CMPT828 CVPR Project Proposal

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

1. Introduction

The widespread adoption of Deep Learning methods within the field of machine learning has dramatically improved data-driven applications [16]. 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 iterations. For this method to work, however, the learning model must be given many annotated data samples to learn from. Not only must these learning samples be well annotated, an expensive and time-consuming task, but the learning data set must also be representative of the population of possible samples the model may see after training. This presents a monumental challenge for many machine learning scientists, as deep learning data sets must be carefully curated to avoid bias [19].

The curation of data sets for deep learning models has a few key challenges. Firstly, data must often be manually annotated. Annotation is time consuming, and for some applications that require many annotations per sample (like human pose estimation), prohibitively expensive [10]. Moreover, for models which operate in the natural world, training data must not be biased towards a particular sample or environment. Lastly, there are growing privacy concerns about how deep learning data is collected and used, particularly for applications that require images of public spaces. These few examples illustrate just some of the current challenges machine learning scientists face, and we expect these challenges to grow with the demand for more sophisticated machine learning technologies [14].

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 [13].

2. Project Proposal

For this project, I have chosen to attempt the MOTSynth-MOT-CVPR22 Multiple Object Tracking challenge [2] to investigate the efficacy of synthetic data for real world object detection.

2.1. Challenge Introduction

The MOTSynth-MOT-CVPR22 challenge ran for the CVPR2022 conference and has since concluded; however, the data and helper resources are still available. The challenge provided training data, called MOTSynth, consisting 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 [5]. A validation set, MOT17, containing over 17,000 frames of real pedestrian footage was also provided [3].

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 [12].

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

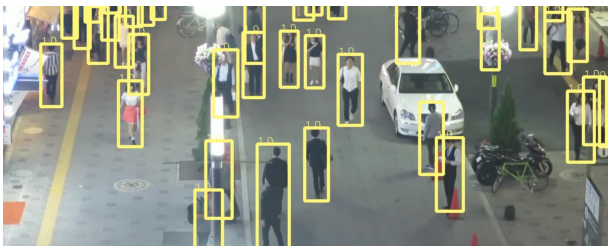

Figure 2. Annotated Examples from MOT17
(Source: //motchallenge.net/vis/MOT17-03-SDP/det/)

2.2. Synthetic Data for Real World Problems

2.2.1 Key Challenges

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 of 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. [8]

2.2.2 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.

2.2.3 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. The quality of such data could surely be good enough for a model to discriminate between humans and animals, for example, but due to limitations in modelling and photo rendering, very fine features such as those required in certain kinds of medical imaging could be very difficult to produce with synthetic data. Moreover, while synthetic data generators have the advantage of reducing bias with appropriate controls [9], 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.

2.3. Related Work and Challenge Submissions

The CVPR 2022 MOTSynth challenge concluded with 8 final submissions using a variety of implementations [4].

The first-place submission, SIA_Track, trained their model with a combination of labelled synthetic data and unlabelled real data [15]. Using a combination of the state-of-the-art object detection model YOLOX [11], and the novel object tracking model ByteTrack [20], the group trained a preliminary model on synthetic data alone. Augmentation was used to mix synthetic-only data and pseudo-labelled cite9643094 real data using the Model Soup method [18] 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 [17]. 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).

2.4. Replication of Existing Methods

The two main submissions discussed above trained their models on the entire MOTSynth dataset using a YOLOX detector and ByteTracker. The SIA_Track baseline model was trained using 8 Nvidia A100 GPUs for 300 epochs. The developer's of SIA_Track have stated that at least 8 GPUs with no less than 40GB of cobined memory are required to reproduce their performance [1], therefore, I expect similar compute capabilities would be required to train a comparable model.

There are a few other challenges I expect to encounter in trying to replicate one of the discussed methods. Due to the massive size of the data set and limitations of available compute resources, I will have to use a subset of the available training data. Moreover, many changes in package dependencies for the implementations used in the 2022 challenge make it difficult to follow their implementations. Because of the fast-moving nature of computer vision models, my own attempt at the MOTSynth pedestrian tracking problem will have likely utilize a newer detector version and slightly different domain adaption methods.

2.4.1 My Goals

I attempted to train a preliminary synthetic-only pedestrian detector on a subset of the MOTSynth dataset using the SIA_Track method. I downloaded a third of the available synthetic data set, consisting of 255 videos each containing 1800 annotated frames. To narrow the size of the data set further for my initial attempt, I selected 30 videos for my training subset.

The main challenge I encountered while attempting to replicate SIA_Track is that their code no longer runs as expected. Many of the packages rely on outdated dependencies. Furthermore, since the helper files provided by SIA_Track interact with each other to read settings and variables, I found it difficult to troubleshoot dependency errors and could not get the code to work as expected. With that said, I feel it would be possible to reproduce a similar model to SIA_Track with the following process:

1. Train a preliminary model with synthetic data on a YOLOv8 detector [6].
2. Use model to generate pseudo-labels on real data. Only keep labels with confidence ¿ 0.7.
3. Apply cross domain mixed sampling (e.g., mosaic augmentation) on synthetic and pseudo-labelled real data and retrain the model.

4. Use the Model soups method to average the weights of the unmixed and mixed model.
5. Repeat steps 2-4 iteratively until model performance gains are marginal.

For this project, I will attempt to build a custom model by following the process outlined above with modern frameworks. I will split the MOT17 data into three categories evenly; training, validation, and test. The training data will only be used for cross-domain training. The validation set will be used for evaluating the model iteratively. The test set will be used for the final evaluation of my model. As per the challenge guidelines, I will use HOTA to evaluate performance. Tentatively, I will cease cross-domain training once HOTA score increases drop below 0.25%; however, I may have to adjust my expectations once I begin generating results. I intend to perform my model training and testing on Google Colab

Figuring out how to get started with YOLOv8 and apply iterative training with mixed domain augmentation has been a challenge. The learning curve seems quite steep, but I have read through much of the available research and documentation for the methods outlined above and feel confident that I will be able to begin training a model soon.

3. Review of Relevant Literature

Since the use of synthetic data for real object segmentation tasks is a developing field, there were not many recent publications focusing on human tracking with synthetic data. Rather than review publications I discussed in my last submission, I opted to instead study some of the methods used in MOTSynth challenge submissions.

3.1. Multi-Object Tracking by Associating Every Detection Box

In their 2022 paper, Zhang et al present ByteTrack [20], a method of multi-object tracking that achieves performance and speed than other state of the art trackers. The authors point out a common flaw in other tracking methods; low-confidence detection boxes are often discarded, causing tracking issues for temporarily occluded objects. By associating low-confidence detection boxes with their predicted tracklets, ByteTrack is able to continue tracking and easily recover occluded objects and achieves state-of-the-art performance on object detection problems.

ByteTracker is an "online" tracker, meaning it works in tandem with an object detector to process live video feeds. When provided with a detection score, the algorithm performs a tracking operation for sequential frames in the video. The algorithm works by storing past object locations in a "track" and using past knowledge to predict the new location of each object. High confidence detections are associated first by matching them with the closest predicted location of an object for their class. Unassociated high-confidence detections are stored and later become the first location of a brand-new track (i.e., a new instance). The remaining tracks are then associated with low-confidence detection boxes. Unassociated low-confidence detections are assumed to be errors and dropped from the tracker. For each frame, ByteTrack outputs a set of associated bounding boxes and tracks.

The authors did not provide insight to challenges they faced during ByteTrack's development, nor did they provide suggestions for future work; however, I feel that experimenting with lightweight multi-class detection models to test how object tracking effects performance and inference speed would be a useful area to focus further research. Developments of lightweight models could advance deployment for robotics and mobile applications.

3.2. Iterative Pseudo Labelling

Semi-supervised learning is a growing field, as demand for improved detection models become more data-hungry. Since annotations can be expensive to produce, some groups have began investigating unsupervised methods for improving detection models. One research group, Benato et al, investigated the efficacy of semi-supervised learning with unlabelled data to improve detection models trained on very small, annotated data sets [7]. By assigning pseudo-labels to unsupervised samples and iteratively training a deep network with images containing high-confidence detections. Results of the work suggested that models generally improve using this semi-supervised method, but that improvements tend to peak after 2-3 training iterations. The work sets a precedent for using iterative self-supervised methods of improving detection models when annotated data is sparse.

3.3. Model Soup

In their 2022 paper, Wortsman et al. propose a method of multi-model combination which improves model performance without incurring additional costs during inference [18]. While many commonly used methods of model combination, such as ensembling, expand the size and consequent compute requirements of applying the model, model soup improves the robustness and accuracy of models by producing a weighted average of the weights of several models. In one experiment, the researchers examined twelve object detection models of varying accuracy. By taking a weighted average of the weights of the five highest-performing models to produce a new model, the group saw classification accuracies improve by approximately 1%. While the reported improvements on state-of-the-art models may not seem substantial, the strategy could prove to be very beneficial for other use cases.

References

[1] Github: Sia_track. https://github.com/SIAnalytics/BMTT2022_SIA_track. Accessed: 02.04.2023. 3

[2] How far can synthetic data take us? 7th workshop on benchmarking multi-target tracking. https://motchallenge.net/workshops/bmtt2022/. Accessed: 02.04.2023. 1

[3] Mot17 test set. https://motchallenge.net/data/MOT17/. Accessed: 02.04.2023. 1

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

[5] Motsynth-mot-cvpr22 training set. https://motchallenge.net/data/MOTSynth-MOT-CVPR22/. Accessed: 02.04.2023. 1

[6] Yolov8. https://docs.ultralytics.com/#ultralytics-yolov8. Accessed: 02.04.2023. 3

[7] 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. 4

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

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

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

[11] 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

[12] 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

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

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

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

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

[17] 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

[18] 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, 4

[19] 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

[20] 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, 4