flip/rotate but a tougher task).

Figure 3 shows the contrastive loss for different variations of SimCLR data augmentations. We can see that this new augmentation (v2.1) improves the representation learning of CIFAR-10 with ResNet-50. The loss and accuracy curves show that jigsaw augmentation on one side used with standard SimCLR augmentation helps with better convergence. However, including this on both sides slightly diminished the performance and we need to run more experiments to understand the impact. We can also see that in Table 1, there is an improvement in linear classifiers for these variations. Overall, we see potential in adding this new sampling/augmentation strategy. However, in the lack of time, we couldn't experiment this on downstream transfer learning tasks.

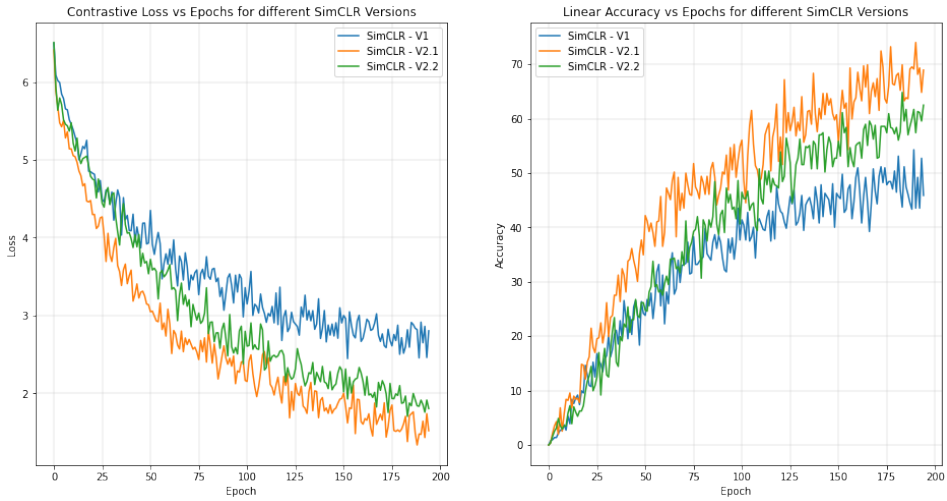


Figure 3: Contrastive loss and linear accuracy for different SimCLR versions during training

4.5 CIFAR-10 Few-Label (Semi Supervised) Evaluation

Self-supervised learning can be used as semi-supervised technique where the representations can be learnt on the whole dataset, while downstream fine-tuning is done only on the labelled samples. We have tested this approach using the SimCLR model with a supervised baseline and Table 2 shows these results. Since the training dataset is small, we have used dropout after activation and fully connected layers to avoid overfitting.

Supervised learning with 1% labels gave an accuracy of 29.7% while SimCLR v1 gave 52.4%. Hence utilizing the unlabelled dataset for representation learning and fine-tuning on labelled dataset definitely helps improve the performance. This effect can also be better seen in 10% labels case. Note that the other variants of SimCLR gave comparable performance with the v1 version suggesting that these representations are similar in the semi-supervised case.

		Fine-tuning Accuracy	
Method	**Pretrained Data**	**1% Labels**	**10% Labels**
Supervised Baseline	-	29.7%	43.8%
SimCLR v1	CIFAR-10	52.4%	70.5%
SimCLR v2.1	CIFAR-10	52.4%	69.9%
SimCLR v2.2	CIFAR-10	49.3%	67.2%

Table 2: CIFAR-10 few label fine-tuning accuracies on different pretraining methods.

4.6 Transfer Learning

We evaluated the pre-trained models on multiple in-domain and out-of-domain downstream classification tasks. While Caltech 101 is very similar in domain compared to CIFAR-10, Satellite land use and Brain tumor MRI datasets can be considered as out of domain. Regarding Oxford Pets, even though SimCLR paper considered this as natural images dataset, given that we pretrained on Oxford Pets, it can be considered as out of domain in this case.

For linear evaluation, we freeze the pretrained ResNet-50 model and train the final non-linear layer only. We can see that across all the datasets, we SimCLR has better performace compared to Supervised pretrained models. This shows that under similar model settings, SimCLR representations are more generalized for different domain datasets. This shows that SimCLR model can be used as a good general feature extraction technique in vision related tasks. The left two plots in the below figures elaborate this point in terms of better validation performance of SimCLR.

For fine-tuning, we start with pretrained weights and update all the layers of the model during training. Here we can see that Supervised model performed better, however, SimCLR results are also very

close. This can be seen the right two plots of the below figures where we can see that the validation performance of SimCLR and supervised models are very close across the datasets. This shows that SimCLR models are close competitors in the transfer learning paradigm that doesn't need explicit labels to create that huge pretrained models.

Method	Caltech 101 (In Domain)	Oxford Pets* (In/Out Domain)	Satellite Land Use (Out of Domain)	Brain Tumor MRI (Out of Domain)
Linear Evaluation				
Supervised	40.6%	11.3%	35.2%	81.8%
SimCLR v1	49.0%	17.2%	52.1%	85.8%
Fine-Tuning				
Random Init	65.8%	18.3%	49.3%	96.5%
Supervised	76.3%	45.1%	63.1%	96.5%
SimCLR v1	73.0%	41.6%	58.8%	96.8%

Table 3: Transfer learning (linear evaluation and fine-tuning) on different in-domain and out-of-domain datasets. Note that the Oxford Pets dataset overfit during training and hence different regularization and hyper-parameter tuning is required to improve it's performance. Nevertheless, the results follow the general trend similar to other datasets.

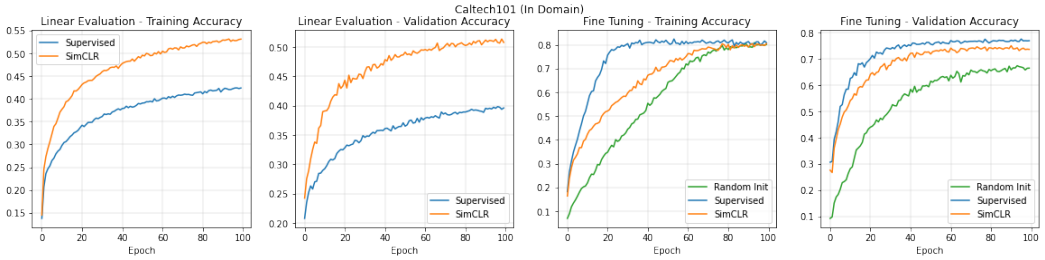


Figure 4: Caltech101 - Accuracies for linear and fine-tuning evaluations

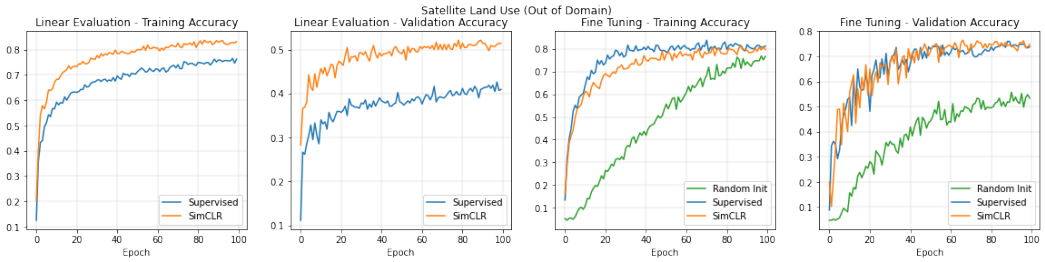


Figure 5: Satellite Land Use - Accuracies for linear and fine-tuning evaluations

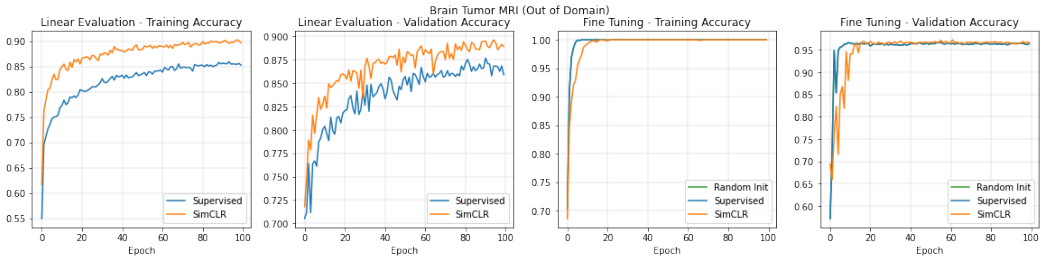


Figure 6: Brain Tumor MRI - Accuracies for linear and fine-tuning evaluations

5 Conclusion

From our experiments it is evident that SimCLR framework can be followed to training models to get the visual representations without any explicit labels. These helps us to deploy models in nor or low label dataset settings. We have also shown that using other dataaugmentation techniques like Jigsaw can better these representations. It can also be understood that these models generalise well in both in-domain and out of domain datasets. These models also work in par with the existing transfer learning based models. In our future work we wish to explore more data augmentation techniques which could help model to learn better visual representations and generalise well. We also want to perform the same experiments on larger models which have shown to perform better almost similar in supervised and unsupervised setting.

6 Supplementary Material

You can watch a video presentation on our project which could be found in this link - video
The link to our presentation slides can be found here - Slides
The link to our GitHub repository can be found here - GitHub Repo

References

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations, 2020.

[2] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015.

[3] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. *CoRR*, abs/1708.07860, 2017.

[4] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[5] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[7] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research).

[8] F.-F. Li, M. Andreeto, M. Ranzato, and P. Perona. Caltech 101, 2022.

[9] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *CoRR*, abs/1603.09246, 2016.

[10] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.