CMPT828 MOTSynth Final Report

Grant Tingstad
NSID: gdt741
University of Saskatchewan
gdt741@usask.ca

Abstract

In this project, a method for training human tracking models on synthetic data is demonstrated. The presented solution relies on automatically generated annotations from synthetic data sources to train a supervised baseline model. Using the baseline model to annotate unlabelled real data, a high confidence pseudo-labelled dataset was generated. This dataset was later augmented with synthetic data to create a semi-supervised domain adapted model which required no manual annotation. Finally, model weight averaging was used to blend the baseline and augmented models and reduce over-fitting. Each model was paired with an object tracking scheme and their tracking accuracies were evaluated. While the overall results scored below state-of-the-art models, tracking performance did improve with each step in the process, presenting a case for further investigation into synthetic data use. The code for this implementation is available at https://git.cs.usask.ca/gdt741/cmpt828-courseproject.git.

1. Introduction

The widespread adoption of Deep Learning methods within the field of machine learning has dramatically improved data-driven applications [19]. Many tasks, like real-time semantic segmentation and language translation were subjects of fiction before deep learning technology became available on consumer-grade devices. Since the adoption of deep learning, such capabilities have been available to anyone with a tablet, smartphone, or computer. Deep learning has become so integrated into our society that it affects us daily in ways many of us don't even realize; corporations use it to drive marketing campaigns, banks use it to drive investment decisions, automakers even use it to drive cars.

While Deep Learning technology has unlocked new heights machine learning potential, the efficacy of these methods are often constrained by the availability of good data. Deep learning, of course, uses many layers of calculations to analyze patterns in data and reinforce strategies that support strong decision-making over many training intervals. For this method to be successful, however, learning models must be given many data samples to learn from. Not only must these learning samples be well annotated, an expensive and time-consuming task [13], but the data must also be representative of the population of possible samples the model may see after training. Moreover, there are growing privacy concerns about how deep learning data is collected and used, particularly for applications that require images of public spaces. These concerns present a monumental challenge for many machine learning scientists [22] and we expect these challenges to grow with the demand for more sophisticated machine learning technologies [17].

To solve the problems associated with manually curated deep learning data sets, some researchers have investigated the use of synthetic data as a substitute. Synthetic data offers many potential benefits over real-world data. For one, synthetic data can be generated in large volumes with desired properties. Synthetic data generation also allows for precise control over many variables that may be difficult to replicate in the natural world. These traits work together to eliminate common data issues such as bias, privacy concerns, and real-world limitations. Ultimately, synthetic data could play a crucial role in advancing deep learning science [16].

1.1. Key Challenges with Synthetic Data

Data privacy, bias, and quality problems are a particularly tricky subject for pedestrian detection. Many automakers have ambitious goals of releasing self-driving technology in their vehicles, but to do this safely, robust human segmentation algorithms must be able to successfully detect people within view. This is particularly challenging because of the variability that a self-driving car might encounter around the world; street layouts and backgrounds vary significantly; weather and lighting conditions vary with season and time; even variability in ethnic and cultural backgrounds of pedestrians could significantly alter key features for people detection. To develop a robust and unbiased pedestrian classification algorithm, training data would

have to contain a comprehensive combination of environmental and human variables.

Considering the challenges associated with collecting suitable pedestrian data sets, the case for investigating the use of synthetic data as a substitute for real pedestrian data is strong. The use of such technology would allow for near-realistic training data to be generated automatically. Further, with thoughtful controls to avoid bias, a large data set of images could be produced for a variety of different environments, weather and lighting conditions, and pedestrian demographics. Such software could also be used to generate annotations automatically, saving costs and reducing human error. With the right software and computational resources, the use of synthetic data could solve most the major challenges we face with curating real-world data sets. This is exactly the aim of MOTSynth. [6, 11], a collection of 767 1800-frame full-HD (1920x1080) synthetic pedestrian videos with automatically generated annotations, which was produced within the popular video game Grand Theft Auto V. The aim of the MOTSynth challenge is to evaluate the efficacy of using synthetically generated data to train real-world deep learning detection models. To date, several methods of using MOTSynth for real-world pedestrian tracking have been documented, including evaluations of domain adaption tricks that bridge domain gaps between synthetic models and live footage.

1.1.1 Domain Transferability

Pedestrian tracking is an obvious use case for synthetic data generation, but there are many other domains where similar strategies could advance computer vision technology. Within the domain of self-driving technology, synthetic data could be used for detection of traffic sign detection, street and sidewalk detection, and traffic sign detection. In other applications, the capability for automatic annotation could be game changing; data sets for human pose estimation are sparse due to annotation limitations, but the capability of software to automatically annotate joint locations and limb orientations in synthetic data sets could mitigate those barriers. As deep learning is utilized for broader automation activities, there will be requirements for larger robust data sets, and synthetic data could make many models constrained by available real data a reality much sooner than we previously thought possible.

1.1.2 Limitations of Synthetic Data

While many of the problems associated with real data sets can be solved or mitigated with the use of synthetic data, there are still limitations to synthetic data. For one, since synthetic data generation is based in virtual environments, the training data produced will never be truly photo realistic. Moreover, while synthetic data generators have the advantage of reducing bias with appropriate controls [12], these capabilities are limited by the environment designer's implicit biases and forethought. With the implementation of synthetic data for object detection models, the existing data and target environments should be studied carefully to effectively design unbiased controls.

The MOTSynth data set is interesting because the virtual environment used to capture synthetic data was not designed by the research group who curated the data set. Since all data for MOTSynth was captured within the popular video game Grand Theft Auto V, the environmental and human variables were limited by the environment and models available in the game. The use of an existing virtual environment to curate a data set is a double-edged sword; on one hand, the simulation offers a massive and diverse virtual environment and variety of human models for the simulation of real-life scenarios; on the other hand, whatever biases are present in the environment also exist in the consequent data (for example, it never snows in the game). Video game environments could prove to be powerful simulators for synthetic data generation, but particular attention should be given to implicit biases that may leave critical feature gaps in the intended end-use of models trained on such data.

1.2. Challenge Introduction

The competition that provided the main motivation for this project was the MOTSynth-MOT-CVPR22 Multiple Object Tracking challenge [2] which aims to investigate the efficacy of synthetic data for real world object detection. The challenge ran for the CVPR2022 conference and has since concluded; however, the data was still publicly available at the time of writing. The challenge provided all of the synthetic training data in a dataset called MOTSynth, which consists of over 1.3 million artificially generated video frames of pedestrians from the game Grand Theft Auto V, complete with automatically generated bounding boxes, instance segmentation masks, key points, and depth masks [6]. Real pedestrian datasets, including MOT17 containing over 17,000 frames of real pedestrian footage [4] were also provided.

Participants were tasked with training a pedestrian detection model on labelled synthetic data and validating the model's performance on annotated images of real pedestrians. At the end of the competition, users submitted their models to the organizers and testing was performed on a private test set. Submissions were scored on a variety of criteria; however, overall score was determined with Higher Order Tracking Accuracy [HOTA], which evaluates detection accuracy and association accuracy, including ID switches and fragmentation [15].

1.3. Related Work

The CVPR 2022 MOTSynth challenge concluded with 8 final submissions using a variety of implementations [5].

The first-place submission, SIA_Track, trained their model with a combination of labelled synthetic data and unlabelled real data [18]. Using a combination of the state-of-the-art object detection model YOLOX [14], and the novel object tracking model ByteTrack [23], the group trained a preliminary model on synthetic data alone. Augmentation was used to mix synthetic-only data and pseudo-labelled real data [9] using the Model Soup method [21] to prevent over-fitting on the synthetic domain. Weights and pseudo-labels were updated iteratively until the final model achieved a favorable HOTA score of 63.2. Arguably, a major reason for the high performance of this method is due to the clever use of unlabelled real data in the training algorithm. The model achieved a HOTA score of 63.2 and processed images at 24.2Hz when evaluated by the workshop organizers.

The third-place submission, PieTrack, reported using a similar approach to SIA_Track which involved training a pedestrian detection model on a combination of labelled synthetic data and unlabeled real data [20]. Also using ByteTrack and YOLOX, a baseline detector was first trained on the synthetic data. During validation, it was discovered that the false positive rate for pedestrian detection was very high for the synthetic-only trained model. To bridge the performance gap between the synthetic and real domain, the researchers used an iterative domain adaption approach which assigned pseudo-labels to real data and fine-tuned the baseline model with increasing detection confidence. Rather than using a model soup method, the group ensembled three models trained on different input resolutions. The HOTA score when tested on the validation set sans ensemble was 56.82. With the ensembling strategy, a HOTA score of 57.68 was achieved. When tested by the organizers, the speed of PieTrack was much slower than SIA_Track at only 3.6Hz.

Interestingly, SIA_Track's preliminary synthetic-only model achieved a higher HOTA score than PieTrack's final mixed-domain model. Since pre-training was not allowed as part of the challenge, SIA_Track was initially trained with 300 epochs, while PieTrack's preliminary model was only trained for 80 epochs. There were other submissions for the challenge; however, the remaining participants either did not submit a paper discussing their implementation, or submitted an implementation that did not aim to achieve goals set out for the competition (e.g., one group proposed an object tracking algorithm trained only on real data).

2. Methods

2.1. Baseline Data Preparation

The baseline model for this experiment was trained entirely on synthetic data. 200 video sequences from the MOTSynth dataset were downloaded, and individual frames were extracted. To reduce training time, I kept the first 500 frames from each sequence, resulting in a final synthetic dataset containing 100,000 images (about 7.25% of the complete MOTSynth set). For the validation set, I used half of the MOT17 dataset, which contains 11235 image frames from real pedestrian footage. The other half of the MOT17 data was reserved for final testing. Fig.1 and Fig.2 show examples from each dataset.

Figure 1. Annotated Examples from MOTSynth Dataset (Source: https://motchallenge.net/workshops/bmtt2022/)

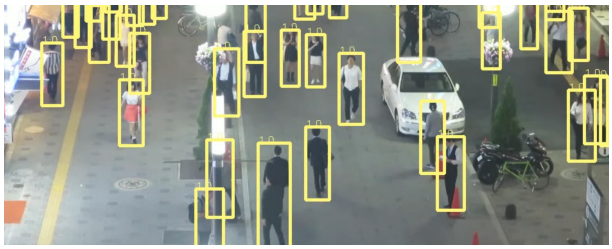

Figure 2. Annotated Examples from MOT17 (Source: //motchallenge.net/vis/MOT17-03-SDP/det/)

2.2. Training the Baseline

I decided to use YOLOv8 to train my object detection model due to it's improved performance over previous models, built-in tracker capabilities, and recent deployment of a Python module, ultralytics [7]. Ultralytics' directory traversal system also makes it easy to batch modify training data. When training in ultralytics, any number of subdirectories may be used between a root folder (e.g., './train') and the image/label pair; the only requirement is that the image and label file must share the same relative path (e.g., './train/images/some-path/0000.jpg' and './train/labels/some-path/0000.txt' would be recognized as a pair). This is a useful property, since it allows some or-

ganization between synthetic and real data during later augmentation.

Labels from all the datasets had to be transformed for compatibility with YOLOv8, as MOTSynth and ultralytics use conflicting bounding box coordinate rules. Within the synthetic dataset, it also seemed that all labels within some programmed distance had been generated in the annotation files, resulting in many instances where annotations for fully occluded pedestrians were included in the native training set, as shown in Fig.3. Therefore, only partially occluded annotations (more than 10% within view) were kept for model training. The baseline was trained on the 100,000 image synthetic dataset for 64 epochs. The model was initialized with yolov8n architecture [8] and randomly initialized weights; pretrained weights were not used to prevent prior exposure to real data. Stochastic Gradient Descent was the chosen optimizer throughout the experiment.

Figure 3. Example of Many Occluded Pedestrian Labels Appearing in Native MOTSynth Training Data

2.3. Domain Adaption

For domain adaption, another real dataset, MOT16, was used [3]. Like MOT17, MOT16 contains 14 sequences of pedestrian tracking footage. Firstly, individual frames were extracted from each dataset and stored separately from the MOT16 video sequences. Inference was then run on each sequence of images using the synthetic baseline model to generate predictions. Finally, a new annotation file was created for each frame, where only bounding boxes with confidence of higher than 0.7 were recorded. If the resulting annotation file contained any entries, the frame was considered a training candidate and the annotation/label pair were copied to a new training folder.

Once the high-confidence pseudo-labelled dataset was generated, a 'mixed' dataset was created, which included one random synthetic image/label pair for each image/label pair in the pseudo-labelled set. The resulting dataset contained a 50/50 split between high-confidence real pseudo-labelled data and synthetic data. A second model was then created by re-training the baseline on the mixed dataset. Mosaic augmentation, first introduced in YOLOv4 [10] was used to ensure a blend of real and synthetic data appeared in several training images. After training of both the baseline and mixed models was complete, a final model was created by averaging the weights of both prior models. The goal of this step was to reduce over-fitting to biased data generated in the pseudo-labelling stage

2.4. Evaluation

Loss, precision, recall, and mean average precision metrics were plotted during model training and used as a quick reference for model performance over training time. However, the main metric used to evaluate model success was HOTA. HOTA considers localization accuracy of detections and temporal accuracy of object tracking over time. The metric also considers fragmentation performance when objects are occluded or disappear and reappear at a later time, and accounts for ID switches when intersections between objects occur. Since its release in 2020, HOTA has widely been considered a key metric for multiple object tracking. I used Jonathon Lutien's TrackEval resource kit for my HOTA evaluations [1] using benchmarks from the MOT17 test set.

2.5. Further Experiments

In addition to the main experiment performed during this project, two other experiments were performed to evaluate the effects of ablation and hyperparameter tuning. Loss curves and HOTA scores were evaluated for each result.

2.5.1 Hyper-parameter Experiment

In an attempt to try and reduce overfitting to biased data during mixed model training, I trained an additional three models on the mixed dataset using dropout. Mixed models were created using dropout values of 10%, 25%, and 50%. The main motivation for this experiment was to introduce an extra layer of randomness when evaluating the weights during model training. Once these models were complete, their weights were averaged with the baseline and resulting HOTA scores were evaluated.

2.5.2 Ablation Study

Since adjustments to model architecture is inadvisable for pretrained networks, I trained a new baseline to serve as a reference for my ablation study. The baseline was trained on the full synthetic training set for 20 epochs using the standard YOLOv8n architecture. All other hyperparameters were left the same as with the main experiment. For my ablation model, I created a custom version of the YOLOv8n architecture which reduced the number of channels in each layer of the YOLOv8n backbone by one half. The result reduced the number of trainable convolution parameters from

4928 to 2464. Since a model's capacity to learn complex representations is related to the the number of trainable parameters, I expected this experiment to result in poorer performance for the affected model.

3. Results

Across all of the experiments, the HOTA scores were low. The baseline, mixed, and averaged models achieved HOTA scores of 2.85, 3.15, and 4.15 respectively, compared to scores of 60+ from state-of-the-art models. In general, I was happy to see that my main experiment improved performance over the baseline, but was a little surprised with the low results overall.

The results of my dropout experiment were not very fruitful either. Compared to a HOTA score of 4.1474 in the main averaged model, each of the dropout-averaged models achieved HOTA scores of 4.1525; slightly better performance over the main averaged model, but not enough to consider the results significant. Oddly, the metrics calculated for the dropout models during the HOTA evaluation were exactly the same for each model.

My ablation model achieved a final HOTA score of 4.715. The baseline ablation model, which was only trained for 20 epochs on the full synthetic training set, achieved a HOTA score of 4.6293. these experiments marked the highest scores of all models produced during this project. Given that the ablation baseline was essentially a lightweight version of the full baseline model used during the main experiment, this finding was intriguing. To investigate the cause of this, loss curves were generated for the models used in the main experiment, shown in Fig.4 and Fig.5. Fig.6 shows a summary of the HOTA scores for each model evaluated during this project.

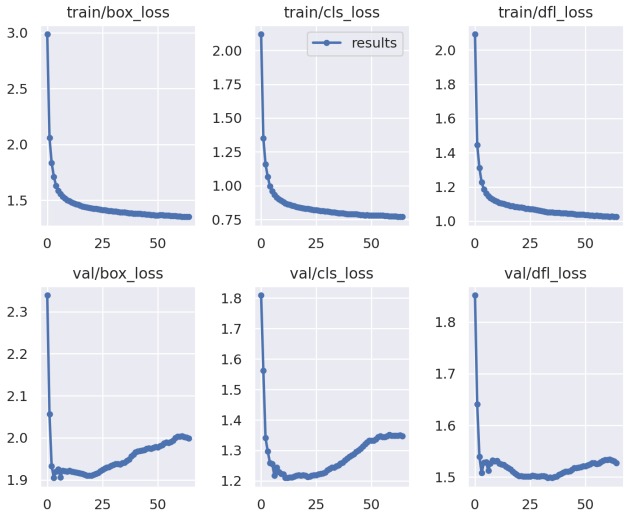

Figure 4. Generalization Performance on Baseline Model

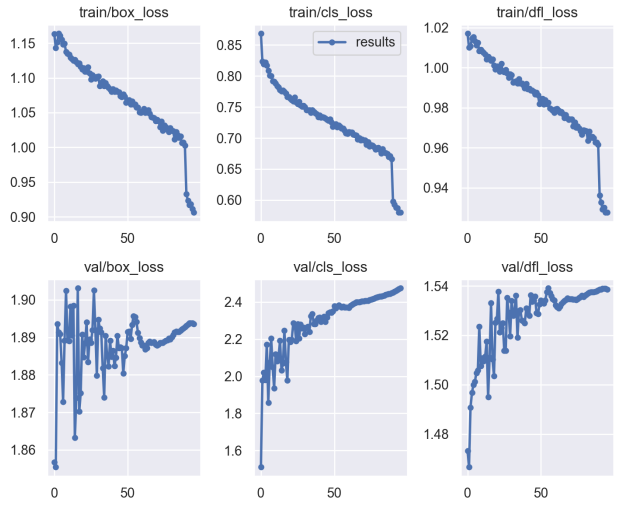

Figure 5. Generalization Performance on Mixed Model

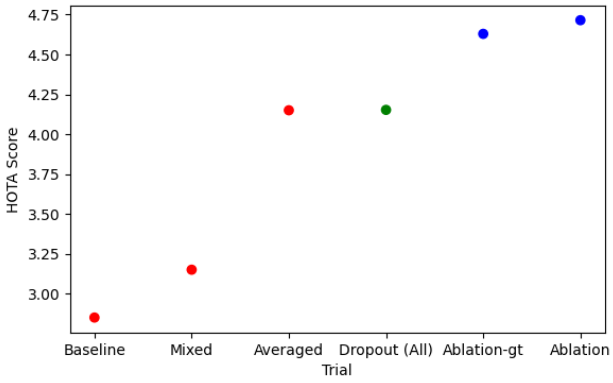

Figure 6. HOTA Scores Across Experiments

4. Discussion

Closer inspection of the generalization characteristics during model training, shown in Fig. 4 and Fig. 5, provide insights on why performance was low. It is apparent across the three loss functions that the baseline model overfit to synthetic data very early on during training. In hindsight, were I to repeat this experiment, I would have spread my training data across all 767 video sequences within MOT-Synth to enhance the variability of the data. Since tracking was not used during model training, I would have also sampled data more randomly from each sequence rather than choosing a sequential set of frames. I expect that these alterations would enhance the overall variability of the scenes in terms of both environment and pedestrian samples, and may improve baseline performance significantly.

Interestingly, the training characteristics in Fig.5 indicate that validation loss increased over time for the mixed model.

I assume that the main cause of this is due to the model reinforcing bias - where low-confidence detections were more likely to be ignored and high-confidence detections were more likely to be identified. One possible reason for the improvement in HOTA score despite a loss of performance on the real validation set is that HOTA may reward successful tracks more than it penalizes missed tracks.

I feel that one of the major issues that may have caused degrading validation loss in this step is the main augmentation scheme used by ultralytics. Mosaic randomly mixes up to four training images together and crops a section of the result out; however, since it assumes all of the training images are well-labeled, false negatives often appeared in the result during mixed model training, as can be seen in Fig.7. A solution to this problem could involve modifying the training data directly, or using a custom augmentation scheme that crops out the background of pseudo-labelled images.

Figure 7. Mosaic Including False Negatives in Mixed Training Data

One other issue I noticed with the synthetic data is that there appeared to be a delay in generating the bounding boxes. For unclear reasons some bounding boxes appeared to be lagging behind the associated object. Fig.8 shows an example of synthetic bounding box lag, which may have reduced the performance of the baseline model. Participants in the MOTSynth challenge were able to achieve high HOTA scores despite this issue, but the dataset could be improved by working to resolve this error.

Figure 8. Bounding Box Lag for Some Pedestrians

4.1. Analysis of Secondary Experiments

Dropout appeared to have very little effect on model performance. More intriguing was the lack of variability between the three dropout models used during the experiment (10%, 25%, and 50%); training runs with these three parameters produced absolutely identical results. I assume that the reason for this result is tied closely to the overfitting to synthetic data that occurred early on during model training. While dropout can help reduce overfitting in large models, it does appear that limitations in the dataset I used produced such aggressive overfitting that dropout had no discernible affect on model performance. Further investigation into dropout is warranted, but the main limitations in the dataset must be addressed first.

Unexpectedly, my ablation study showed a slight improvement over the ablation baseline, however, the fact that the ablation baseline scored so much higher than models produced in the main experiment provides some explanation for this; my data lacked variation. Since the model overfit to synthetic data so quickly, we actually see peak performance on the baseline ablation model, which was only trained for 20 epochs, compared to the main baseline model that trained for 64 epochs. I believe that the performance increase observed with the ablation model is due to the reduction in trainable parameters, which likely slowed the model's ability to overfit to synthetic data.

5. Conclusions

This project provided a glimpse into the feasibility of using purely synthetic labelled data for human tracking models. Furthermore, a method for synthetic-to-real domain adaption using pseudo-labelling and model averaging was investigated, which improved tracking performance over the baseline model. It was found that dropout had no significant affect on reducing overfitting, and that reducing the trainable parameters for the dataset used in this project improved model performance; however, these results are

consistent with my suspicion that the synthetic dataset was lacking in variability and caused overfitting to synthetic data early on during training. Overall, several methods were investigated for improving model capacity when pairing annotated synthetic data with unlabeled real data, and the discoveries made during this project provided an excellent framework for future work.

References

[1] Github: Trackeval. https://github.com/JonathonLuiten/TrackEval. Accessed: 04.15.2023. 4

[2] How far can synthetic data take us? 7th workshop on benchmarking multi-target tracking. https://motchallenge.net/workshops/bmtt2022/. Accessed: 02.04.2023. 2

[3] Mot16. https://motchallenge.net/data/MOT16/. Accessed: 04.15.2023. 4

[4] Mot17 test set. https://motchallenge.net/data/MOT17/. Accessed: 02.04.2023. 2

[5] Motsynth-mot-cvpr22 results. https://motchallenge.net/results/MOTSynth-MOT-CVPR22/. Accessed: 02.04.2023. 3

[6] Motsynth-mot-cvpr22 training set. https://motchallenge.net/data/MOTSynth-MOT-CVPR22/. Accessed: 02.04.2023. 2

[7] Yolov8 docs. https://docs.ultralytics.com/. Accessed: 04.15.2023. 3

[8] yolov8.yaml. https://github.com/ultralytics/ultralytics/blob/main/ultralytics/models/v8/yolov8.yaml. Accessed: 04.15.2023. 4

[9] Bárbara C. Benato, Alexandru C. Telea, and Alexandre X. Falcão. Iterative pseudo-labeling with deep feature annotation and confidence-based sampling. In *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 192–198, 2021. 3

[10] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. abs/2004.10934, 2020. 4

[11] Fabbri et al. Motsynth: How can synthetic data help pedestrian detection and tracking? In *ICCV*, 2021. 2

[12] Tremblay et al. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *CVPR*, 2018. 2

[13] Varol et al. Learning from synthetic humans. In *CVPR*, 2017. 1

[14] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. https://arxiv.org/abs/2107.08430, 2021. 3

[15] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taix, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129(2):548–578, 2020. 2

[16] Sergey I. Nikolenko. *Synthetic Data for Deep Learning*. Springer, 2017. 1

[17] Cloud Sammut and Geoffry I. Webb. *Encyclopedia of Machine Learning and Data Mining*. Springer, 2017. 1

[18] Minseok Seo, Jeongwon Ryu, and Kwangjin Yoon. Bag of tricks for domain adaptive multi-object tracking. https://arxiv.org/abs/2205.15609, 2022. 3

[19] Haohan Wang and Bhiksha Raj. On the origin of deep learning. https://arxiv.org/abs/1702.07800, 2017. 1

[20] Yirui Wang, Shenghua He, Youbao Tang, Jingyu Chen, Honghao Zhou, Sanliang Hong, Junjie Liang, Yanxin Huang, Ning Zhang, Ruei-Sung Lin, and Mei Han. Pietrack: An mot solution based on synthetic data training and self-supervised domain adaptation. https://arxiv.org/abs/2207.11325, 2022. 3

[21] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. https://arxiv.org/abs/2203.05482, 2022. 3

[22] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. https://arxiv.org/abs/1506.03365, 2015. 1

[23] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. https://arxiv.org/abs/2110.06864, 2021. 3