# Self-Supervised Feature Learning for Online Multi-Object Tracking

Bradley Collicott
Stanford University
collicott@stanford.edu

Mrunal Sarvaiya
Stanford University
mrunaljs@stanford.edu

Brian Weston
Stanford University
bweston@stanford.edu

## Abstract

*The multi-object tracking problem has a rich history in both model-based and data-driven approaches. This project serves as an investigation into improving the performance of an online multi-object tracking pipeline by introducing a self-supervised feature extraction network to provide dense feature representations of detected pedestrians in video frames. Multiple pretext tasks for self-supervised representation learning are explored and evaluated against a baseline method from literature on the downstream task of multi-object tracking. Results show that while pre-trained networks do not significantly benefit from self-supervised transfer learning, self-supervised learning can be a viable replacement for supervised feature learning. However, it is shown that a smaller network trained using metric learning for person re-identification can outperform larger fully-supervised networks.* [1]

## 1. Introduction

Multi-object tracking (MOT) from video lies at the intersection of multiple core problems in computer vision. Described in [3], the MOT process consists of 4 stages: Detection, Feature Extraction, Motion Prediction, and Data Association. As explored in the seminal work by Fortmann, Bar-Shalom, and Scheffé in 1980 [5], there is a rich history of using the model-based methods of joint probabilistic data association (JPDA) and multi-hypothesis tracking (MHT) to solve MOT. Only recently have the advances in deep learning permeated the field with impressive results. DeepSORT [23], one of the current state-of-the-art methods in real-time tracking, uses deep learning to provide a dense feature representation for use in a data association algorithm. In contrast, other works have attempted an end-to-end deep learning approach to MOT using modern architectures like recurrent neural networks and transformers [1, 13]. These methods prove effective but suffer high training time, computation, and memory costs.

[1] Project code is available on GitHub.

The data association step (i.e. associating a detection with an existing trajectory) in MOT is widely considered to be the bottleneck for current performance. Therefore, this work seeks to investigate methods for improving feature representation to reduce the likelihood of ID switching (IDSW) and incorrect associations. Recent works have shown that self-supervised learning [10] can imbue networks with the ability to learn context and salient features by solving a non-trivial pretext task at training time. The selection of pretext task depends on the downstream application, but in all cases the solution to the task is self-evident from the input, e.g. ordering a set of unordered video frames. With this in mind, the effectiveness of using transfer learning on a pre-trained feature extraction network with multiple pretext tasks is investigated for improving MOT performance. Additionally, a fully self-supervised network is trained from scratch to compare learned features against that of a network trained in a fully-supervised manner.

## 2. Related Work

### 2.1. Deep Learning in Multi-Object Tracking

The roots of deep learning in MOT can be traced back to Wang et al. [21], where deep features are learned for model-free tracking. The feature learning network resembles the modern autoencoder and includes pre-training to learn generic feature before transferring the model to multi-task learning on multiple objects. Since then, deep learning in multi-object tracking has exploded, largely due to advancements in object detection like as the Faster-RCNN [17]. Although gains in deep MOT have largely been in detection and feature description, some works attempt deep affinity scoring to better match feature vectors [20], deep motion prediction with recurrent neural networks [18], and end-to-end solutions with the a paradigm shift towards Transformers [13].

### 2.2. Self-Supervised Learning and Pretext Tasks

Pretext tasks such as predicting image rotations are commonly used to train models to perform image classification

on large datasets such as ImageNet. The authors from [7] train an AlexNet model to predict the correct image orientation in a self-supervised fashion. Similarly, [15] introduces the jigsaw puzzle for self-supervised feature learning for transfer to classification and detection tasks. Towards using self-supervision for transfer learning, Gidaris et al. [6] demonstrated that few-shot learning could be improved by conducting transfer learning with a pre-trained feature extraction network using self-supervised representation learning.

## 3. Problem Approach and Methodology

The objective of MOT is to detect the objects of interest and track them across multiple time steps. This process can be decomposed into detection, feature extraction, motion prediction, and data association steps. In the tracking-by-detection paradigm, which is adopted here, this includes extracting the features of each detection and comparing with previously extracted features to associate each detection with the correct ID – in this way, MOT is largely a data association problem. Further, online MOT addresses the problem of tracking objects in real-time by sequentially processing video frames. To limit the scope of this project to self-supervised feature learning, the DeepSORT MOT pipeline [23] is adopted as the motion prediction and data association backend. Likewise, the public detections available with MOT17 dataset [14] are used to allow for explicit comparison of the effectiveness between feature extraction networks without conflating with detector performance. All neural networks are implemented using PyTorch [16]. The proposed self-supervised learning approaches are presented here.

### 3.1. Pretext Tasks for Self-Supervised Transfer Learning

Pretext tasks are used in self-supervised learning to encourage neural networks to learn a representation that is useful for a separate downstream task. Given an image classification network pre-trained on a large dataset, the goal of this investigation is to augment the learned features using self-supervised training. For this task, a ResNet-18 [9] pre-trained for classification on the ImageNet dataset [4] was selected as the backbone network. The ResNet architecture is shown in Fig. 1.

The ResNet-18 is an 18-layer network consisting of four 'residual blocks' that reduce the spatial extent of the image while increasing the channel dimension. The pre-trained ResNet, provided natively in PyTorch, accepts $224 \times 224$ RGB images as inputs and requires that the images be normalized by a pre-defined mean and standard deviation. The network uses a global average pool before it's final fully-connected layer, resulting in a $512 \times 1 \times 1$ feature vector for use in the downstream application. This feature vector is the

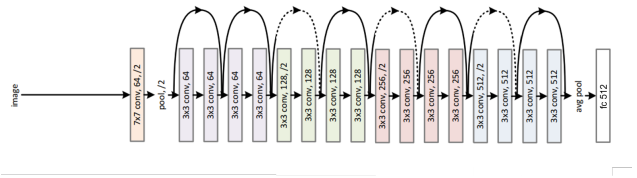



Figure 1: ResNet-18 Architecture (adapted from [9])

subject of interest for transfer learning with self-supervised training – two pretext tasks were selected to learn representations for evaluation on the downstream application of MOT.

#### 3.1.1 Puzzle Identification

To encourage the pre-trained network to learn features relevant to that of a pedestrian, the puzzle identification task is considered. The traditional jigsaw puzzle task is such that the network is provided with 2 or more image segments, and the task to label each image segment with the correct ordering that would reconstruct the original image. In this case, the network is only presented with one image that is sampled as one of four quadrants from the original image. The target label is the number corresponding to the sampled quadrant. This scheme is visualized in Fig. 2.

The pre-trained ResNet-18 is used as the backbone architecture for this task. The network is augmented by removing the final fully-connected layer and re-initializing the weights in the final residual block using Xavier initialization [8]. The global average pool is retained and fully connected layers of dimension $512 \times 4$ is added to output logits for predicting the label for the puzzle task. For evaluation in the downstream MOT task, the feature vector is taken as the output of the global average pool layer, flattened to dimension $512 \times 1$.

#### 3.1.2 Predicting Image Rotations

One of the pretext tasks considered is predicting image rotations. The goal of this network is to learn features that enable it predict the correct 2D rotation that is applied to the image as an input. We start with the pretrained network used as the feature extractor in DeepSORT. We replace the last two layers (Batch and l2 normalization and Dense 10) with two fully connected layers as done in the original RotNet model [11]. Finally, the loss function is the same as the one used in [7] which is the negative log likelihood of the data set.

The model will be trained in a self supervised way using the cropped detection bound boxes from the MOT17 dataset. Since the model will learn to predict the correct rotations of humans, we expect the learned features to be use-

ful in similar tasks, which in this case would be the affinity computation stage in the MOT tracking framework.


Figure 2: Puzzle and Rotation Task Example

### 3.2. Fully Self-Supervised Learning

In contrast to pretext tasks for self-supervised transfer learning with a pre-trained ResNet, here we implement a fully self-supervised representation learning approach that does have any reliance on pre-trained models using labels. Here, we train our own convolutional autoencoder network to learn useful representations by reconstructing images. The convolutional autoencoder network is shown in Fig. 3. An autoencoder has an encoder and decoder block, where the encoder compresses the input features and the decoder tries to recreate the input features from the compressed bottleneck provided by the encoder.

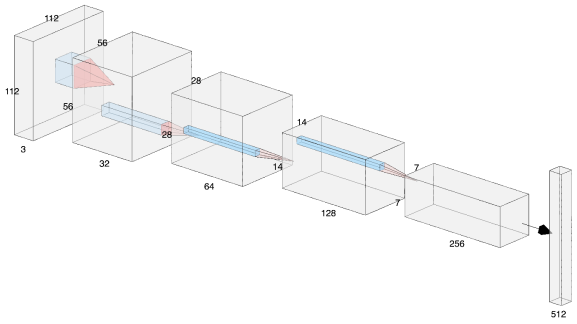



Figure 3: Convolutional autoencoder network architecture (encoder portion shown).

The convolutional autoencoder accepts $112 \times 112$ RGB images as inputs and outputs RGB images of the same size. The encoder's convolutional layers have a kernel size of $3 \times 3$, a stride of 2, and a padding of 1, which results in shrinking of the spatial size by a factor of 2 for each convolutional layer. Each convolutional layer is followed by a Leaky ReLU activation function. The bottleneck linear layer has a feature vector size of $512 \times 1 \times 1$, which is used as a feature extractor for downstream tasks. The decoder has the same architecture as the encoder, but the convolutional layers are replaced by transposed convolutions with a kernel size of 4, a stride of 2, and padding of 1. Since we only use the autoencoder for feature extraction, after training is finished the decoder is discarded and the encoder can be used for feature extraction on new (MOT17) input data.

### 3.3. Baseline

The baseline method will be a network trained using the deep cosine metric [22] from the original DeepSORT feature extractor. This model was trained using a modified softmax loss to encourage features embeddings that are similar for input images of the same person and dissimilar to those for images of different people. The network used in DeepSORT is relatively shallow – the architecture is shown in Fig. 4. The network is substantially smaller than most modern CNN classifiers, and consideration will be given to the parameter count during discussion and comparisons.

| Name | Patch Size/Stride | Output Size |
|---|---|---|
| Conv 1 | $3 \times 3/1$ | $32 \times 128 \times 64$ |
| Conv 2 | $3 \times 3/1$ | $32 \times 128 \times 64$ |
| Max Pool 3 | $3 \times 3/2$ | $32 \times 64 \times 32$ |
| Residual 4 | $3 \times 3/1$ | $32 \times 64 \times 32$ |
| Residual 5 | $3 \times 3/1$ | $32 \times 64 \times 32$ |
| Residual 6 | $3 \times 3/2$ | $64 \times 32 \times 16$ |
| Residual 7 | $3 \times 3/1$ | $64 \times 32 \times 16$ |
| Residual 8 | $3 \times 3/2$ | $128 \times 16 \times 8$ |
| Residual 9 | $3 \times 3/1$ | $128 \times 16 \times 8$ |
| Dense 10 | | 128 |
| $\ell_2$ normalization | | 128 |

Figure 4: Baseline Network Architecture [22]

## 4. Datasets and Metrics

Several datasets were used for training and the ultimate evaluation. Each dataset and accompanying evaluation metrics are briefly described here.

### 4.1. MOT Evaluation

The primary evaluation dataset will be the MOT17 [14] pedestrian tracking dataset. This dataset provides pedestrian detections from Faster-RCNN, Deformable Parts Model (DPM), and scale-dependent pooling (SDP) object detectors with ground truth annotations for over 1300 unique IDs across over 11,000 frames. A sample frame from the MOT17 training data is shown in Figure 5.

Ground truth annotations for the MOT17 test split are not provided, so networks are evaluted using the training split. For this reason, additional datasets are used for the self-supervised transfer learning and training.

(a) Raw (b) Annotated

Figure 5: MOT17 Sample Scene

#### 4.1.1 MOT Metrics

The CLEAR MOT metrics established in [2] will be used to evaluate tracking performance. For a tracker that outputs a set of hypotheses consisting of paired bounding box and ID labels, we define the following terms for computing metrics:

- False Negative: Ground truth object for which no hypothesis was output.
- False Positive: Hypothesis for which no ground truth object exists.
- ID Switch: An erroneous change to a correct ID association.

The catch-all metric for tracking is the multi-object tracking accuracy (MOTA).

$$MOTA = 1 - \frac{FN + FP + IDSW}{GT} \tag{1}$$

Where $FN$, $FP$, $IDSW$, and $GT$ are the number of false positives, false negatives, ID switches, and ground truth tracks across all frames, respectively. Another holistic measure of MOT performance is the Multi-Object Tracking Precision (MOTP).

$$MOTP = \frac{\sum_{t,i} IOU(d_{t_i}, \hat{d}_{t_i})}{\sum_t c_t} \tag{2}$$

Where $IOU(d_{t_i}, \hat{d}_{t_i})$ is the intersection-over-union of bounding box $i$ in frame $t$, and $c_t$ is the total number of ID matches in frame $t$.

In addition to the CLEAR MOT metrics, it is common to use the Mostly Tracked (MT) and Mostly Lost (ML) metrics to capture the number of trajectories correctly and incorrectly tracked for at least $80\%$ of the frames in which the tracks are present. Additionally, a the number of Fragments (FRAG) is defined as a trajectory hypothesis which covers less than $80\%$ of a ground truth trajectory. Since this project will use public detections, the ML, MT, IDSW, and FRAG metrics will be most informative for assessing performance.

### 4.2. Self-Supervised Training Data

The Motion Analysis and Re-Identification Set (MARS) [19] served as the primary dataset for self-supervised transfer learning on the rotation and puzzle pretext tasks. MARS contains bounding box images of over 1200 pedestrians from multiple camera views with the intent to train networks to recognize the same person from multiple views or across consecutive frames. A sample track from the MARS dataset is shown in 6.

Figure 6: MARS Dataset Sample Track

The convolutional autoencoder network was trained on the Microsoft Common Objects in Context (MS COCO) dataset. The 2017 unlabeled MS COCO dataset contains over 100,000 images of common objects. A random subset of images is shown in Fig. 7.

Figure 7: MS COCO Dataset Sample Images

#### 4.2.1 Self-Supervised Training Metrics

Puzzle prediction and rotation are both classification tasks which use the cross entropy (CE) loss. This can be computed for a batch of $N$ inputs with $C$ possible classes as:

$$\mathcal{L}_{CE} = \sum_{i=1}^{N} \frac{e^{s_{y_i}}}{\sum_{j=1}^{C} e^{s_j}} \tag{3}$$

where $s_i$ represents the output logit for class $i$.

The convolutional autoencoder is trained using self-supervised image reconstruction. This task uses a mean-squared erorr (MSE) loss between the input image $I$ and its reconstructed image $\hat{I}$.

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^{N} \left( I_i - \hat{I}_i \right)^2 \tag{4}$$

# 5. Results and Discussion

## 5.1. Pretext Task Training

Although high performance on the pretext tasks is not the end-goal for this study, the training procedure and results for the self-supervised learning is briefly reviewed.

### 5.1.1 Puzzle Task

The puzzle task was trained using the Adam optimizer with a learning rate of $1e{-}5$ on mini-batches of size 128 for 10 epochs. The training data consisted of 25600 images from the MARS training split and was validated using a withheld set of 3200 images from the MARS test split. The network achieved a validation loss of $\mathcal{L} = 0.824$ and validation accuracy of $92.2\%$.

### 5.1.2 Rotation Task

The rotation task was trained using the Adam optimizer and cross entropy loss for 5 epochs. An initial grid search was run to find the initial set of values for the learning rate, betas, batch size and decay weight. After a finer search, the following values yielded the best results - learning rate: 6.7e-5, betas: (0.901, 0.986), batch size: 4, weight decay: 0.044. The training data consisted of 100000 images from the MARS dataset and was validated against a separate set of 100000 images. The network achieved a validation accuracy of $98.8\%$.

### 5.1.3 Image Reconstruction Task

The convolutional autoencoder image reconstruction task was trained using an Adam optimizer with a learning rate of $1e{-}3$ on mini-batches of size 64 for 30 epochs. The training data consisted of 100,000 images from the 2017 unlabeled MS COCO dataset. The network achieved a validation loss of $\mathcal{L} = 0.013$. As seen in Fig. 8, the convolutional autoencoder does quite well in reconstructing the original images with a noticeable blur due to the small bottleneck layer of size 512.

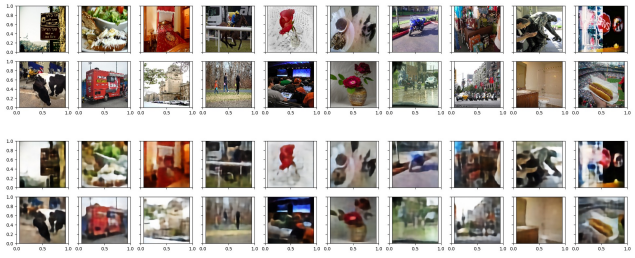


Figure 8: Top figure: sample of 20 images from MS COCO dataset. Bottom figure: output reconstructed images from the trained autoencoder.

## 5.2. MOT Performance

All networks were evaluated on the MOT17 dataset [14] using the same set of provided SDP, DPM, and FRCNN detections and the DeepSORT backend [23] for motion prediction and data association. Additionally, metrics were generated using TrackEval [12], the official evaluation tool for the MOT challenge. The 7 videos from the MOT17 training split were used for evaluation – between the three detectors, these videos amount to 15948 frames and 1638 pedestrian tracks. Results for this evaluation are shown in Table 1.

As shown, baseline network generally outperforms the much larger pre-trained and transfer learning networks, except in the Mostly Tracked, Mostly Lost, and False Positive metrics. The pre-trained ResNet does, however, provide competitive results without direct metric training or other feature augmentation for person re-identification.

Additionally, networks adapted for transfer learning using self-supervised pretext tasks failed to outperform the baseline in most metrics, except for the MT, ML, and FP metrics as mentioned previously. These networks also generally do not exceed the performance of the pre-trained ResNet, despite achieving high accuracies in their respective pretext tasks. This indicates that the tasks may not have been difficult enough or contextually significant enough for the network to learn additional useful features, resulting in the weights of the re-trained layers adding little-to-no information to the feature vector compared to the ResNet backbone.

The self-supervised autoencoder shows promising performance that rivals that of the ResNet-18, which was pre-trained in a fully-supervised manner on ImageNet. The autoencoder is also generally competitive with the baseline model with the exception of ID switching, which occurs approximately $41\%$ more often in the autoencoder than the baseline model. This suggests that self-supervised feature learning alone does not surpass metric learning for person re-identification, despite the performing better than the pre-trained ResNet.

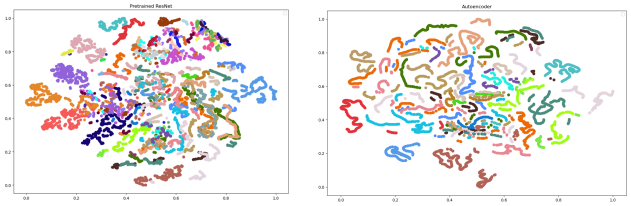
When comparing an autoencoder trained on the MS COCO dataset against one trained on the MOT17 bounding boxes used for evaluation, we see similar performance between the two. This suggests that the autoencoder is not sensitive to the choice of dataset in training and that there is not a significant loss in capability by training on a non-pedestrian dataset. The autoencoder trained on MOT17 was not considered in the performance comparison with the baseline as it was trained on the evaluation data – this network was included only to consider the sensitivity of the autoencoder to it's training process.

| | No. Params | MOTA ↑ | MOTP ↑ | MT ↑ | ML ↓ | IDSW ↓ | FRAG ↓ | FP ↓ | FN ↓ |
|---|---|---|---|---|---|---|---|---|---|
| Baseline [23] | 2.8m | *48.207* | *83.746* | 340 | 573 | *1458* | *3680* | 161291 | *11738* |
| ResNet-18 [9] | 11.2m | 47.869 | **83.561** | 362 | 546 | 2176 | 4175 | 159140 | 14308 |
| ResNet-18 + Puzzle | 11.2m | 47.488 | 83.541 | 363 | **543** | 2412 | 4231 | 159183 | 14301 |
| ResNet-18 + Rot. | 11.2m | 47.112 | 83.297 | 354 | 545 | 4100 | 4603 | 159582 | 14494 |
| Autoencoder (COCO) | 6.8m | **47.947** | 83.557 | **367** | 546 | **2064** | **4172** | **159054** | **14243** |
| Autoencoder (MOT17) | 6.8m | 47.970 | 83.559 | 370 | 544 | 2048 | 4136 | 159064 | 14174 |

Table 1: MOT17 Evaluation Results – **Bold** = Best Results in this Study; *Italics* = Baseline Obtains Best Results

### 5.3. Feature tSNE

To qualitatively analyze how effective the models are at differentiating different images of humans, we used tSNE to project the feature vectors generated on the MOT17 dataset onto a 2D plane.



(a) tSNE: ResNet18 (b) tSNE: Auto-Encoder

Figure 9: tSNE visualization

Each color in the figures above denotes a single tracked human in the video sequence. A pretrained ResNet-18 model and auto-encoder generate distinct clusters for data points with the same labels and almost always only have one cluster per label. The auto-encoder performs better since it generates clusters that are further apart, which makes it easier to distinguish different labels. This qualitative analysis is supported by the quantitative results presented.

### 5.4. Feature Vector Ablation

To investigate the sensitivity of the MOT process to feature vector length, the feature dimension was varied for the pre-trained ResNet-18 and the network trained using the puzzle task. The reduced dimension feature vectors were obtained from the ResNet using a 1D adaptive average pooling layer to reduce the dimensions from the original 512-dimension feature vector. The reduced dimension feature vector from the puzzle-task network was obtained by training multiple networks with differing linear output layers, resulting in learned feature vectors of varying lengths. The effect of this variation is shown on in Fig. 10 for the MOTA and IDSW metrics.

As shown in the figure, the MOT performance of the pre-trained ResNet-18 is resilient to changes in feature vector length, whereas the puzzle-task network shows a pre-

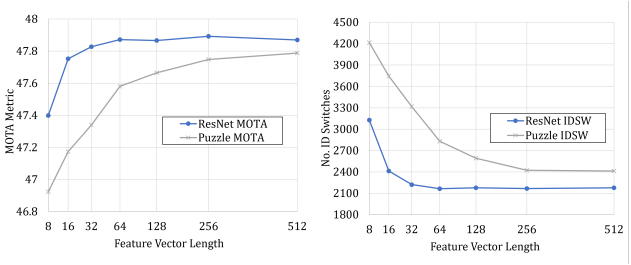



Figure 10: Effect of Varying Feature Vector Length on MOT Performance

mature performance drop-off. This suggests that the feature vector produced by the puzzle network is comprised of less salient information than the ResNet, and that the self-supervised transfer learning proved counterproductive.

With regard to comparing the performance of networks with different feature vector lengths, Fig. 10 suggests that the comparison is fair so long as the feature vector reaches a critical length in which the results no longer improve. For the ResNet-18, this is approximately 128 – which is the dimension of the baseline network's feature vector. Since the performance of does not substantially increase for feature lengths greater than 128, fair comparison can be made between networks with feature vectors of different lengths.

## 6. Conclusion

Several methods were investigated for learning a feature embedding conducive for person re-identification via self-supervised learning. Two pretext tasks were used to augment the learned feature embedding of a ResNet-18 pre-trained on ImageNet, and another network was trained from scratch using self-supervised image reconstruction. These methods were compared to a baseline metric learning method from literature on the MOT17 dataset. Although self-supervised learning is a burgeoning research area for learning useful feature embeddings from large sets of unlabelled data, it did not prove particularly effective in transfer learning with the pre-trained network. The pre-trained ResNet-18 was shown to generally outperform or match the performance of the networks trained on additional pretext

tasks. The self-supervised autoencoder performed similar to the pre-trained ResNet-18 with slight improvements in the MOTA, MT, IDSW, FRAG, and FP metrics. This suggests that fully self-supervised learning is a viable strategy for learning a useful feature embedding for person re-identification.

Several additional observations were made regarding the requisite length of a feature embedding for person re-identification as well as a reduced-dimension analysis of the feature space for several networks. The feature vector length may be reduced to lessen memory and computational requirements at the loss of MOT performance beyond a certain threshold. It was also shown that the ResNet-18 pre-trained network and self-supervised autoencoder produce feature embeddings that allow for clustering, further suggesting their suitability for person re-identification.

Although the self-supervised transfer learning attempted in this effort was unsuccessful, there is future work in self-supervision for multi-object tracking. Extensions to this work could include: incorporating a meta-learning algorithm to guide the self-supervision process; training a self-supervised autoencoder jointly with metric learning tasks to improve performance on person re-identification; and merging the detection, feature description, and/or affinity calculation segments in a push towards end-to-end MOT.

## 7. Contributions and Acknowledgements

This work makes use of several open-source repositories: DeepSORT for the MOT backbone and pre-trained baseline network; and TrackEval for evaluating the MOT metrics.

Group member contributions are as follows: B.C. led report writing and contributed the core PyTorch code framework, puzzle pretext task, model evaluations, and feature vector ablation study. B.W. was responsible for the autoencoder image reconstruction pretext tasks and led the poster writing. M.S. setup the DeepSORT framework to support using custom dataloaders and models for feature extraction, trained the model rotation pretext task and generated tSNE visualizations.

## References

[1] F. Bastani, S. He, and S. Madden. Self-supervised multi-object tracking with cross-input consistency, 2021.

[2] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. 1 2008.

[3] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88, 3 2020.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[5] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. In *1980 19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, pages 807–812, 1980.

[6] S. Gidaris, A. Bursuc, N. Komodakis, P. Perez, and M. Cord. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[7] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Y. W. Teh and M. Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[10] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:4037–4058, 11 2021.

[11] J. E. Johnson, S. Sundaresan, T. Daylan, L. Gavilan, D. K. Giles, S. I. Silva, A. Jungbluth, B. Morris, and A. Muñoz-Jaramillo. Rotnet: Fast and scalable estimation of stellar rotation periods using convolutional neural networks, 2020.

[12] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, pages 1–31, 2020.

[13] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer. Trackformer: Multi-object tracking with transformers. 1 2021.

[14] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler. Mot16: A benchmark for multi-object tracking, 2016.

[15] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 69–84, Cham, 2016. Springer International Publishing.

[16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[17] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[18] A. Sadeghian, A. Alahi, and S. Savarese. Tracking the untrackable: Learning to track multiple cues with long-term

dependencies. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 300–311, 2017.

[19] Springer. *MARS: A Video Benchmark for Large-Scale Person Re-identification*, 2016.

[20] S. Sun, N. Akhtar, H. Song, A. Mian, and M. Shah. Deep affinity network for multiple object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):104–119, 2021.

[21] L. Wang, N. T. Pham, T.-T. Ng, G. Wang, K. L. Chan, and K. Leman. Learning deep features for multiple object tracking by using a multi-task learning strategy. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 838–842, 2014.

[22] N. Wojke and A. Bewley. Deep cosine metric learning for person re-identification. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018.

[23] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. *Proceedings - International Conference on Image Processing, ICIP*, 2017-September:3645–3649, 2 2018.