# Master Thesis Ideas

1. Compare Self-Supervised approaches (Adversarial Representation Learning, Contrastive Methods, Siamese Networks without negative pairs)
2. Similar to 1: Analysis of Self-Supervised approaches for computer vision and improvement based on the analysis? (Special focus on analysis and avoidance of collapse?) + Different architectures (Joint-Embedding Architecture, Generative Architecture, Joint-Embedding Predictive Architecture, Reconstruction (Autoencoders))
3. Improve existing methods (One or two(?) of the above)
   1. How can one as simple as possible avoid collapse? What actual criteria determine whether there is a collapse? -> SimSiam does not explain why collapsing is prevented! (SimSiam 5.3) ->How CAN it be prevented?
      1. Temperature parameter in the loss like in (<https://arxiv.org/pdf/2204.07141.pdf> ) -> End of page 5
      2. ME-Max regularizer? -> Tries to maximase the entropy (More in <https://arxiv.org/pdf/2204.07141.pdf> beginning page 6)
   2. Figure 3 in SimSiam page 8 -> Can we remove the predictor?
      1. Predictor itself is not the primary reason it collapses (though it helps in prevention), because if stop-gradient is removed, it still collapses (SimSiam above 4.2)
      2. Predictor can be removed, if we use a moving-average similar to maintain a memory bank (E.g. MoCo, Unsupervised Feature Learning via Non-Parametric Instance Discrimination)
      3. We need to approximate the distribution over the possible augmentations (SimSiam end of page 6)
4. VT Autoencoders -> Collapse can’t happen (<https://arxiv.org/pdf/2111.06377.pdf> )

* Create the same representation for different augmented versions of the same image?
* No option should jointly minimize the same loss function? Would lead to collapse? See BYOL 3.2
* Make undesirable equilibria unstable (BYOL beginning page 5)
  + Somehow prevent the network from just outputting the same value
* Use ME-MAX regularization? (Masked Siamese Networks for Label-Efficient Learning Appendix B <https://arxiv.org/pdf/2204.07141.pdf> )
* For comparison like the following use own reproduction of other approaches if possible

A graph with lines and numbers

Description automatically generated

## Things to consider

* LSTMs and CNNs are biased (inductive bias) -> They steer the network into a solution direction
  + Transformers are more general! (<https://arxiv.org/pdf/2010.11929.pdf> page 2)
* Self-Supervised on uncurated datasets? -> Not on datasets originally created with labels (Introduction of SEER <https://arxiv.org/pdf/2103.01988.pdf?fbclid=IwAR2pqhYda6MV9r2b3Afx_0eKUiZhX-Es6Pa_FbLOqH8fglQzO2kY3yKxZE8>)
  + Should also be class imbalanced, not like in ImageNet-1k where all classes are balanced (<https://arxiv.org/pdf/2210.07277.pdf> )
* **Result should also be a fast learner (low shot) -> Use (extreme) low-shot dataset?** (<https://arxiv.org/pdf/2112.12004.pdf> ) (<https://arxiv.org/pdf/2204.07141.pdf> Introduction and Table 1)
* MoCo does not enforce sparsity of latent variable (in this case the encoding of the image/patch), and does not limit the capacity?
  + Isn’t this important?! -> <https://ai.meta.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/>
* Do proof of concept and finalization for all ideas on cifar10 (or even cifar100?)
  + Validate with ResNet-50 or the usual benchmark framework used for self-supervised learning
* **Use Vision Transformer as masking (mask tokens) is easier?** (<https://arxiv.org/pdf/2111.06377.pdf> Intorduction)
* For Data Augmentation on one example to create two (like in e.g. BYOL)
  + The type of augmentation used is essential for the success
  + Random crop, at the beginning of the training the Intersection Over Union can be higher, but is gradually lowered the further we progress in training? -> Makes it harder and harder
  + Also try augmentation techniques from “On the use of Cortical Magnification and Saccades as Biological Proxies for Data Augmentation” (<https://arxiv.org/pdf/2112.07173.pdf> )
* Do prediction of color channel? -> Colorization task?
  + <https://arxiv.org/pdf/1611.09842.pdf>
  + Learns that e.g. trees are green etc.
* Multi-objective task? Like colorization, patch prediction etc? Change up the tasks? Does that make sense?
* Standard deviation of the output should be 1/sqrt(d)? Indicates no collapse (See SimSiam 4.1)

## Ideas of papers

### Unsupervised Feature Learning via Non-Parametric Instance Discrimination

* Contrastive method
* Use a memory bank with “embeddings” of images
* Each instance is seen as one class
* Similar to softmax process in word embeddings
* Update the embedding for each image (the memory bank) in each iteration
* Originally:
  + Softmax over all embeddings of the of the mini batch size
    - Cosine similarity (dot product and all embeddings l2 normalized) for the embedding of the current image should be the same as the one in the memory bank -> 1
    - Should be 0 for all other instances the training is done (not the same instance i.e. class)
* But softmax over this many instances costly
* Do the same as in word embeddings:
  + Binary target
  + Is the embedding from the “real distribution”? 1 example per batch
  + Means does it belong to the i-th image in the memory bank? -> So the embedding of the same image, the network just produced a new embedding for, from the previous iteration
  + Or is it from the “fake distribution”, so the embedding of another image of the memory bank (prev iteration)? (k-1 samples of the batch
* Still softmax used, but approximated using monte carlo
* Which negative samples are used is randomly sampled
  + Samples are updated when they were last seen -> can be outdated, which makes separation easier? Fixed by MoCo (3.2 Relation to previous mechanisms)!
* Loss includes a factor that ensures that the embeddings of each image stabilizes over the iterations, meaning the distance between embeddings of the same image from the current iteration and the previous one should go towards zero

### MoCo

* What is the use of the queue exactly? In the pseudo code it is only used to update the dictionary? And how is it used then?
* Why is the moving average (momentum) encoder so important/successful? Is also part of BYOL

### BYOL

* Neural Network q tries to predict the representation of an augmented version of an image from the representation of another augmentation of the same image
  + Tries to negate the augmentation
  + Aim of augmentation is to destroy all the non-semantic information
* q(fΘ(A(x)=q(f𝛆(A’(x)))
  + q should negate difference between augmentations and the parameters of the network(s)
* Types of augmentations have high influence on the result!!!
* Maybe a better approach is to try to create the same representation for different augmented versions of the same image?
  + Augmentations shouldn’t be too hard, but should destroy all non-sematic features, while keeping all semantic features
  + Careful with collapse! -> Use moving average (MoCo and BYOL) and stop gradient (SimSiam and BYOL)?
  + Avoiding collapse is the most difficult/important problem to solve (prevent the network from just outputting the same value)
* Avoiding collapse:
  + Should be easier to create a good representation than to collapse
  + Appendix H
  + If q is optimal, this means that the expected value of q, if the input is z, so the output of the online network, is z’, the output of the target network
    - Means the mean squared error would be close to 0
    - Practically this means that the predictor q can predict the latent variable of the same image, on which the other augmentation t’ was applied, based on the latent variable of the other augmentation t on the same image.
    - The representation needs to be predictable, so that q can predict the other representation
    - What if f is actually outputting the same representations from t(x) and t’(x)? **Does q actually do something?** Check in paper and practically!
      * Wouldn’t it be better if f does this directly?
    - f needs to learn to ignore the augmentations and produce very similar representations for t(x) and t’(x) so that q even has the chance to predict z’ from z! This is because z can’t possibly know what augmentation t’ was used, e.g. how t’ and t differ. Do we even need the moving parameters? -> S**imSiam**?
  + => E[z’|z], target ist z’
  + Mit der folgenden Beziehung des MSE zur Varianz folgt, dass wir in die Richtung der minimalen Varianz gehen, die Abweichung von z bezüglich z‘ soll also minimal werden
  + A white background with black text

    Description automatically generated
  + A math equations on a white background

    Description automatically generated
  + If the predictor is optimal, then there are no gradients for q anymore
  + The only gradients that are flowing through the system are those that are optimizing the encoder f?

### SimSiam

* Goal is invariance of two observations (z.B. augmentations) if the input
  + Network should output the same representation for both observations
  + Means ignoring all non-semantic content
* Siamese networks do this naturally (Vergleiche Anfang Seite 2 SimSiam paper)
* Idea formulated above seems to not work (SimSiam 4.2)
* 5.1:
  + Theta (Parameter) are cluster centers (centroids), ηx is the one hot vector (indicating to which cluster x belongs?)
  + Equ. 7: First the new cluster centers θt are created from the members ηx of the clusters from the previous iteration(t-1)
  + Equ. 8: Then ηx are updated based on the new centers θt
  + Stop-gradient on ηx means we only update the centers?
  + Stop-gradient on θt means we only update the samples x?
  + Solving for ηx : First distribution only over the augmentations T, as θ is fixed due to the stop-gradient
  + One-step alternation: Augmentation is only samples once, so expectancy is ignored, but because we have an alternation, meaning we do it over and over (just for different x), we have an approximation of the augmentation distribution -> **So it is a lot about the augmentations used!?**
  + Predictor output also goes over distribution of augmentations
    - Goal is to compute the expectation over the augmentations (end page 6)
      * **Shouldn’t that be the task of F? Can F also do this task?**
    - Sampling of T, so the distribution(?), is created because of the multiple epochs with different augmentations

### I-JEPA

* Is not view-invariant pretraining like SimSiam, BYOL etc. (Aim at those models is to create invariant embeddings -> Learning goal is the representation of an image should be the same even if create a new view through augmentation like cropping=
* They also use an exponential moving-average for the target encoder, like in BYOL and SimSiam!!!

What if we prevent collapse by continuously changing/adjusting the objective?

* Humans learn concepts not only from doing the same task over and over again
* Patch prediction
* Context prediction
* View-invariant representation
* Would probably mean adjusting the prediction head and continuously training the backbone?

Main difference generative models and view-invariant models: Loss in generative models is in input/image space, for view-invariant models in representation space

### Masked Siamese Networks for Label-Efficient Learning

* Uses maximation of entropy in the loss to avoid collapse (ME-Max regularization)
* Is based on patch prediction
* Uses normal cross-entropy
  + Creates distributions for this by measuring the similarity of each patch/image representation with K prototypes
  + Penalizes if predicted patch representation does not have the same distribution, after similarity comparison with prototypes, compared to the target
* Proof for avoiding collapse (Appendix B)
* Uses Siamese Network with VisionTransformer architecture and exponential moving average of predictor as the target network

### Masked Autoencoders Are Scalable Vision Learners

Page 3…

### Data2Vec

* Predicts latent representations
  + Are contextualized -> Incorporate relevant features from the entire image, instead of only the e.g. patch that is to be predicted
  + Because outputs of layers are predicted, the targets contain a context, because they are created using self-attention blocks of the teacher model
  + BYOL and SimSiam learn transformation-invariant representations
  + No structural information learned (Data2Vec Chapter 6 end)