TABLE OF CONTENTS

1.INFORMATION ABOUT THE INSTITUTION

ITU Vision lab is a multidisciplinary research laboratory in the field of computer vision and computational neuroscience. It is led by Prof. Dr. Gözde Ünal and It takes place on Istanbul Technical University, Ayazaga Campus, Faculty of Computer and Informatics Engineering, Room 1203. There are 4 Ph.D. students, 1 Ph.D. candidate, 5 Master's degree student and 10 undergraduate student in the laboratory. The research approach followed by the laboratory is based on human visual system, therefore artificial neural networks and deep learning provide a variety of solutions to computer vision problems. Current main reseach area of the laboratory is focused on continual learning, 3D vision, uncertainty ,explainable AI and representation learning. Since it is a research laboratory, there no strict distriction of topics between people therefore; it is not possible to say one person leads one topic. My internship took place mainly on representation learning however, I also benefited from people for working on other topics.

2. INTRODUCTION

This internship was a continuation of the ongoing project from the last spring term with ITU Vision lab. In short, the main purpose of the project was creating a research paper which uses novel contrastive learning method in audio domain which is image domain in the original paper. Even though there exist new research papers in the field of contrastive learning, for the sake of simplicity and the reproducibility we choose the most cited and the first published paper in the field of contrastive learning as the baseline which is "A Simple Framework for Contrastive Learning of Visual Representations''by Ting Chen, Simon Kornblith, Mohammad Norouzi and Geoffrey Hinton. This paper will be abbreviated as SimCLR during this report.

My work during the internship can be divided into three parts; extending the existing literature review, implementing the novel contrastive learning algorithm and conducting the experiments. All code is written in Python and can be accessible from GitHub. From beginning to end, I have been supervised by Prof. Dr. Gözde Ünal, Yusuf H. Şahin and Alican Mertcan.

3. DESCRIPTION AND ANALYSIS OF THE INTERNSHIP PROJECT

The main purpose of this project is transforming SimCLR from vision domain to audio domain by using audio augmentation methods. In the literatur review part you will see the detailed explanation of SimCLR because It has been taken as a baseline and my implemantion is mainly SimCLR working with audio augmentations. On the other hand you will see less information in the Music Genre Classification part because our main interest was not to discover new topics about music information retrieval but rather show that contrastive learning is not only an idea of computer vision.

3.1 Literature Review

This part consist of three subtopics. First, initial point of the problem will be explained. Later SimCLR paper will be comprehensively covered. Finally, results from my literature review over music genre classification will be shown.

3.1.1 Problem Statement

Learning of the data representations with minimum human supervision is a challenging problem. Even though supervised methods achieves state of the art results, labor cost of the labeling and annotating make researchers to develop better methods for unsupervised learning. The most common approaches to this problem are generative and discriminative methods. Generative methods learn to generate pixels based on the input space. Due to their computational expensivity they are not efficient in terms of representation learning. Discriminative methods try to optimize objective functions similar to supervised learning methods. In this project SimCLR which is a novel discriminative method, has been used as a baseline and further improvements made accordingly.

3.1.2 SimCLR

A novel method of selfsupervised contrastive learning framework in visual domain outperforms many unsupervised methods. The basic motivation behind the contrastive learning is, creating a similarity loss and optimizing it. Optimization is being done by using the contrast between positive pairs,which is the augmented version of the groundtruth, and the negative pairs. Given an batch size N, except the particular sample x_i and its augmented version x_j; $2(N-1)$ negative pairs will be created. After having positive pairs, using normalized temperature-scaled cross-entropy loss; model will not only try optimize similarity between positive pairs also it will try to increase dissimilarity between negative pairs. Example of pair creation process can be observed from **figure 1**.

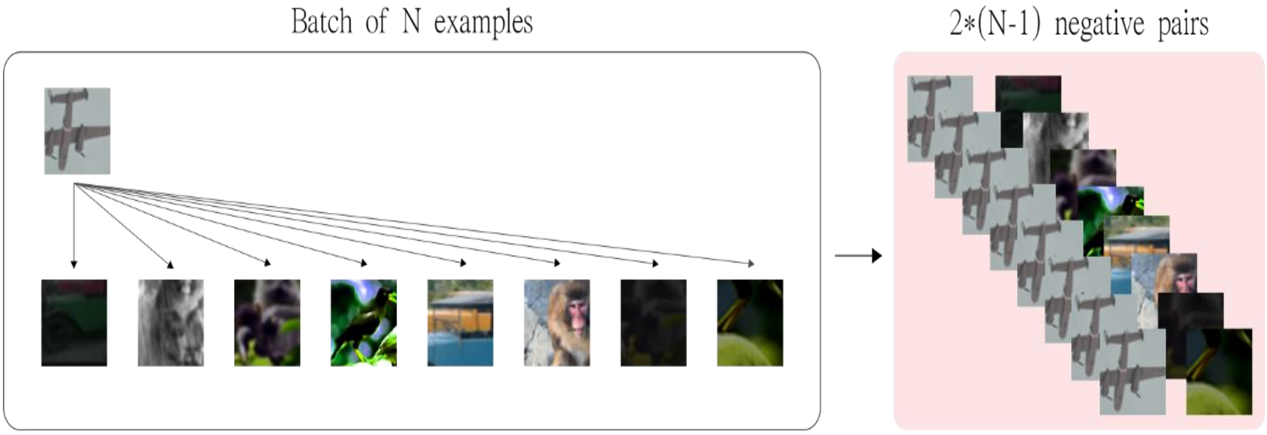


Figure 1: Example of negative pairing[1]

The contrastive learning frame work consist of three stages where it can be seen from **figure 2**. In the first stage stochastic data augmentation is applied to the input, yielding two augmented versions of input x. In the second stage, a feature extractor neural network used in order to get representations. In the third stage one layered nonlinear MLP is used before optimizing the positive pairs.

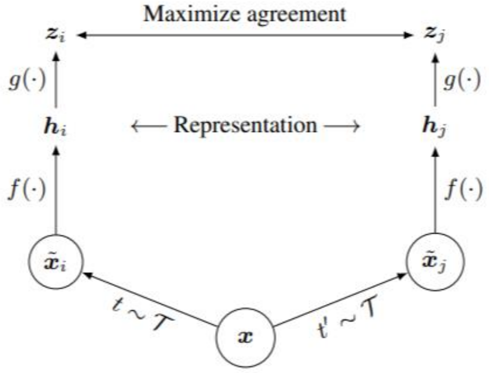
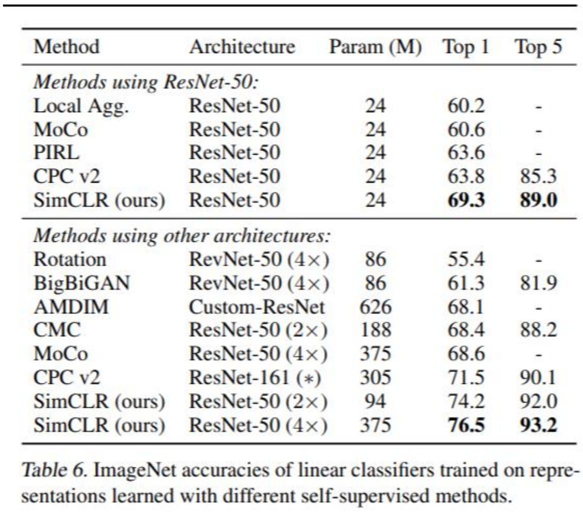


Figure 2. A simple framework for contrastive learning of visual representations. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training is completed, we throw away the projection head $g(\cdot)$ and use encoder $f(\cdot)$ and representation $\boldsymbol{h}$ for downstream tasks.

Figure 2: Learning framework [2]

Important implementation details that are stated in the paper are; composite stronger data augmentations take a crucial role in optimizing contrastive loss, usage of a non-linear projection after having representations improves the results significantly, larger batch sizes and longer training times yield better results and finally deeper and wider networks improve the framework.

One of the main reasons that we took this approach as a baseline is, the significant improvement in the classification task which can be monitored from **figure 3**. During their experiments they used a percentage of data to train a linear classifier top of the representations. In **appendix 1** they also compared their model with their previous works and supervised algorithms.



Method	Architecture	Param (M)	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (∗)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

Figure 3: Linear Classification [2]

3.1.3 Music Genre Classification

Music genre classification is a subtask of music information retrieval. There are two main approaches in music information retrieval which are content and context based. Content based approach is using raw audio data and Its extracted features, on the other hand context-based approach is using metadata such as song name, singer name, year of release. In this internship a comprehensive research has been done on content-based approach. First, state of the art papers has been found and their implementations are reproduced and details from each work collected. Second, public datasets have been found and their related papers are investigated. It is found that some datasets which have been used in previous works are not healthy in terms of data distribution therefore they cannot be used in our experiments. After finishing the literature review it is found that FMA, GTZAN and MSD datasets are suitable for our experiments due to their popularity in the music information retrieval and their stability. Also, it is observed that best results achieved by transforming raw audio to Mel spectrogram and then feeding them into a convolutional neural network. Melspectrogram whose examples can be seen from **figure 4**, is a nonlinear transformation in frequency domain which allows us to see the spectrogram as we hear it. For an example many people can differentiate 1000 Hz and 2000 Hz but they cannot differentiate 19 000 from 20 000 Hz even though the difference is same 1000 Hz.

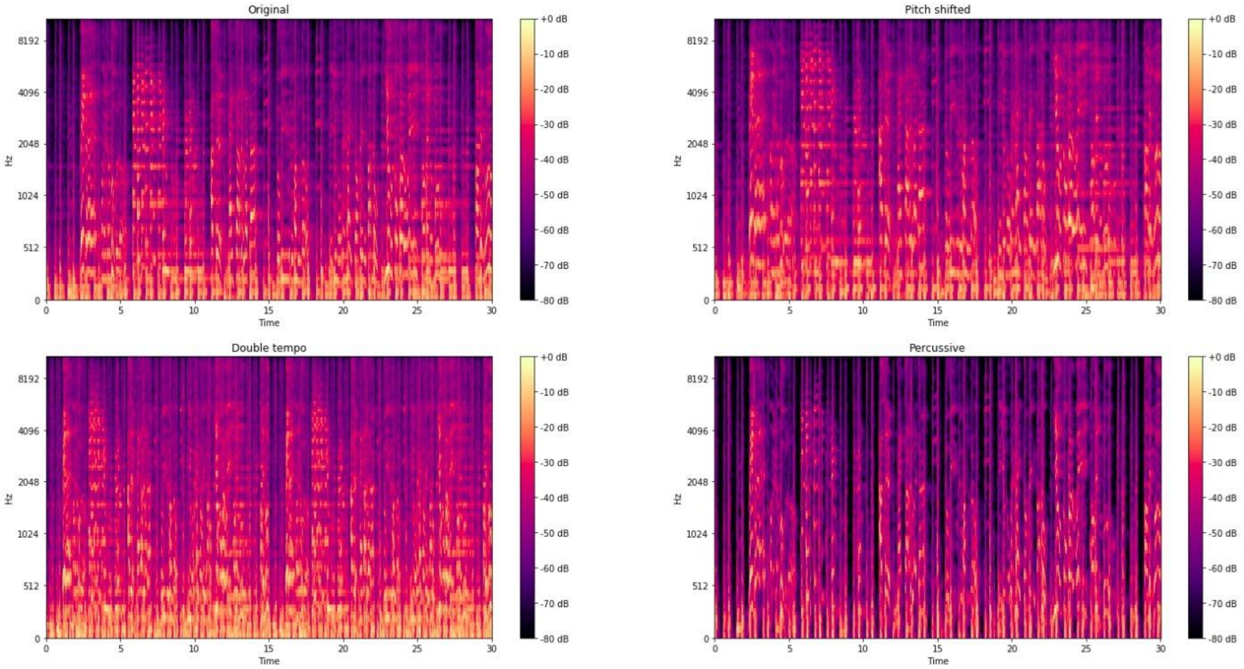


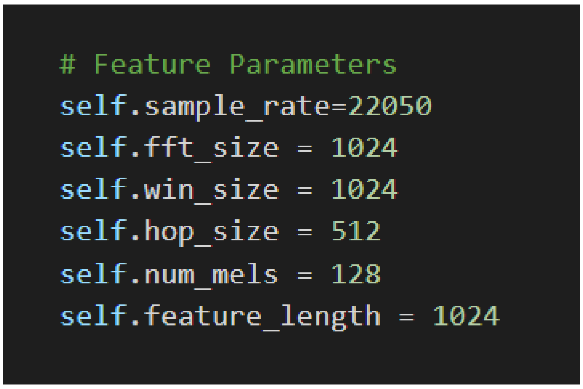
Figure 4: Example of Mel Spectrograms

3.2 Implementation of SimCLR with Audio

Implementation consist of 4 stages; first preprocessing the data, second creating the loader class, third training the model and fourth evaluating the model. For all augmentation and audio processing Librosa library of python is used and for the neural network parts Pytorch library is used. Pytorch is one of the most common libraries in the field of deep and machine learning. Implementation can be reached by using the link provided in **appendix 3**.

3.2.1 Preprocessing

SimCLR algorithm requires strongly augmented data therefore, before feeding the model with audio; there shall be data augmentation. In our setup, random pitch and speed change are used as audio data augmentations due to their high performance in supervised learning methods. Random pitch perturbs the frequency values of given audio and speed change alters the audio's speed. Raw audio data is too big in terms of size to create a meaningful spectrogram and to feed it into a neural network. Thus, after the augmentations raw audio should be sampled. From state-of-the-art model it is observed that best way to handle the sampling is dividing the songs into 8 second parts. After having divided the songs, Mel spectrograms have been calculated. Parameters of sampling and spectrograms can be seen from **figure 5**.



```
# Feature Parameters
self.sample_rate=22050
self.fft_size = 1024
self.win_size = 1024
self.hop_size = 512
self.num_mels = 128
self.feature_length = 1024
```

Figure 5: Details of Mel Spectrogram

3.2.2 Data Loader

Loader is implemented by inheriting Pytorch Dataset Class which is the base class for implementing the data pipeline. It takes raw audio and labels, applies augmentations and feed it to the network in an indexed format. Writing a Pytorch loader was one of the hardest things while implementing this algorithm due to audio augmentations and its pair structure. As explained in the previous part, first augmentations are done later audio is transformed into to spectrograms. In order to have better results, augmentation should be done during the training dynamically which means process explained above should be repeated during each batch which is the number of samples feed into the model at once in order to update model parameters. In addition to that, to train contrastive learning model for each sample x there should be two augmented spectrograms such as x_i and x_j.

3.2.3 Model Training

The original SimCLR paper uses a pretrained Resnet50 as a backbone, which is the part of the model who extracts information, however in our project EfficientNet is used. EfficientNet is a model architecture search paper which is stating there is no need of enormously deep and wide models in order to get best results but rather there is need for scaled depth and width models according to the data size. The main reasons why we are using EfficientNet is, as explained in the literature review SimCLR works better with larger batch sizes and by shrinking the model size we can increase batch size. As it can be seen from the **figure 6** EfficientNet outperforms almost all models with the smaller number of parameters in Imagenet. We did not had chance to reproduce SimCLR because of the batch size. In SimCLR paper they are having the experiment results with the batch size of 2048 which is not possible for us due to GPU limitation. In order to test the EfficientNet with audio data, instead of SimCLR paper; state of the art transfer learning method has been tested with EfficientNet and the result were almost same with the Resnet50. Transfer learning method is a basic method of using a pretrained model on another task. All testing process is done on Google Collaboratory

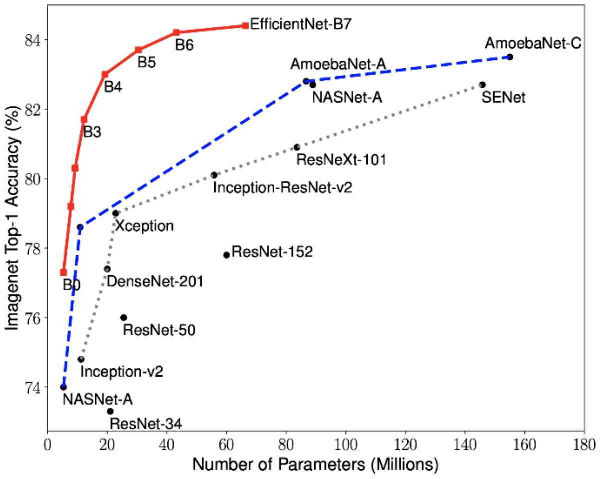


Figure 6: EfficientNet Comparison [3]

After selecting the backbone training procedure is configured. Stochastic gradient descent is used as an optimizer and non-linear projection head is added into network. Since spectrogram data has one channel, it is stacked up to three channels in order to use Pytorch convolutions efficiently. Representation learning stage of the framework has been trained over 100 epochs which is observed as the optimum number before overfitting. Before evaluating the model, non-linear projection head should be replaced by a linear classifier head. Linear classifier head is trained over 50 epochs only with the %10 percent of the data as described in the SimCLR paper.

3.2.4 Experiments and Results

The first experiment has been done with GTZAN dataset. GTZAN consist of 1000 audio tracks with 30 seconds long. There exist 10 genres and each genre represented by 100 tracks. It may seem as a balanced dataset however; due to semantic classification, it is an ambiguous problem. An experiment conducted on people show that, agreement rate on the genre of a given audio is only %76 [4]. For example diferantiating a blues song from a metal song is easy but a blues song from a rock song is tricky because they may share the same musical elements. Beside the genre problem explained above, GTZAN has an artifact problem. Songs from genres are usually selected from same album, therefore models are learning artifacts from audio recorders and recording studio acoustics instead of the musical elements from the song.

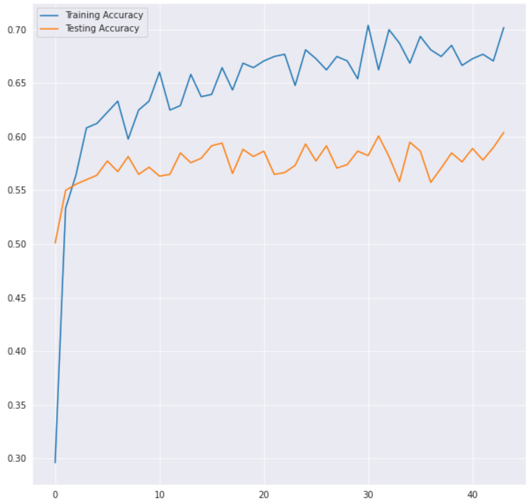


Figure 7: Accuracy Plot from First Experiment

GTZAN is used as the toy dataset because; the number of tracks is relatively small, it is free to use, tracks are in mp3 format which can easily processed and there exist a variety of benchmarks which we can compare with our work. FMA and MSD datasets were also scheduled as the further experiments however; due to time limitation of the internship, those experiments did not conduct. After the first model is trained, as it can be seen from the **figure 7** top 1 accuracy results were about %60. The reason why score was bad, I forgot to replace the non-liner projection head with the linear one. After fixing the bug and hyperparameter tuning, accuracy results increased to over %75. The final results were not bad however, it was not as close as the supervised state of the art model like in the original SimCLR paper was.

As internship was ending, this project was put to an end due to several reasons. First, by the time we got the first experiment results, there start to exist papers about implementing contrastive learning in the audio domain. Second, current music information retrieval journals are more interested in multimodal and domain specific approaches so there was a high risk of not being accepted. Third our results were not high as the supervised ones.

4.CONCLUSION

To summarize, during this summer internship a detailed literatur review has been done in the field of contrastive learning and music genre classification. All process are supervised by Prof. Dr. Gözde Ünal and Yusuf H. Şahin. We had regular weekly meetings and deadlines which in my point of view, was very productive and efficient. First, SimCLR paper is choosen to implement for the audio domain. Later, results of SimCLR is reproduced; then implementation of the audio part is started. Further on, in order to use audio data efficiently; a variety of augmentations and transforms are applied on the raw data. Finally relatively small pretrained EfficientNet model is choosen and several experiments conducted with it. As a summer research internship, it was a very informative introduction to the world of acedemy.

5. REFERENCES

1. T. Silva, “Exploring SimCLR: A Simple Framework for Contrastive Learning of Visual Representations,” *Medium*, 02-Mar-2020. [Online]. Available: https://towardsdatascience.com/exploring-simclr-a-simple-framework-for-contrastive-learning-of-visual-representations-158c30601e7e. [Accessed: 27-Apr-2020].

2. Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton: “A Simple Framework for Contrastive Learning of Visual Representations”, arXiv:2002.05709, 2020.

3. Mingxing Tan, Quoc V. Le: “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”, 2019, International Conference on Machine Learning, arXiv:1905.11946 , 2019.

4. Stefaan Lippens, Jean-Pierre Martens, and Tom De Mulder. A comparison of human and automatic musical genre classification. In Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP’04). IEEE International Conference on, volume 4, pages iv– iv. IEEE, 2004.

6.APPENDIX

1.

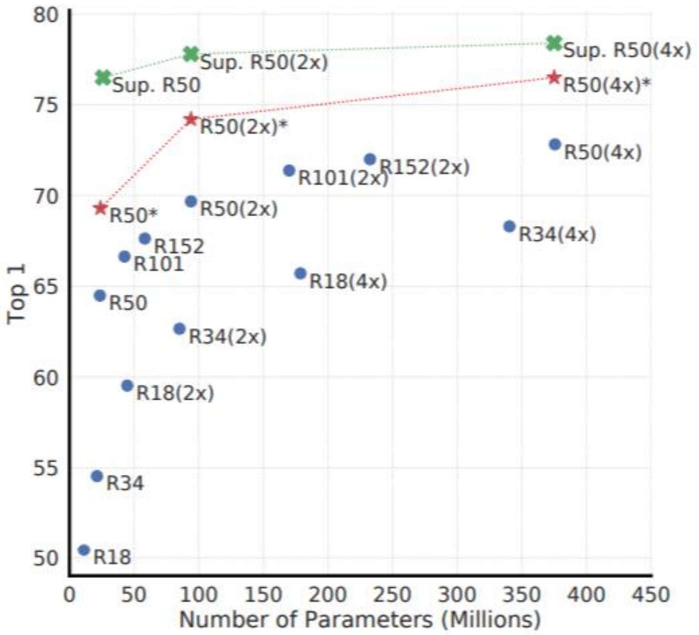
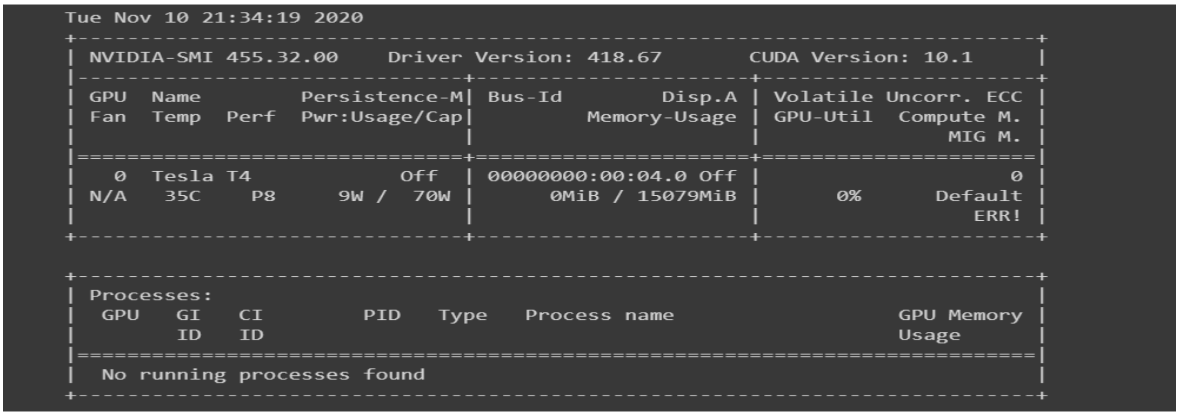


Figure 7. Linear evaluation of models with varied depth and width. Models in blue dots are ours trained for 100 epochs, models in red stars are ours trained for 1000 epochs, and models in green crosses are supervised ResNets trained for 90 epochs[7] (He et al., 2016).

Comparison with other works [2]

2.



```
Tue Nov 10 21:34:19 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 455.32.00    Driver Version: 418.67       CUDA Version: 10.1     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   35C    P8     9W /  70W |      0MiB / 15079MiB |      0%      Default |
|                               |                      |                 ERR! |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

GPU of Google Collaboratory

3. https://github.com/TrubadurOsman/SimCLRAudio