Tasks left:

1. End to End pipeline - DL object detection (bounding box resizing, bounding box size positive, IoU)
2. Resnet 50 update
3. Channel size

Model base: Convert all the models to base resnet 50.

Bounding box 1*4 format : [x_min, y_min, x_max, y_max]

Algorithm for making bounding box positive -

For the x values(check for both values separately] :
If x_min = 0, it remains 0.
If x_min > 0, then x_min = 462 + (x_min/40)*462
If x_min < 0, then x_min = 462 - (|x_min|/40)*462
Do the same for x_max also.
For the y values, do(separately for y_min and y_max) :
If y = 0, it remains 0,
If y > 0, y = 256 + (y/40)*256
If y < 0, y = 256 - (|y|/40)*256

Architecture plan:

I) Semi Supervised Labelling (Self supervised learning SIMCLR (10% of unlabelled data set) -> 90% dataset) - tau (temperature) (part of cross entropy)) ->
Use Transfer learning from Unlabelled dataset ->Labelled DataSet (i) Measure performance.

ii) Get Representations from Unlabelled dataset (SIMCLR) -> add linear layer in the end for classification.

Self supervised:

1. Make code modular - plug and play.
2. Make an end to end pipeline.
3. Data Augmentation
4. Custom Loss function
5. GANS, Autoencoders, ensemble

6. Network Architecture
7.

+ Implement paper baseline
+ Then improve incrementally
+ Ask katherina
+ Read papers

Things to keep in mind

1. Class imbalance
2. Learning Rate scheduler
3. Selection of loss functions
4. Normalization
5. Random seed and reproducibility

Some key things:

Self supervision
Upsampling
convTranspose2D

Semi supervised learning project:

Dataset Analysis:

- **Bounding boxes**
-

Tasks:

- Object Detection and classification.
- Lane Detection
- Coordinate and frame mapping.

Semi tasks:
- Exploratory data analysis of annotation.csv

Understanding the required

Literature Survey:

https://towardsdatascience.com/lane-detection-with-deep-learning-part-1-9e096f3320b7
https://github.com/mvirgo/MLND-Capstone
https://github.com/imran3180/SSL-Image-Classification (Semi Supervised pipeline)
https://github.com/aniket03/pirl_pytorch (PIRL)
https://github.com/aniket03/self_supervised_bird_classification (Self supervised)
https://github.com/chandu-97/fair-sslime (Self Supervised)
https://github.com/abvnithish/self_supervised_machine_listening (Self Supervised)
https://towardsdatascience.com/tutorial-build-a-lane-detector-679fd8953132
https://www.kaggle.com/soumya044/lane-line-detection/comments#60795
Semantic Segmentation - ie road detection
1.https://towardsdatascience.com/semantic-segmentation-with-deep-learning-a-guide-and-code-e52fc8958823
2.
https://www.learnopencv.com/pytorch-for-beginners-semantic-segmentation-using-torchvision/
3.https://github.com/CSAILVision/semantic-segmentation-pytorch
4. Pytorch -
https://pytorch.org/docs/stable/torchvision/models.html#semantic-segmentation
5. https://paperswithcode.com/task/object-detection

Self supervised learning:
https://www.fast.ai/2020/01/13/self_supervised/
https://amitness.com/2020/04/illustrated-self-labelling/?utm_source=feedburner&utm_medium=email&utm_campaign=Feed%3A+amitness+%28Amit+Chaudhary%29

Papers
https://github.com/amusi/awesome-lane-detection
https://towardsdatascience.com/tutorial-build-a-lane-detector-679fd8953132

https://youtu.be/APki8LmdJwY (SIMCLR)

To check:

“Monocular 3D Object Detection in Autonomous driving” by Patrick Langechuan Liu
https://link.medium.com/tOwX7jGpJ5

https://arxiv.org/abs/1801.09057

Nbs for help:

https://colab.research.google.com/github/pytorch/vision/blob/temp-tutorial/tutorials/torchvision_finetuning_instance_segmentation.ipynb

Models:

1. Generative models with latent variables.

Questions:

1. Like if the labelled dataset is generated as per time frame, is it suffering from bias?
2. Lane generation - 6 cameras? - Road map generation.
   Every sample -> 800 *800(10 degree of overlap between the cameras). The six cameras will generate 60 degrees of the lane. Generate 6 tensors and combine.

   Combine these 6 images into a single image. Remove overlap, and then do lane detection - 800, 800.

   Else jigsaw for self supervision, learn from shuffling -> Invariant, generate lanes.

   Tasks:
   1. Build pipeline
   2. Object Detection and classification
   3. Literature Survey

Multi Task Self supervised learning (Ensemble of various self supervised learning)

NB Links:

EDA:

https://colab.research.google.com/drive/10KpO8oJ_vToBmTlvkLIfDyP2pzSJwpxw#scrollTo=DBrvwXuA2amA

Things to figure out:

1. **Image combining - 6 images**
2. **Bounding box detection and classification - with these 6 images and labels being these 8 classes with boundaries.**
3. **Road_image generation (ego image generation)**

Using the problem statement:

Using unlabelled data for pretraining (Semi supervised - Self Supervised) (references):

https://towardsdatascience.com/google-open-sources-simclr-a-framework-for-self-supervised-and-semi-supervised-image-training-72b06d5d58a0

The goal of self-supervised and semi-supervised learning methods is to transform an unsupervised learning problem into a supervised one by creating surrogate labels from the unlabeled dataset.

Discriminative: These methods learn representations using objective functions similar to those used for supervised learning, but train networks to perform pretext tasks where both the inputs and labels are derived from an unlabeled dataset. Many such approaches have relied on heuristics to design pretext tasks, which could limit the generality of the learned representations.

Contrastive learning can be defined as the ability to learn representations by enforcing similar elements to be equal and dissimilar elements to be different.

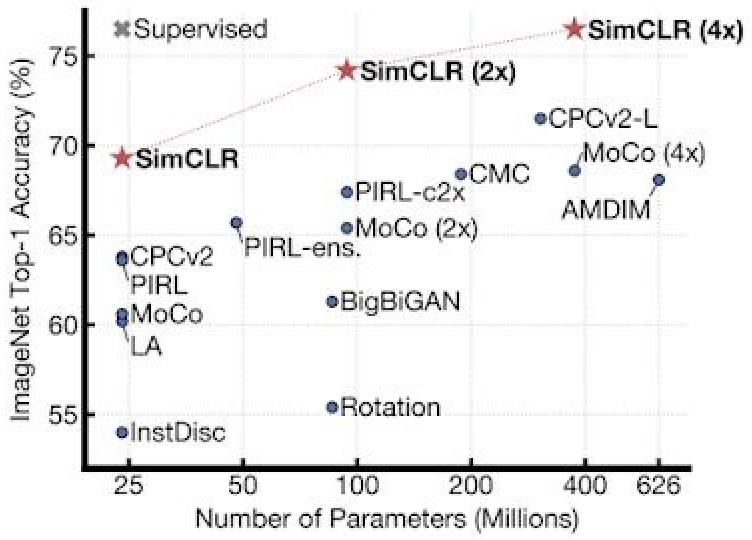
The key idea behind contrastive learning is to expand learning beyond high level features. For instance, suppose that we are trying to determine differences in the tail of two images of whales. Sometimes high level features such as color and size are not sufficient because they are going to be too similar in the different pictures. Contrastive learning, combines those high level features with local and global features that can contextualize the analysis in order to determine similarities and differences between images.

SimCLR

Building on the principles of contrastive learning, SimCLR provides a model that learns representations by maximizing agreement between differently augmented views of the same data example via a contrastive loss in the latent space.

Comparison of various techniques/Analysis:

(I have seen PIRL ka code)
(but I think SimCLR is more advantageous)



:

https://ai.googleblog.com/2020/04/advancing-self-supervised-and-semi.html **(Reference)**

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 epochs7 (He et al., 2016).

https://github.com/google-research/simclr **(github link)**

https://arxiv.org/pdf/2002.05709.pdf (paper)

SimCLR

Title:	**A Simple Framework for Contrastive Learning of Visual Representations**
Authors:	Ting Chen, Simon Kornblith, Mohammad Norouzi, Geoffrey Hinton
Blog:	Arxiv Link
Read:	✓
Referee:	

Motivation to care about this paper

After training their network in an unsupervised fashion on images. A linear layer trained on top of their frozen network got 76.5% top 1 accuracy on ImageNet. This is a huge deal as previous unsupervised approaches only got up till about 69% even after training on over 1 billion images.

Approach

1. For a batch of N images, each of them is augmented twice resulting in an overall batch size of 2N.
2. The loss used is a simple negative log loss, with an added temperature parameter. The intuition is to maxmiise similiarity betweeen the feature of an image augmented versions. The rest of
3. 2N−1
4. samples (removing the self-similarity) are treated as negative samples.
5. l
6. ij
7. =−log
8. exp[(sim(
9. z
10. i
11. ,
12. z
13. j

14.)/τ)]
15. ∑
16. 2N
17. k=1
18. 1
19. k≠i
20. exp[(sim(
21. z
22. i
23. ,
24. z
25. k
26.)/τ)]
27. wehre
28. sim(u,v)=
29. u
30. ⊤
31. v/(∥ u ∥ · ∥ v ∥)
32. .
33. No memory bank needed, no dictionary needed. Very simple formulation. Here the
34. z
35. terms are a non linear transformation of the feature representations of the image. Hence, first we input the augmented image into the neural net to get a hidden representation
36. h
37. . Then we feed that image into a 1 layer neural network (that means a matrix multiplication, a relu and another matrix multiplication) to get
38. z
39. .
40. The authors trained this network on a very large batch size, they used a batch size of 8000 images, hence getting an effective batch size of ~16000 images. (2N). They were able to do this using the LARS optimiser that has different learning rates for different layers of the neural net. Using this optimiser is necessasary for huge batch sizes. They realised that a large batch size makes a big difference.
41. Another thing that has proved to be a game changer is to optimise contrastive losses on this
42. z
43. representation instead of the
44. h
45. representation. Turns out contrastive losses work best on a space that is different from the feature representation space.
46. The image augmentations used are a simple random resize and crop, colour distortions, and gaussian blur. One crucial finding is that mixing colour distortion with random crop and resizing is crucial. This is because just random crop and resizing on their own are not a hard

enough task to model under this contrastive loss. Another big finding is that the augmentations are more effective for contrastive losses, even though they don't yield accuracy benefits in a supervised setting. The bigger the augmentation, the better for unsupervised models.

47. They used the SGD optimizer and the learning rate is warmed up first. Starting from a very low quantity, it is gradually increased to some quantity in a short amount of epochs. Then the rest of the training the learning rate is gradually decayed. They used Resnet50 as their base encoder but their 76.1% result came from a network 4 times that size. The unsupervised results came out with 1000 epochs training. Supervised training models trained for 100 epochs. TLDR: Unsupervised models take longer to train under contrastive loss settings. One cool finding is that supervised nets (same # of params) trained for 90 epochs are ~2% better than unsupervised nets trained for 1000 epochs. Supervised nets are much more data efficient!
48. Big finding though not surprising: *Unsupervised learning benefits more from bigger models than its supervised counterpart.*
49. Short and sweet conclusion: *Our results show that the complexity of some previous methods for self-supervised learning is not necessary to achieve good performance. Our approach differs from standard supervised learning on ImageNet only in the choice of data augmentation, the use of a nonlinear head at the end of the network, and the loss function. The strength of this simple framework suggests that, despite a recent surge in interest, self-supervised learning remains undervalued.*

Self supervised:

https://piazza.com/class/k5spqaanqk51ks?cid=28

"**Downstream tasks** are what the field calls those supervised-learning tasks that utilize a pre-trained model or component". (The model is generated through training a pretext task on an unlabelled dataset (Self supervised)).

https://arxiv.org/pdf/2003.04297.pdf (improving SIMCLR baselines)
https://arxiv.org/pdf/1902.06162.pdf (Theory)

Applying the MoCo update rule on SimCLR techniques (MLP head, more augmentation, more epochs) imrpoves performance. - Soham Tamba

Questions:

Can we use pretrained model to generate the auxiliary label / info? such as the bounding box in the camera image? (piazza) (https://piazza.com/class/k5spqaanqk51ks?cid=235)

3D monocular object detection: (Distance measurement)

https://arxiv.org/pdf/1801.09057.pdf
https://towardsdatascience.com/monocular-3d-object-detection-in-autonomous-driving-2476a3c7f57e

Semi Supervised learning:

Semi-supervised learning is a domain in which generally you will get a small labeled dataset and a large unlabeled dataset, and You have to make use of both labeled and unlabeled data to train the model.
There are several methods to approach to solve the semi-supervised learning.

--- Proxy method
--- distillation method
--- and many more

What you have described is an example of Proxy method and called self-training, in which the model adds samples from unlabeled data into a labeled dataset using its own prediction if it is more confident than a threshold.

The threshold depends upon the task, dataset. usually, its hyperparameter for which you will have to find the optimal value.

1. Distillation ((https://github.com/karanchahal/distiller)
2. Proxy methods (https://github.com/imran3180/SSL-Image-Classification)

Architecture Reference:

1. Youtube Video: https://youtu.be/APki8LmdJwY (SIMCLR)
2. Papers: SIMCLR - https://arxiv.org/pdf/1902.06162.pdf (self supervised Theory)
   - **https://ai.googleblog.com/2020/04/advancing-self-supervised-and-semi.html**
   - **https://arxiv.org/pdf/2002.05709.pdf (paper)**
   - https://arxiv.org/pdf/2003.04297.pdf (TAMBA IS USING THIS-> MOCO with SIMCLR which doesnt need too much GPU) See piazza link
   - https://towardsdatascience.com/google-open-sources-simclr-a-framework-for-self-supervised-and-semi-supervised-image-training-72b06d5d58a0
   - http://openaccess.thecvf.com/content_ICCV_2019/papers/Zhai_S4L_Self-Supervised_Semi-Supervised_Learning_ICCV_2019_paper.pdf
   - **https://arxiv.org/pdf/1911.05722.pdf (Moco)**
   - **https://arxiv.org/pdf/1708.07860.pdf (Multi task)**
   - **https://arxiv.org/pdf/1905.01235.pdf (Scaling and Benchmarking Self-Supervised Visual Representation Learning)**

3. Github:

   - https://github.com/imran3180/SSL-Image-Classification (Semi Supervised pipeline)
   - **https://github.com/google-research/simclr (SIMCLR github link)**
   - https://github.com/karanchahal/distiller **(Distillation)**
   - **https://github.com/sthalles/SimCLR (pytorch)**
   - **(https://sthalles.github.io/simple-self-supervised-learning/) blog to above github**
   - **https://github.com/mdiephuis/SimCLR (pytorch)**
   - **https://github.com/leftthomas/SimCLR (pytorch)**
   - **https://github.com/google-research/dads (pytorch)**
   - **https://github.com/Spijkervet/SimCLR (pytorch)**
   - **https://colab.research.google.com/drive/1ObAYvVKQjMG5nd2wIno7j2y_X91E9IrX#forceEdit=true&sandboxMode=true**
   -

4. Piazza (big 4 thinking, self supervised)
   https://piazza.com/class/k5spqaanqk51ks?cid=28

5. To think more:

Pretext task:
3D monocular object detection:
Latent Variable models

6. Dataset:
   http://ai.stanford.edu/~acoates/stl10/

ISSUES:

Tasks Left:

1. **Setting up Hpc accounts (Lakshmi)**
2. **Making code modular**
3. **Baselines establish (Mostly Resnet 50) - Gowri**
4. **Rnd Thorough Analysis (Latest SOTA for Model analysis) - Gowri & Amartya**
5. **Plotting and analysis**
6. **Hyperparameters**
7. **Running Huge models (Infrastructure Analysis) - Lakshmi will give guidance (resnet 50 / Resnet 101 / TPU analysis)**

Factors to test for:

1. Widening of network (Resnet backbones - width/ GPU work)
2. Parallel processing (Data.parallelize)
3. Ratio of Datasets
4. Hyperparameters(Optimizers, LR, epochs, test/validation split)

Architecture plan:

1. Ensembled Self supervised Learning - SIMCLR, MOCO, PIRL, Jigsaw, Rotation { Someone needs to see how to merge various classification model layers)
2. Semi Supervised Learning (Pseudo learning)
3. Transfer Learning -> Supervised learning
4. Metric evaluation

We can do:

1. Self Supervised -> Semi Supervised -> Ensemble Supervised Learning
2. Ensembled Self Supervised (multi Task Self supervision) -> Semi Supervised -> Ensemble Supervised Learning

Future:

1. 3D monocular object Detection
2. Latent Variable model

Issues:

1. Prediction shape [1*4] -> [2*4]
2. Iou metric Analysis

Deliverables:

1. PPT (Lakshmi)
2. Paper (All)
3. Video (Amartya and Gowri)

Frameworks for Self Supervised Learning

1. **Applying the MoCo update rule on SimCLR techniques (MLP head, more augmentation, more epochs) imrpoves performance.** https://arxiv.org/abs/2003.04297 **(Guess we don't need 512 TPUs)**

2. **SIMCLR**
3. **PIRL**

Semi Supervised: Distillation/pseudo labelling (2 tasks)

Supervised/ Ensemble and Transfer:

1. **Train our models on 3 of the Self supervised framework and use ensemble of 3 supervised tasks -> (Ensemble of 18 Models) -> Accuracy:**

Xgboost, Logistic Regression and 1 More

FRRN B (Road image)
https://github.com/HRNet/HRNet-Object-Detection

3 self supervised representation * 2 semi supervised (optional) * 3 classification

Benchmarking and accuracy:

https://github.com/facebookresearch/fair_self_supervision_benchmark

To see:

Mean Average Precision (mAP) - evaluation of pretext task on downstream task
Diff between cluster fit and model distillation?
whats. diff between PIRL and SIMCLR?
Generative models with SimCLR (Gans/Autoencoders) - (Contrastive methods EBM)
Semisupervised and self supervised PIRL MOCO SIMCLR
Distillation?

Box coordinates confirmation (Y - 1st tensor, next tensor is x coordinate)
Y we have to make 2*4 -> 1*4 (the neural network can make any shape)

Not a standard dataset (as previous one)

-> Bounding boxes can be upside down (Need to check, Iou Code/ reshape code)
->

What is threat score w.r.t object detection and Iou

Threat score calculation:(Have to see this)

Sending him architecture:
Ask doubts:

Rubrics

Full-Resolution Residual Networks (FRRN) https://github.com/meetshah1995/pytorch-semseg
https://pypi.org/project/pytorch-semseg/
https://github.com/mrgloom/awesome-semantic-segmentation

https://github.com/PKUbahuangliuhe/CBNet
Super Resolution
Variational Auto Encoder

Project Proposal:

Hi All,

We went through the competition and were able to build basic models for both road map prediction and object detection using simple resnet18 for the former and faster RCNN for the later. We are now planning to move further in the project using self and semi supervised learning approaches and wanted guidance on our approach to see whether we are in the right direction.

Post a lot of literature survey and reading a lot of state of art techniques, we plan to do:

A)Using SIMCLR framework on our unlabelled data (which has to be used for pretraining) for doing self supervised learning to generate representations for our pretext task and then downstream it on our labelled dataset with latest SOTA models like Full-Resolution Residual Networks (FRRN B) for road image prediction (98.2%) (https://arxiv.org/pdf/1611.08323.pdf) adding it to these SIMCLR representations as the last layers of our model and RetinaNet for object detection.
Doubt questions for this architecture:

1. We are trying to understand how to bring in semi supervised learning in the above framework. We studied two methods (Pseudo labelling and Distillation) and were wondering as to use part of the unlabelled dataset for self supervision to generate labels and then use this neural network model for the rest of the unlabelled dataset to generate labels (Pseudo labelling)? Is this the right thinking process?

B) Moreover, we were planning to do a kind of ensemble technique to improve accuracy. Like train our unlabelled data of different self supervision tasks like (Moco, SIMCLR) and get various representations and use semi supervised techniques to get representations for the entire unlabelled dataset and then multiple classifiers in the downstream classification tasks. So say we build 3 self supervision representations on unlabelled dataset and 3 classification models for prediction (like FRRNB, Resnet50 etc). We could get a total combination of 8 neural network models, which we could ensemble to get our predictions. Does this look like a good idea. or do we need to research more? (**This looks like a time consuming process, as we are quite new to CV and DL as such and also using HPC to resolve quite a lot of environment issues which we might face.**) (Any Suggestions for this?)

Issues in this project:

NUMPY VERSION - 1.17

So since June to aug, I wanted to work on my research area of interest.(Unpaid/remote) would not be an issue for me. I specifically reached out to you as I wanted practical experience in conversational AI, which is my research interest.

The intrinsics matrices:

```
[[879.03824732   0.         613.17597314]
 [  0.         879.03824732 524.14407205]
 [  0.           0.           1.        ]]
[[882.61644117   0.         621.63358525]
 [  0.         882.61644117 524.38397862]
 [  0.           0.           1.        ]]
[[880.41134027   0.         618.9494972 ]
 [  0.         880.41134027 521.38918482]
 [  0.           0.           1.        ]]
[[881.28264688   0.         612.29732111]
 [  0.         881.28264688 521.77447199]
 [  0.           0.           1.        ]]
[[882.93018422   0.         616.45479905]
 [  0.         882.93018422 528.27123027]
 [   0.            0.            1.         ]]
[[881.63835671   0.          607.66308183]
 [   0.          881.63835671 525.6185326 ]
 [   0.            0.            1.         ]]
```

The order of the camera is

'CAM_FRONT_LEFT', 'CAM_FRONT', 'CAM_FRONT_RIGHT', 'CAM_BACK_LEFT', 'CAM_BACK', 'CAM_BACK_RIGHT'

https://arxiv.org/pdf/1808.00327.pdf (Luke Martin Paper)

Image preprocessing: Camera Intrinsics

Task left: (Resnet 50) - we won't be able to run DeepLabv3/ FRRNb

1. **Camera intrinsics**
2. **Monecular Depth Estimation**
3. **SIMCLR + DeepLabv3**
4. **Semi Supervised Learning**

Later:

Semi-supervised

CAMERA Intrinsics:

https://github.com/darylclimb/cvml_project/tree/master/depth/self_supervised_sfm

Depth Estimation:

https://github.com/nianticlabs/monodepth2

Semi Supervised:

https://ruder.io/semi-supervised/

(Proxy Labels, Graph Convolutions networks)

TASKS LEFT

1. **END TO END TRAINING - SIMCLR**
2. **Downstream - DeepLab**
3. **Downstream - Cascade Mask RCNN**
4. **Semi Self Supervised Learning research**
5. **Camera Intrinsics**
6. **Monocular Depth Estimation - self supervised**
7. **End to end Pipeline build - Scripting**
8. **Why is bounding box zero?**
9. **Report, PPT, video (8th -11th)**

Object Detection Task - Good links - for Lit Survey :

1.https://github.com/nianticlabs/monodepth2
2.
https://towardsdatascience.com/monocular-3d-object-detection-in-autonomous-driving-2476a3c7f57e
3.

```
_, depth = disp_to_depth(disp, 0.1, 100)
depth_to_3d = BackprojectDepth(1, original_height, original_width)
K = np.array([[[882.93018422, 0, 616.45479905, 0], [0, 882.93018422, 528.27123027, 0], [0,
0, 1, 0], [0, 0, 0, 1]]], dtype=np.float32)
K[:2,:] = K[:2,:]/4
inv_K = np.linalg.pinv(K)
inv_K = torch.from_numpy(inv_K)
pointclouds = depth_to_3d(depth, inv_K)
points_to_TV = ProjectTV(1, original_width, original_height, original_width,
original_height, 1)
top_view = points_to_TV(pointclouds)
save_topview(top_view, 'tv_test')
```

Tasks left:

1. **TopDown/simclr training**
2. **GPU scripts**

3. **Downstream Training - RoadMap**
4. **Downstream Object Detection**

Slides Format:

1. **Intro+ performances**

2. BASELINES (Resnet 18, Faster RCNN)
3. SSL- SIMCLR
4. SSL - Monocular Depth +SIMCLR
5. Prediction Image through model built wrt. GT
6.Other techniques tried(FRRN B, DeepLab, Pseudo Lidar) and issues faces
(Add pseudo lidar code)

Object detection - links -
https://github.com/charlesq34/frustum-pointnets
http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d
https://github.com/mileyan/pseudo_lidar/tree/6e60ebabd47082af4031fed4481d23d9bca4598f
https://arxiv.org/pdf/1812.07179.pdf
https://towardsdatascience.com/monocular-3d-object-detection-in-autonomous-driving-2476a3c7f57e
https://arxiv.org/pdf/2002.08394.pdf
https://arxiv.org/pdf/1812.07179.pdf
https://github.com/nianticlabs/monodepth2

Moco - video:

https://github.com/danielgordon10/vince