Final Project
Object Detection & Tracking with DiffusionDet

Gabriel Fiastre, Shane Hoeberichts
gabriel.fiastre@dauphine.eu, shane.hoeberichts@dauphine.eu

Abstract

We propose to revisit a new framework for object-detection based on diffusion models, DiffusionDet, and extend it to the multi-object tracking task for pedestrians on MOT17. We prove that this approach can be efficient for the detection task and that we can take advantage of the diffusion model architecture for the tracking task. Two different prior modification methods are considered to incorporate temporal information for object detection in video sequences. The designed framework based on DiffusionDet and SMILEtrack achieves 58.2 *HOTA on the MOT17 validation set, proving this approach to be an effective candidate for multi-object tracking.*

1. Introduction

DiffusionDet proposes a novel framework for object detection based on diffusion models : bounding box prediction is considered as a denoising diffusion process (DDPM) [7], presenting random boxes as object candidates and achieving favorable performance on the COCO [9] and LVIS [4] datasets over well-established detectors (DETR, Sparse RCNN ...). We present a study of the DiffusionDet properties by replicating some of the paper's experiments. We then extend the model to the multi-object tracking task on the MOT17 dataset [12] and improve the model's performance, taking advantage of the diffusion process structure by adapting it to 2+ frames sequences. We propose modifications of the prior to move from independent frame processing to a video-adapted framework using information from previous frame detections.

2. COCO & LVIS Results

2.1. Detection Results

We reproduce results of the original paper using the pre-trained models available on the DiffusionDet Github [2]. The main accuracy results on COCO and LVIS are reported in Tables 1 and 2 for the Resnet50 [5] and the Swinbase backbone [10] architectures : they are close to identical to the reported ones in the official paper, demonstrating robust performance on these datasets.

Step	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Resnet 50						
1	44.99	64.08	48.39	26.80	47.73	61.22
4	46.12	65.97	49.66	28.62	48.67	61.59
8	**46.40**	**66.59**	**49.83**	**29.41**	**48.85**	**61.70**
SwinBase						
1	51.56	71.69	55.62	34.37	55.37	68.60
4	52.39	73.18	56.27	35.29	55.86	68.53
8	**52.67**	**73.44**	**56.90**	**36.01**	**56.10**	**68.73**

Figure 1. COCO detection results for Resnet50 & SwinBase.

The detection accuracy increases with the sample step size used on both datasets. Furthermore, the SwinBase backbone produces significantly more accurate results.

Step	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	AP_r	AP_c	AP_f
Resnet 50									
1	28.95	40.36	30.15	18.89	37.27	46.61	22.25	26.83	34.25
4	31.27	44.27	32.31	21.87	39.16	47.54	23.58	29.15	37.01
8	**31.93**	**45.28**	**33.21**	**22.76**	**40.15**	**47.80**	**24.01**	**29.74**	**37.84**
SwinBase									
1	39.30	52.88	41.41	26.57	48.63	60.94	33.73	38.16	43.03
4	41.36	56.18	43.58	29.49	50.66	**62.68**	33.67	40.46	45.73
8	**42.19**	**57.66**	**44.28**	**30.53**	**51.43**	62.53	**35.17**	**41.05**	**46.55**

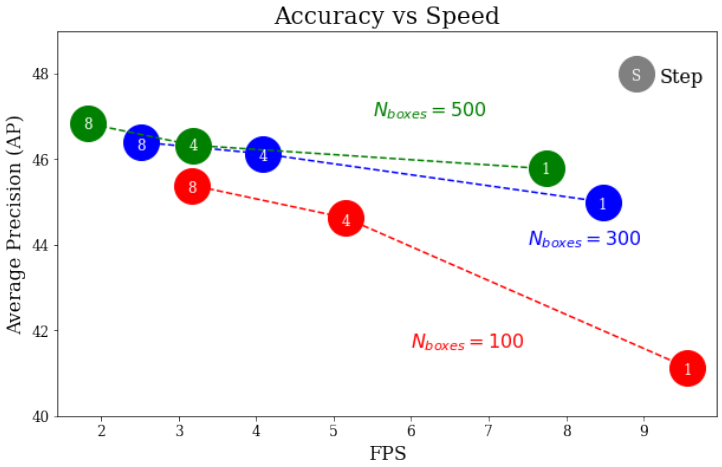
Figure 2. LVIS detection results for Resnet50 & SwinBase.

2.2. Progressive Refinement Analysis

DiffusionDet presents the significant advantadge of progressive refinement. This property can be emphasized in the following results reproduced on COCO using the resnet50 architecture. As expected, the detection accuracy increases with both the number of sample steps and box proposals while the processing speed decreases. Table 3a provides the numerical results for the experiment and Figure 3b illustrates the evolution of accuracy with respect to speed for the sampling steps and boxes used. We note that the gaps in accuracy and speed are far more pronounced from sample steps 1 to 4 and from 100 to 300 proposal boxes.

N_{boxes}	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	FPS
			1-Step				
100	41.13	57.96	44.32	22.37	43.96	59.41	**9.56**
300	44.99	64.08	48.39	26.80	47.73	61.22	8.47
500	**45.77**	**65.42**	**49.31**	**27.81**	**48.30**	**61.73**	7.74
			4-Step				
100	44.63	63.35	47.89	26.38	47.67	61.12	**5.15**
300	46.12	65.97	49.67	28.62	**48.67**	61.59	4.09
500	**46.32**	**66.39**	**50.08**	**28.98**	48.66	**61.86**	3.19
			8-Step				
100	45.39	64.80	48.77	27.84	47.79	61.33	**3.18**
300	46.40	66.59	49.83	29.41	48.85	61.70	2.52
500	**46.74**	**67.04**	**50.12**	**29.78**	**48.92**	**61.96**	1.83

(a) Progressive Refinement results on COCO using Resnet50 backbone.



(b) Plot of Progressive Refinement results on COCO using Resnet50 backbone.

3. DiffusionDet Adaptation for Pedestrians

3.1. MOT 17 Dataset

The MOT17 dataset [12] is a widely used benchmark dataset for multi-object tracking of pedestrians in video sequences. It includes various scenarios and challenges, different lighting conditions, occlusions, and object interactions. Object detections along with ground-truth trajectories for the objects are provided.

The dataset presents two different evaluation protocols, MOT17Det and MOT17. MOT17Det is designed to evaluate the performance of object detection methods, while MOT17 evaluates multi-object tracking methods. The dataset can thus be used for a complete evaluation of the detection and tracking pipeline.

Extending DiffusionDet to a multi-object tracking algorithm on the MOT17 dataset requires the detection decoder to perform well for pedestrian detections in crowded scenes. We propose to adapt the model pretrained on COCO and evaluate the model detections on MOT17Det. We can then implement a tracker on top and conduct an ablation study of the complete pipeline.

3.2. Crowdhuman Dataset

The CrowdHuman [13] dataset is one of the most recent and challenging benchmarks for object detection in crowded scenes focused on humans. It covers a wide range of object scales, poses, occlusions and provides a large number of annotations for each object. Training a model on CrowdHuman confers a significant advantage for the detection task on MOT17, as it simulates real-world scenarios commonly encountered in crowded scenes. Most of the best performing models on the MOT17 Challenge are pretrained on Crowdhuman.

3.3. Training for Pedestrian Detection

For these reasons, we decided to finetune the pre-trained COCO model with SwinBase backbone on the Crowdhuman dataset by changing the model heads (predict only one class). It was trained for $50K$ iterations (≈ 6.25 epochs) with SGD reproducing the setup of DiffusionDet COCO training, except for the number of epochs and the batch size. With the same setup it was then trained for $10K$ iterations (≈ 1.25 epochs) on half of the MOT17 training set.

4. DiffusionDet for Detections on Videos

4.1. From Images to Videos

For the object detection task, DiffusionDet processes each image independently. However, in videos, particularly with no camera motion, the object detection task can be improved using information from the previous frame. By considering information from previous frames, object detection algorithms can use temporal context and motion information to improve the accuracy of object detection in videos. This can be particularly useful in scenarios where objects are partially occluded or have similar appearance. Furthermore, it can reduce the search space, and thus increase computation speed.

4.2. DiffusionDet Prior Modifications

To adapt DiffusionDet to the object detection task for videos, two different methods are considered for improving the initialization of the random boxes prior to the inference stage. The first approach consists in directly incorporating the detections from the previous frame into the n_{boxes} random boxes. This process incorporates a small set predefined boxes into the following frame; we will refer to this method as mini anchors. The number of proposal boxes remains the same, but the n_d first random boxes are changed to the co-

ordinates of the n_d bounding box detections of the previous frame in the video. The motivation behind this method is that there is minimal motion of pedestrians in two consecutive frames of a video. As such, it may be beneficial to change some of the random boxes to previous detections, which are very likely to represent pertinent regions of interest to use as inputs for the image decoder and the DDIM.

The second method relies on the same idea, but increases the importance given to the previous frame by additionally generating n_+ random boxes with small random noise σ_+^2 around the detections of the previous frames. These are then incorporated among the randomly initialized random boxes, without changing the overall number of box proposals. These new boxes, if well initialized and in a sufficient number, should cover any movement of a pedestrian given its previous detection. It is then about finding a tradeoff between having enough boxes to cover any movement of previously detected objects and purely random boxes remaining to cover the whole image, allowing detection of lost and new objects. We implement this noisy anchor method with a first naive approach of $\sigma_+^2 = 1e^{-4}$ and $n_+ = \frac{n_{boxes}}{5}$ (thus replacing the first $\frac{n_{boxes}}{5}$ random boxes to maintain the number of box proposals). An illustration of both proposed prior modifications can be found at the end of the report in Figure 9.

5. Multi-Object Tracking with DiffusionDet

5.1. Location-based Trackers

DiffusionDet is an object detection algorithm which provides detections treating each frame independently. To adapt DiffusionDet results to the multi-object tracking task, it is therefore necessary to add an algorithm which incorporates a temporal dimension to associate detections in sequences of frames and assigns IDs.

The first type of trackers considered are location-based trackers, which assign IDs to detections based solely on their bounding box coordinates and the frame indices. Simple Online and Realtime Tracking (SORT) [1], which uses classical tracking techniques like the Kalman Filter and the Hungarian algorithm, is one of these trackers. The implementation used is from the multi-object tracking python library [3]. The SORT tracker is finetuned by setting the Kalman filter process noise scale to 0.3 and the measurement noise scale to 10 to maximize tracking performance.

5.2. Appearance-based Trackers

More recent multi-object tracking algorithms have started combining the location and appearance information of the detections. Typically, this is achieved using feature extraction on the detection bounding boxes with the aim of matching these features on subsequent frames for robust identification.

SMILETrack [14] is one of the algorithms that combines detection appearance and location for multi-object tracking, and is chosen for evaluation as it is ranked 1^{st} on the MOT17 challenge on papers with code. The algorithm uses Similarity Learning to extract important object appearance features, using Fast ReID [6], and a procedure to combine object motion and appearance features effectively. SMILEtrack also implements video camera movement correction for videos. The object motion branch of SMILETrack is the same as SORT, thus providing a fair experiment on the benefits of using appearance information.

5.3. Metrics for Tracking Evaluation

Standard multi-object tracking evaluation metrics include the Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Identity F1 Score (IDF1), Mostly Tracked (MT), Mostly Lost (ML), False Positive Rate (FP), False Negative Rate (FN), ID Precise (IDP), and ID switches (IDs). Out of these, the MOTA and IDF1 are two most commonly used metrics.

The Higher Order Tracking Accuracy (HOTA) [11] metric has recently gained popularity as it provides an evaluation of three tracking subtasks: detection, association and localization. It evaluates and combines aspects that aren't considered in the standard metrics. It is calculated from a set of metrics named the HOTA metrics defined by Luiten et al [11]. The metrics were computed using the official MOT17 evaluator [8].

6. MOT 17 Detection Results

6.1. Fine-tuning Results

Step	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	FPS
COCO Pretrained							
1	6.65	15.99	**5.01**	**0.90**	4.23	**17.47**	**2,7**
4	**6.79**	16.69	4.91	0.82	**4.45**	16.76	2.04
8	6.75	**16.87**	4.74	0.72	4.37	16.00	1.40
COCO + CrowdHuman							
1	29.49	59.40	26.12	6.44	19.02	42.31	**2,93**
4	31.86	65.78	27.39	**6.59**	22.84	43.23	2.05
8	**32.52**	**66.87**	**27.93**	6.36	**23.98**	**43.56**	1.26
COCO + CrowdHuman + MOT (validation)							
1	41.82	72.02	44.45	12.68	31.09	55.13	**2,91**
4	43.29	74.72	46.15	**13.58**	32.98	56.08	1.93
8	**43.68**	**75.40**	**46.40**	13.55	**33.52**	**56.30**	1.34

Figure 4. Progressive Refinement results on MOT validation using Resnet50 backbone.

The results in Table 6 show the benefit of the consecutive training on CrowdHuman and MOT. The accuracy is increased drastically while the computing speed stays the same for the number of sample steps. The results illustrate that the final detection model can be considered a robust basis for multi-object tracking.

6.2. Detection Performance for Prior Modifications

Prior Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Crowdhuman Model : testing						
Original	31.2	**59.8**	29.2	3.5	22.0	45.3
Mini anchors	**31.3**	**59.8**	**29.3**	3.5	**22.1**	45.3
Noisy anchors	31.2	59.0	**29.3**	**3.7**	21.9	**45.4**
MOT17 Model : validation						
Original	46.1	**71.9**	**51.4**	**20.7**	36.7	59.7
Mini anchors	46.2	**71.9**	**51.4**	20.3	**36.8**	**59.8**
Noisy anchors	**46.3**	**71.9**	**51.4**	20.3	36.6	**59.8**

Figure 5. Prior initialization methods evaluation for detection

The results in Table 5 show the benefits of adapting the DiffusionDet model to video sequences (2+ frames) through our prior change methods by comparing the detection results with or without our naive implementation of these methods. The results show a slight but consistent improvement. They indicate a potential benefit in performance, in particular with a more fine-tuned implementation of the noisy anchors prior adaptation. One could also potentially improve performance by considering slightly different approaches, such as making the noise proportional to the size of the bounding boxes, etc.

7. Multi-Object Tracking Results

7.1. Tracking Evaluation on MOT17

The results for the multi-object tracking task using DiffusionDet detections are reported in Figure 6. SMILEtrack with re-identification and interpolation produces the best results, with a HOTA value of 58.2. As a comparison, the best ranked tracker on the MOT17 challenge website (MrMOT) has a HOTA of 64.7 on the test set. Both re-identification and interpolation improve tracking results (0.5 and 0.4 points gained respectively for SMILEtrack).

7.2. Tracking with Prior Change

The ablation study of tracking results based on our naive implementation of the prior adaptation (Table 7 reinforces the idea of a potential strong benefit of theses method combined with progressive refinement, as the Mini Anchor model with track interpolation achieves the same $HOTA$ as the best model.

An interesting experiment would once again be to complete the fine-tuning of the noisy-anchor method and compare its tracking performance.

Prior Method	$HOTA$	$MOTA$	$IDF1$	$Recall$	$Precision$	Ids	MT	ML
Original	57.9	**66.9**	69.2	**75.9**	89.8	**782**	**50.4**	14.5
Noisy Anchors	57.4	66.5	68.6	74.6	**90.6**	814	47.6	**13.9**
Mini Anchors	**58.0**	66.5	**69.3**	74.8	90.5	809	48.9	14.1
With Interpolation								
Original	**58.2**	**67.0**	**69.7**	**76.0**	**90.0**	**794**	50.0	14.5
Noisy Anchors	57.5	66.7	68.7	75.7	89.9	814	49.8	13.9
Mini Anchors	**58.2**	66.7	69.4	75.9	89.7	809	**51.3**	**13.5**

Figure 7. Prior initialization methods evaluation

We conclude these experiments by emphasizing that maintaining consistently high tracking performance indicates the robustness of the DiffusionDet-based pipeline.

Video	$HOTA$	$MOTA$	$IDF1$	$Recall$	$Precision$	Ids	MT	ML
MOT-02	44.8	53.7	55.3	61.1	90.0	110	32.3	19.4
MOT-04	63.0	69.2	73.4	78.6	89.5	144	55.4	16.9
MOT-05	55.8	66.3	70.0	78.7	87.3	143	47.4	15.0
MOT-09	53.3	72.7	61.2	80.3	92.1	40	65.4	0
MOT-10	53.8	69.1	67.4	77.1	91.2	112	59.6	3.5
MOT-11	67.3	72.9	77.7	84.6	88.1	105	56.0	13.3
MOT-13	55.4	67.3	72.0	76.2	90.1	155	52.7	14.5
Combined	58.2	66.7	69.4	75.9	89.7	809	51.3	13.5

Figure 8. Best Tracking model with Mini Anchor

8. Conclusion

Our results suggest that DiffusionDet offers a promising framework for the multi-object tracking task, as it can

Tracker	Re-ID	Interpolation	$HOTA$	$MOTA$	$IDF1$	$Recall$	$Precision$	Ids	MT	ML
Finetuned SORT		×	53.9	64.2	61.8	73.4	**90.6**	2172	43.4	12.6
		✓	54.2	65.3	61.6	75.6	89.5	2172	48.0	**11.5**
SMILEtrack	×	×	57.4	66.3	68.8	74.5	90.5	**782**	47.6	14.3
	✓	×	57.9	66.9	69.2	75.9	89.8	**782**	**50.4**	14.5
	×	✓	57.8	66.4	69.4	74.6	90.5	794	47.6	14.3
	✓	✓	**58.2**	**67.0**	**69.7**	**76.0**	90.0	794	50.0	14.5

Figure 6. Tracking Results from DiffusionDet detections using SwinBase backbone, 8 sample steps and 500 proposal boxes.

achieve great detection performance and be adapted to 2+ frames video sequences. The final model achieves 58.2 validation HOTA which is close to the best test performance benchmarks, suggesting potential for more advanced models with fine-tuned prior modification.

9. Further Work

In the future, extensive analysis of the prior modifications is necessary to improve results. In particular, an ablation study of the prior modifications with respect to the progressive refinement properties could provide insights on the true performance improvement for the detection task on videos. In particular, it is expected that more pronounced differences in accuracy could be obtained for lower values of sampling steps and box proposals respectively.

Furthermore, other prior modifications should be tested and evaluated; for example, combining prior detections and generating boxes along the borders of the image could provide a better framework for the initialisation of new tracks.

References

[1] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, sep 2016. 3

[2] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection, 2022. 1

[3] Aditya M. Deshpande. Multi-object trackers in python. https://github.com/adipandas/multi-object-tracker, 2020. 3

[4] Agrim Gupta, Piotr Dollár, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation, 2019. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1

[6] Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. Fastreid: A pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631*, 2020. 3

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 1

[8] Arne Hoffhues Jonathon Luiten. Trackeval. https://github.com/JonathonLuiten/TrackEval, 2020. 3

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. 1

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 1

[11] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixe, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129(2):548–578, oct 2020. 3

[12] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking, 2016. 1, 2

[13] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018. 2

[14] Yu-Hsiang Wang, Jun-Wei Hsieh, Ping-Yang Chen, and Ming-Ching Chang. Smiletrack: Similarity learning for multiple object tracking, 2022. 3

Figure 9. Illustration of the proposed changes for box initialisation. Mini anchor method (top) and noisy anchor (bottom). The detections from the previous detections are plotted in green, the classic random boxes in white, and the boxes generated from the detections with noise for the noisy anchor method in blue.