Self-Supervised Learning by Rotation Feature Decoupling

Sabaina Haroon
University of Central Florida
4000 Central Florida Blvd, Orlando, FL 32816
sabainaharoon@knights.ucf.edu

Abstract

This paper aims to enhance the baseline method proposed by Feng et al [1]. It introduces a self-supervised method that attempts to decouple rotation feature knowledge learnt as a pretext task in Self-supervised learning, from instance representation. In this way, proposed method enhances the generalization ability of classification and other down-stream tasks related to an unlabeled dataset. Feature decoupling is done by combining three different methods, positive unlabeling, self-supervised learning with rotation classification pretext task and non parametric learning of instance classification using Noise Contrastive Estimation. All three methods combined helps the network to be rotation invariant and generalized enough to recognize instances on individual level. Different modifications were applied to this method in order to enhance the baseline accuracy which included adding skip connections to the AlexNet with different configurations with combination of DenseNet and Resnet. Since baseline method uses three different loss functions with weight of one for each, our modified method applies weight scheduling scheme to further enhance the performance. Lastly we add different kind of augmentations to the data owing to the fact that self-supervised methods benefit majorly from augmentations applied to the unlabeled dataset as it allows the network to learn data distribution with more generality and robustness.

1 Introduction

The work of Krizhevsky et al. [2012] showed the Convolution Neural Network (CNN) to be a compelling solution to image classification. Neural network classifiers like CNN's require the training of numerous parameters. The RESNET-50 architecture He et al. [2016] for example, has 23 million trainable parameters. Large datasets such as video, typically require individual image annotations as ground truth for training deep learning methods of classification. Image annotation is a labor intensive effort and often contains some amount of noise due to human imprecision. As an example labor estimate, if an annotation takes about 10 seconds per video clip, the Kinetics video action dataset **?** with 650,000 labeled video's, represents approximately 1,805 labor hours. For Image dataset, we can break down this timeline and it would still be a considerable annotation time.

Many efforts have sought to mitigate the need for human labeled data with methods such as weakly supervised learning, semi-supervised learning, self-supervised learning, and unsupervised learning. Weakly supervised learning Gokberk Cinbis et al. [2014], **?**, **?**, **?** uses data which has weak labels such that each image has one or more labels but no localization bounding boxes. With semi-supervised learning Jeong et al. [2019], **?**, Nguyen et al. [2019], **?**, Oliver et al. [2018], Zhu [2005], a combination of supervised and unsupervised learning is done using a labeled dataset and unlabeled dataset, respectively. In self-supervised learning **?**, **?**, **?**, Ahsan et al. [2019], **?**, **?**, **?**, **?**, **?**, **?**,supervisory signals for the training dataset are generated by extracting contextual or correlational

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

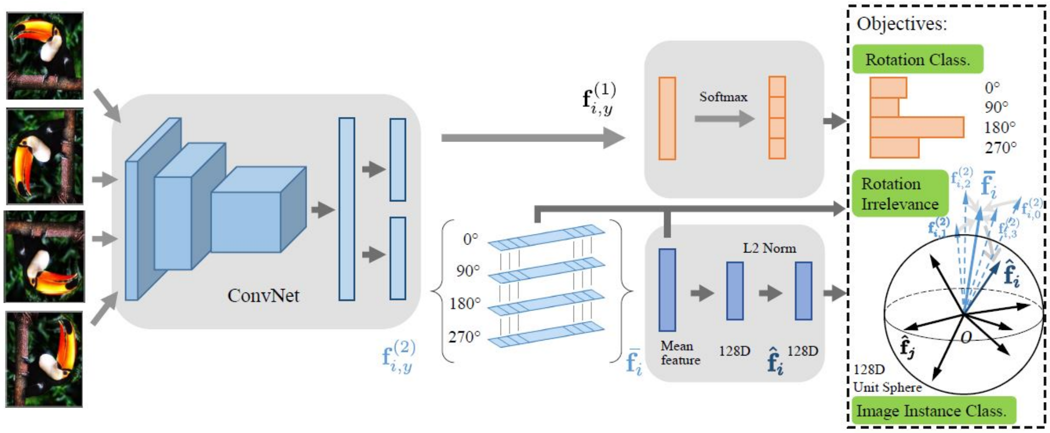


Figure 1: Illustration of the proposed method by Gidaris et al. [2018]. The neural network outputs a decoupled semantic feature containing rotation related and unrelated parts. The first part is trained by predicting image rotations. Noises in rotation labels are modeled as a PU learning problem, which learns instance weights to reduce the influence of rotation ambiguous images. The other part is trained with a distance penalty loss to enforce rotation irrelevance together with an instance discrimination task by using non-parametric classification

information and is considered a special case of supervised learning. Unsupervised learning Zhang et al. [2019], **?**, **?** looks for patterns in the dataset and models probability densities over the input samples. In this paper, we will be focusing onto self-supervised learning and how transformation invariance to a distribution can be learned by applying modification to rotation pretext task by Gidaris et al. [2018].

Method proposed in this paper passes rotated copies of an unlabeled input image through a ConvNet. Features learned through ConvNet are used for two projection heads. One projection head learns Rotation classification with Positive unlabeled learning and second projection head learns instance classification for mean features for all the rotated feature representations of an image. Instance classification learns features that are independent of rotation classification hence decoupling rotation feature learning from instance classification. Features learned through this method also work for rotation agnostic images present in the dataset, enhancing further the performance of conventional rotation self-supervised learning, by mitigating the error induced into the training due to images that are symmetrical, circular or invariant to rotation. On the other hand, instance discrimination task tries to learn individual instances inside a dataset, which can be greater or less than the available categories in a dataset. Categories in an unlabeled dataset are unknown to us.

Modifications introduced on top of this method are as follows. Note that these modifications are in accordance with the limited time available for this project. Experimental results gave further insight for what can be done in future.

- Due to time constraint experiments were performed on TinyImagenet dataset. For tinyImagenet baseline dataloader had to be modified. Image size for TinyImagenet was 64x64, therefore I changed ConvNet used in baseline accordingly to fit with new image size.
- I tried adding three different kind of skip connections to the ConvNet, skip connections added by convolution for downscaling, skip connections added by concatenation such as DenseNet, and as ResNet, connections are added after batch normalization and before applying linearity ReLU. Last type of modification to the architecture was done by adding two ResNet blocks in the beginning of the ConvNet and two in the later layers.
- Image Augmentation plays a very important role in self-supervised learning, since it makes the procedure robust for various complexities attached with unknown distributions. I added different transformations to the input images along-with rotation such as Random crop, random horizontal flip, random change in saturation, brightness, hue and contrast via application of color jitter to input images

- Lastly three different type of loss functions are used in feature decoupling learning, baseline paper used weight of 1 for all the losses, I applied a ramp function weight scheduling scheme for all the losses.

2 Related Work

This paper works by knitting together different successful methods discovered in the field of self-supervision learning, Positive Unlabeled learning and non-parametric Learning through Noise Contrastive Estimation. We will talk about these related methods that are involved in the architectural design of this project along with other works done in the field.

2.1 Self Supervised Learning

Self-supervised learning learns input data distribution by defining classification networks with different pretext tasks for unlabeled datasets such as work done in image colorization Zhang et al. [2016], Exemplar-CNN Dosovitskiy et al. [2014], Rotation Gidaris et al. [2018], Image Patches Learning Doersch et al. [2015], Noroozi and Favaro [2016], Generative Modeling with denoising autoencoder Vincent et al. [2008], context encoder Pathak et al. [2016], Bidirectional GANs Donahue et al. [2016], Contrastive Predictive Coding van den Oord et al. [2018], CDC is also inspired by Noise Contrastive Estimation (NCE) Gutmann and Hyvärinen [2010] which we will also be using in our method, Momentum Contrast He et al. [2019], SIMCLR Chen et al. [2020] and BYOL Grill et al. [2020]. These representations are strong baselines for several downstream tasks compared to random weight initialization. Whether we use these self-supervised representations for fine-tuning, semi-supervised learning or self learning, It provides a strong prior from tremendous amount of unlabeled data available today.

2.2 Positive Unlabeling

Positive Unlabeled (PU) problem is a form of semi-supervised learning paradigm where labeled data is treated as positive examples and unlabeled data is used as negative examples. We train a standard classifier on this data such that classifier should predict higher scores for positive examples than negatives. Among the unlabeled data, treated as negative examples, samples that have highest probability are treated as positives and classifier is trained again based on this newly added positive samples. This process is repeated until convergence is reached or some stopping criteria is met. This naivest approach of solving PU was also described in Elkan and Noto [2008]. Different works have been done in this domain using bagging, and random forest algorithms.

In Mordelet and Vert [2010], bagging is used with bootstrap method to train using unlabeled data points. A training set consists of positive data points and a single negative bootstrap sample that is replaced after every pass. Classifier trained is then used to test scores on Out of bag data points (unlabeled) and are recorded. This process is repeated several times and the average OOB score is assigned to each unlabeled data point. In Li and Hua [2014] authors implemented PU problem with parallel computing thereby introducing PU Gini index also called PURF-GI. in Kaboutari et al. [2014] authors used two-step approach, where in first step a subset of unlabeled data-points are identified that can be confidentally classified as negative samples. In second steps these negative examples and positive labeled examples are used to train the standard classifier and apply this process to remaining unlabeled points.

Our approach uses PU learning for classifying rotation agnostic images or images in default orientation as positive samples and input image rotated via four different offsets as negative samples, to minimize the error caused by rotation invariant instances in the self-supervised rotation task learning.

2.3 Instance Discrimination

Instance Discrimination tries to classify images into individual instances instead of dataset defined or manually annotated categories. This helps us to learn data distribution in an unsupervised manner and generalizes the learnt network well for better robustness and many downstream tasks. The problem with naively implementing instance discrimination is that number of instances in a dataset range to the size of sample points inside the data. Softmax operations for such networks tend to be highly

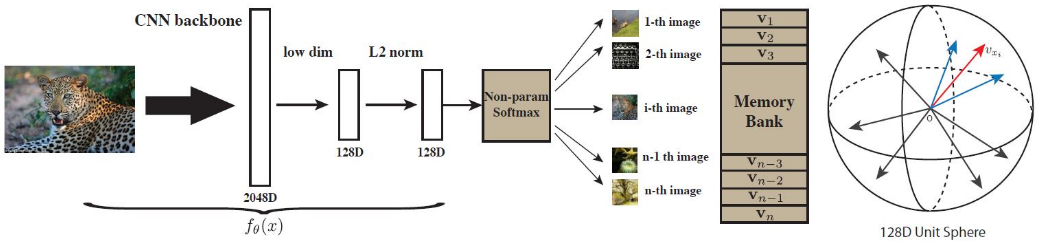


Figure 2: Detailed view of Instance Discrimination part of our method. We use a backbone CNN to encode rotated copy of each image as a feature vector, which is projected to a 128-dimensional space and L2 normalized. The optimal feature embedding is learned via instance-level discrimination, which tries to maximally scatter the features of training samples over the 128-dimensional unit sphere.

computationally intense since one has to sum millions of weights from entire dataset in denominator of softmax operation, depending on the size of the dataset. Different approaches that try to tackle this problem include hierarchical softmax by Morin and Bengio [2005], negative sampling by Mikolov et al. [2013], and Noise Contrastive Estimation by Gutmann and Hyvärinen [2010].

This project uses method introduced by Wu et al. [2018], where each image is encoded as a 128-Dimensional feature vector of instances using noise contrastive estimation to approximate the non parametric regression between positive and negative samples, by sampling a limited number of negative samples from the dataset according to probability of them being dissimilar from a given feature vector of an instance, instead of entire dataset.

3 Method

Method used in this paper can be broken down into three parts. In the first part let us discuss about the rotation prediction problem. Self-supervised learning is used to learn features for unlabeled dataset by providing some sort of supervisory signal as a pretext task. Pretext task of rotation classification was introduced by Gidaris et al. [2018]. We use this same method in our approach, where four different rotated copies of an image are generated against an input image. Rotations are done with an offset of 90°. If we have N unlabeled input images and each image is donated by X_i. Then Rotnet generates a new set of input images with a function as;

$$X_{i,offsetRotation} = \textit{RotNet(}X_i\textit{, offsetRotation)}$$

The objective function for RotNet can then be defined as;

$$\min_{\theta} \frac{1}{N*4} \sum_{i=1}^{N} \sum_{y=1}^{4} \mathcal{L}(F(X_{i,y}, \theta), y) \qquad (1)$$

Where $F(X_{i,rot-offset}, \theta)$ is a convolutional neural network that takes in rotated image and encodes it into feature vector θ. And y is equal to *offsetRotation* variable defined in previous equation. Our objective is to then learn these feature vectors such that the classifier is able to differentiate between different rotations applied to an image.

Before applying this objective function in form of a CNN, we should take into account the noise, network will encounter due to the presence of rotation agnostic images inside the dataset. These are the images that are symmetrical, circular or not in an upfront posture. RotNet would not be able to predict rotation of such images, as after rotation with an offset of 90 degree they will still be in 0° position. To mitigate the affect of such noise we treat input samples as a positive unlabeled problem. Where input image with no orientation is a positive sample and rotated copies are unlabeled negative images. Binary classification is then used to give weight $w_{i,y}$ to the images thereby predicting if an image is rotated or not. This weight is obtained from binary classifier as conditional probabilities of this network. And then applied to equation 1 to mitigate the effect of noise in rotation prediction.

Table 1: Training and Testing Results of the proposed feature decoupling architecture. Results show precision for Rotation instance classification together with loss

	Training				Validation			
Experiment	precRot	$\mathcal{L}_{\nabla}$	$\mathcal{L}_{\mathcal{MSE}}$	$\mathcal{L}_{\mathcal{NCE}}$	precRot	precInv	$\mathcal{L}_{\nabla}$	$\mathcal{L}_{\mathcal{INV}}$
Baseline	64.31	0.83	13e-2	1.81	62.19	28.21	0.92	1.38
Weight Scheduling	69.94	0.71	4e-3	1.80	65.49	25.31	0.87	1.39
Skip1	70.92	0.69	4e-3	1.79	66.63	24.32	0.84	1.39
Skip2	72.83	0.65	4e-3	1.79	65.92	26.12	0.85	1.38
Swap	72.88	0.65	4e-3	1.79	65.16	18.92	0.97	1.40
Skip3	81.32	0.45	5e-3	1.79	68.54	27.97	0.94	1.38
Resize	81.26	0.48	3e-3	1.64	77.53	38.40	0.58	1.33

Equation 1 then becomes;

$$\min_{\theta} \frac{1}{N*4} \sum_{i=1}^{N} \sum_{y=1}^{4} w_{i,y} \mathcal{L}_{\nabla \wr \sqcup}(F(X_{i,y}, \theta), y) \tag{2}$$

Mean feature of all the four different rotated feature maps is then used for NCE loss and Mean square error (MSE) loss between rotated feature maps and their mean feature, as they should be similar to one another. This MSE loss, $\mathcal{L}_{\mathcal{MSE}}$, is applied as distance objective function. Third loss function, $\mathcal{L}_{\mathcal{NCE}}$, is applied using NCE average criterion. In this method we tried classifying M different instances in a dataset. Each image instance is classified as a category of its own, allowing us to get more features against each image. This is individual instance classification for every image instead of set categories defined in a labeled dataset. We used 128 feature vectors against each image to classify the instance. For instance discrimination NCE was used to made this process less computationally expensive since for softmax operation of instance discrimination task, summation of entire datasets features in the denominator is time consuming for large datasets. Instance discrimination here aims to bring feature vectors of similar images closer and those of dissimilar images distant. NCE chooses a subset of samples from the dataset as negative samples, having feature vectors belonging to different instances. These samples are chosen by NCE having a probability of them being from negative samppling distribution. Distribution for negative sampling is obtained using binary classification by NCE. NCE then approximates the normalization constant for the softmax function of instance discrimination. The process of getting NCE feature vectors can be seen in figure 2.

The overall loss function then becomes as follows;

$$\mathcal{L} = w \cdot \mathcal{L}_{rot} + w \cdot \mathcal{L}_{MSE} + w \cdot \mathcal{L}_{NCE} \tag{3}$$

3.1 Modifications

Different modifications that I applied to the above defined method as part of this project consisted of three different parts, Architecture change for Deep Neural Network, Weight Scheduling and Image Augmentations implemented in form of transformations.

3.1.1 Architectural Change

I experimented with three different variations of architectural change to the original Alexnet ConvNet used to generate rotation feature maps for Noise contrastive estimation and rotation classification. Original AlexNet architecture used in baseline had 5 convolution layers each followed by batch normalization and ReLU non-linearity. These convolution layers have a projection head of fully-connected layer outputting 4096 size feature vector. Which is split into half for Rotation and instance classification. I applied skip connections to the architecture having following variations;

- For first experiment of skip connections, two different skip connections are added for five original conv layers of AlexNet. First skip connection connects output from conv layer 1 to conv layer 3 and second skip connection connects the output from conv layer 2 to conv layer 4. Down sampling for spatial size of feature maps was done using stride operation

Table 2: Top-1 linear classification accuracies on Tiny ImageNet training and validation set using activations from convolutional layers as features.

	Training Precision					Validation Precision				
Experiment	conv1	conv2	conv3	conv4	conv5	conv1	conv2	conv3	conv4	conv5
Baseline	40.61	46.17	50.65	48.91	45.87	15.79	23.24	26.20	25.39	23.67
Weight Routine	46.14	56.72	61.45	61.62	59.62	15.88	25.32	29.05	28.45	25.88
Skip1	45.59	58.15	61.40	63.12	60.96	16.11	24.64	27.99	28.31	26.47
Skip2	45.45	57.10	53.20	55.14	57.35	16.47	25.00	27.76	28.67	26.01
Swap	45.96	56.96	52.96	54.54	57.34	16.37	25.22	27.61	29.3	25.21
Skip3	44.68	59.86	62.47	61.43	54.59	11.43	19.21	21.51	20.57	20.51
Resize	28.80	49.60	63.41	64.71	67.94	17.43	25.09	30.55	31.38	30.54

in convolution and up sampling of channels was done by increasing the number of output channels using convolution operation.

- In second experiment, instead of using just conv layers to down-sample the filter size and up-sample channels, we use concatenation like DenseNet. Feature maps from Conv layer 1 of AlexNet and conv layer 3 are concatenated together and then down-sampled to an additional layer having channel number equal to the original conv 3 layer of AlexNet. I perform similar concatenation for conv layer 2 and conv layer 4 and inserted an additional conv layer in place of original conv 4 layer of AlexNet. Concatenation was done before Non linearity and after batch normalization bringing in the essence of ResNet with our skip connections.
- In third experiment, Two ResNet blocks were added after the first conv layer of baseline architecture. And similarly two ResNet blocks were added after second convolutional layer of baseline architecture.

3.1.2 Image Transformation

As we know image augmentation is an essential element in self-supervised learning. Different kind of image transformations were applied to input data samples before rotation. Transformations applied were random horizontal flip with a probability of 35%. Color jitter transformation was applied randomly with a probability of 40% with changing brightness, contrast, saturation and hue of an input image.

3.1.3 Weight Scheduling

Baseline architecture uses weight of one for all three loss functions in equation 3. I have used weight scheduling where all three weights differ during the training. Total epochs are divided into three equal portions. In first part of training, ramp function is used to generate weights for Rotation classification. Weights for MSE loss and instance classification are very minimal at this stage. In second stage, feeding in the learned features from RotNet we try to bring all the rotated feature maps against an image closer by giving larger weightage to distance loss. In last stage now learned features from rotation and MSE loss are fed into Noise Contrastive learning for instance classification. Weight schedules were calculated using ramp function routines. They can be seen from figure 5.

4 Experimentation

Dataset used for our experimentation in accordance with the time constraint is Tiny Imagenet. For experiments, I have used AlexNet architecture and 128 instances classification. Learning rates and loss criterion are used similar as Gidaris et al. [2018], the baseline paper. Experiments and their training results are listed in table 1. Results for these trained models from table 1 were tested by attaching linear classification heads after each convolution layer to assess the learnt feature maps at each layer in depth. Results for Linear classification heads attached at the end of conv layers are listed in Table 2. Results for non linear classification heads attached at the end of conv layer 4

Table 3: Top-1 Non linear classification accuracies on TinyImageNet training and validation set using activations from last two convolutional layers as features.

	Train		Test	
Experiment	conv4	conv5	conv4	conv5
Baseline	99.97	99.81	32.93	31.66
Weight Routine	99.97	99.98	38.26	35.41
Skip1	99.99	99.97	38.18	36.24
Skip2	99.95	99.96	40.46	34.80
Swap	99.96	99.96	41.33	34.59
Skip3	99.98	99.93	28.23	27.21
Resize	99.37	99.48	41.99	41.09

and 5 are listed in table 3. Nonlinear classification is only tested for later layers since earlier layers performance does not contribute much to non linear feature heads.

For baseline experiment, I have converted AlexNet architecture by changing filter size for first conv layer to fit in with input image size of 64x64x3. Baseline code was written for Imagenet, changing it to tinyImagenet required me to change the dataloader for validation set images. All the other configurations were kept same as Gidaris et al. [2018]. Experiments listed in Table 1-3 are done incrementally, thereby incorporating change from previous experiment into next one for some of the experiments. Second Experiment, Weight Routine, introduces weight scheduling scheme to the baseline experiment, where rotation classification is learnt in first phase, rotation feature similarity is enforced in second phase and in last phase Instance classification is learnt. Accuracies are improved with 2-3% with linear and non-linear classification heads as seen in table 2 and 3. For Skip1 experiment architecture was changed on the top of weight scheduling. Here two additional convolutional layers are added as skip connections to add the feature maps from layer earlier layers to the later layers. Skip connections were added using convolution. Results improved further using these skip connections. Skip2 uses concatenation as described in method section to add skip connections. Results show that results for conv4 improve for both linear and non linear classification.

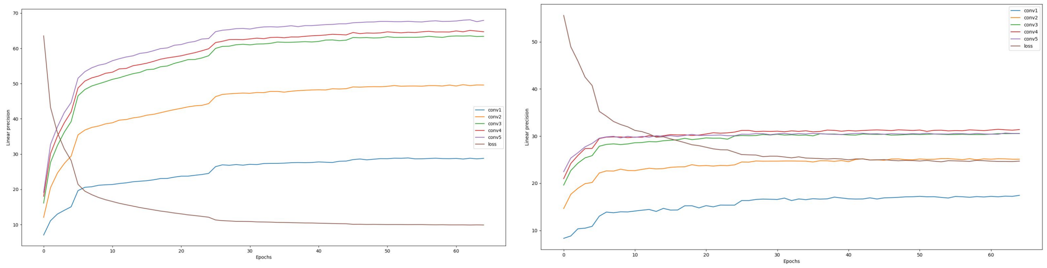
In Swap experimentation, I have broken down the 4096 feature vector from AlexNet into 4 parts whereas baseline breaks this into 2 parts, assigns one to rotation head and one for instance learning. I swap four parts in even and odd pairs, concatenate them back in the form of two vectors of size 2048 and feed them to rotation head and instance classification head. Intuition behind swapping was to ensure that low level features learnt are shared among the two networks invariant to the order of logits in 4096 size vector. Results show improvement after swap operation for both linear and non-linear classification.

For Skip3 on top of weight scheduling and swap operation, I added 4 conventional ResNet blocks to the AlexNet architecture as explained in the method section. Accuracies dropped in this experimentation. It might seems that the accuracies here are likely affected by over-fitting.

In last experiment named Resize, skip connections were removed, Image transformations were applied to the dataset as described in method section. Image was resized and random cropped in the size of original Imagenet images. This allowed to use AlexNet with same architecture as Gidaris et al. [2018]. I have also increased the number of instance classes here as 248, instead of 128. Although it is evident that artifacts must be introduced to the dataset with resizing with scale of four but results still shows considerable improvement from all the previous experiments as shown in table 1-3. This experiment had image transformations with resize and random crop, weight scheduling and swap operations applied along with basic configurations as discussed previously.

5 Discussion

I drew certain insights with these experimentations performed on top of baseline. Given the limited time constraint for this project, experiments revealed that their is great margin of improvement in future work. Such as, my experiments were motivated and guided towards decreasing the gap between training and testing accuracy, but it seems their is still a large gap to overcome, as can be seen by



(a) Linear Classification results for training

(b) Linear Classification results for training

Figure 3: Linear classification trends are shown for Experiment Resize

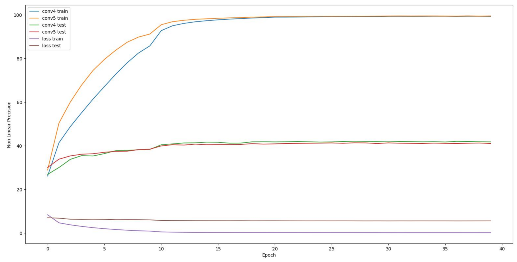


Figure 4:
Non Linear Classification trends shown for Experiment Resize

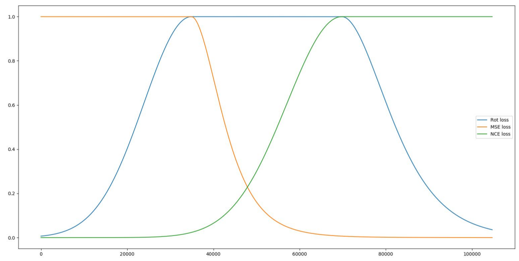


Figure 5:
Weight Scheduling Scheme Introduced for Loss functions

Figure 4, with injecting further techniques similar to image transformation such as 5-crop or 10-crop schemes. Accuracy of instance classification lags behind with a larger margin. If we combine it with feature maps from some other pretext task also, it would improve further. Another thing that I noticed in code was that for validation of rotation classifier and instance discrimination classifiers, supervisory signal or self generated signals were used as 90 degree offset angles. Using it for rotation is justified, but for instance classification which we are trying to decouple from rotation features, it might not be a good criteria to calculate the performance of this classifier based on its prediction for the angle of rotation applied to the input image. Additionally we have seen that all the experiments conducted in this paper have improved over the baseline and when different methods were combined for the last experiment, accuracies improved the most.

6 Conclusion

Objective of this paper was to apply modifications on top of the baseline method by Gidaris et al. [2018] which tries to decouple rotation classification from instance discrimination in an unsupervised way. We have performed experiments on Tiny Imagenet by applying changes to the AlexNet architecture, adding weight scheduling scheme for the loss functions, increase individual instance classes and applying Image transformations to the input dataset. It can be seen that additional experiments improved the results but accuracies would not change with a larger margin with each incremental addition. In future work, different approaches can be tested to first decrease the overfitting or the training and testing precision gaps. One way to do this is to generate augmented copies of the image thereby increasing the dataset size instead of applying just the image Transformations to the input image as done in this project, due to computational gains associated with larger datasets.

References

Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 179–189. IEEE, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015. URL `http://arxiv.org/abs/1505.05192`.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *CoRR*, abs/1605.09782, 2016. URL `http://arxiv.org/abs/1605.09782`.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin A. Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *CoRR*, abs/1406.6909, 2014. URL `http://arxiv.org/abs/1406.6909`.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. pages 213–220, 08 2008. doi: 10.1145/1401890.1401920.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018. URL `http://arxiv.org/abs/1803.07728`.

Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Multi-fold mil training for weakly supervised object localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2409–2416, 2014.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Y.W. Teh and M. Titterington, editors, *Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, volume 9 of *JMLR W&CP*, pages 297–304, 2010.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. JMLR Workshop and Conference Proceedings. URL `http://proceedings.mlr.press/v9/gutmann10a.html`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722, 2019. URL `http://arxiv.org/abs/1911.05722`.

Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In *Advances in Neural Information Processing Systems*, pages 10758–10767, 2019.

Azam Kaboutari, J. Bagherzadeh, and F. Kheradmand. An evaluation of two-step techniques for positive-unlabeled learning in text classification. *International Journal of Computer Applications Technology and Research*, 3:592–594, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Chen Li and Xueliang Hua. Towards positive unlabeled learning for parallel data mining: a random forest framework. In Xudong Luo, Jeffrey Xu Yu, and Zhi Li, editors, *Advanced Data Mining and Applications*, Lecture Notes in Computer Science (LNCS), pages 573 – 587. Springer, 2014. ISBN 9783319147161. doi: 10.1007/978-3-319-14717-8. URL `https://link.springer.com/book/10.1007/978-3-319-14717-8`. International Conference on Advanced Data Mining and Applications 2014, ADMA 2014 ; Conference date: 19-12-2014 Through 21-12-2014.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Neural and Information Processing System (NIPS)*, 2013. URL `https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf`.

Fantine Mordelet and Jean-Philippe Vert. A bagging svm to learn from positive and unlabeled examples, 2010.

Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *AISTATS'05*, pages 246–252, 2005.

Nhu-Van Nguyen, Christophe Rigaud, and Jean-Christophe Burie. Semi-supervised object detection with unlabeled data. In *international conference on computer vision theory and applications*, 2019.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. *CoRR*, abs/1603.09246, 2016. URL `http://arxiv.org/abs/1603.09246`.

Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016. URL `http://arxiv.org/abs/1604.07379`.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. URL `http://arxiv.org/abs/1807.03748`.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning proceedings*. 2008.

Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination, 2018.

Chunjie Zhang, Jian Cheng, and Qi Tian. Unsupervised and semi-supervised image classification with weak semantic consistency. *IEEE Transactions on Multimedia*, 21(10):2482–2491, 2019.

Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. *CoRR*, abs/1603.08511, 2016. URL `http://arxiv.org/abs/1603.08511`.

Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.