S2F2: Single-Stage Flow Forecasting for Future Multiple Trajectories Prediction

Yu-Wen Chen, Hsuan-Kung Yang, Chu-Chi Chiu, and Chun-Yi Lee
Elsa Lab, Department of Computer Science, National Tsing-Hua University, Hsinchu, Taiwan

Abstract

*In this work, we present a single-stage framework, named **S2F2**, for forecasting multiple human trajectories from raw video images by predicting future optical flows. S2F2 differs from the previous two-stage approaches in that it performs detection, Re-ID, and forecasting of multiple pedestrians at the same time. Unlike the prior approaches, the computational burden of S2F2 remains consistent even as the number of pedestrians grows. The experimental results demonstrate that S2F2 is able to outperform two conventional forecasting algorithms and a recent learning-based two-stage model [1], while maintaining its tracking performance on par with the contemporary MOT models.*

1. Introduction

In the past few years, human trajectory forecasting from image sequences has been implemented as a two-stage process: (1) *the detection and tracking stage*, where targets in a single video frame are first located (i.e., detection), and then associated to existing trajectories (i.e., tracking) with or without the help of re-identification (Re-ID); and (2) *the forecasting stage*, where the previous trajectory of each person is fed into a forecasting model to predict its potential future locations over a short period of time. This branch of methods is referred to as the *two-stage approaches* in this work, and is illustrated in Fig 1 (a). Among them, previous works concentrated only on the second stage, and utilized the pre-processed bounding boxes and tracking histories [1–6]. Albeit effective, two-stage approaches inherently suffer from several limitations. First, their forecasting performances are constrained by the quality and correctness of the first stage. Second, despite that the first stage only processes the input in one pass, the second stage usually requires multiple passes of forecasting if the input image sequence contains multiple pedestrians [1, 4, 6].

In light of these shortcomings, a promising direction to explore is the use of a single-stage architecture. Single-stage architectures often possess favorable properties such as multitasking, fast inference speed, etc., and have recently been actively investigated in a wide range of other application domains [7–12]. Despite their successes, the previous single-stage approaches are mostly designed for tasks involving only a single image frame. The human trajectory forecasting task, however, requires temporal information encoded from multiple past frames, making previous single-stage architectures not readily applicable. As this problem setup has not been properly investigated, the challenges to be addressed are twofold. First, it requires various types of information (e.g., detection results, past trajectories, context features, etc.) to be concurrently encrypted to the latent features. Second, it necessitates temporal information to facilitate plausible predictions. Therefore, this human trajectory forecasting problem can be considered as a unique and complicated multitask learning task.

To this end, we present the first single-stage framework, called **S2F2**, for predicting multiple human trajectories from raw video images. S2F2 is inspired by the concept of optical flow forecasting, and is constructed atop the design philosophy of an anchor-free one-stage multiple object tracking (MOT) framework [7]. Fig. 1 highlights the differences between S2F2 and the prior two-stage approaches. S2F2 differs from them in that it performs detection, Re-ID, as well as forecasting of multiple pedestrians at the same time. Unlike two-stage approaches, the computational burden of S2F2 remains consistent even if the number of pedestrians grows. We show that with the same amount of training data, S2F2 is able to outperform two conventional trajectory forecasting algorithms and a recent learning-based two-stage model [1], while maintaining its tracking performance on par with the contemporary MOT models.

2. Methodology

2.1. Problem Formulation

Consider a sequence of raw images from a static scene $\{I_0, I_1, I_2, ...\}$, the objective of this work is to estimate and track the current and future locations of all pedestrians within the image sequence. Given the current timestep t and the information encoded from the previous trajectory, the objective of multiple human trajectory forecasting is to derive a set of bounding boxes $B_i^t = \{b_i^t, b_i^{t+1}, b_i^{t+2}, ..., b_i^{t+n}\}$ for the future n frames, where b_i^t stands for the bounding box of each identifiable pedestrian i at timestep t.

2.2. An Overview of the Proposed S2F2 Framework

Fig. 2 illustrates the S2F2 framework. To accomplish human trajectory forecasting for multiple pedestrians in a single stage, S2F2 employs two modules: (a) a *context feature extractor* for processing and encrypting the input image

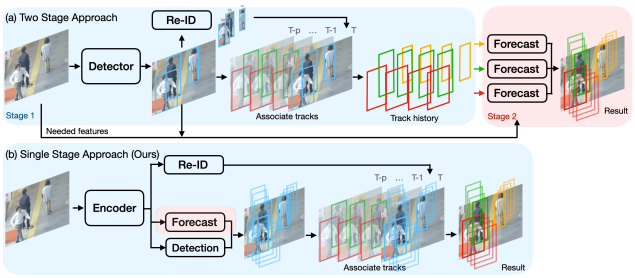

Figure 1. A comparison between the previous two-stage approaches, and our proposed one stage S2F2 framework.

frame of the current timestep t, and (b) a *forecasting module* for recurrently encoding the latent features and predicting the future optical flows, which are later used for deriving the future trajectories. Given a raw input image I_t, it is first processed by the backbone K of the context feature extractor to generate a feature embedding $\mathcal{X}_t$, which is used for three purposes: detection, Re-ID, and forecasting. To derive the future flow map, the forecasting module takes $\mathcal{X}_t$ as its input, and leverages a series of gated recurrent units (GRUs) to generate the optical flow map f_{t+n} corresponding to timestep $t+n$. This flow map represents the estimated offsets of each pixel from I_t to I_{t+n}, and thus can be used to perform forward warping of the detection results from timestep t to $t+n$ to derive the future bounding boxes b^{t+n} in a scene, as shown in Fig. 2 (i.e. the blue bounding boxes). Subsequently, all the bounding boxes are processed by a tracking algorithm, and are associated into distinct tracks, forming the final forecast b_i^{t+n} for each pedestrian i.

2.3. Context Feature Extractor

The context feature extractor of S2F2 inherits the design of [7], in which an enhanced version of Deep Layer Aggregation (DLA) [9] is used as the backbone to generate $\mathcal{X}_t$. The detection and Re-ID tasks are accomplished by four heads, including a heatmap, an offset, a size, and a Re-ID heads. Except predicting the bounding boxes and Re-ID features of I_t, these heads ensure that $\mathcal{X}_t$ can serve as an adequate representation of locations and object appearances, and offer sufficient information for the forecasting module.

2.4. Forecasting Module

2.4.1 GRU Encoder Block

The GRU Encoder encodes the context features $\mathcal{X}_t$ extracted by the backbone K from the context feature extractor. It is a single convolutional gated recurrent unit (ConvGRU) [13]. At timestep t, feature map $\mathcal{X}_t$ is passed as input into the ConvGRU along with the corresponding previous state S_{t-1} to compute the updated state $S_t = GRU(S_{t-1}, \mathcal{X}_t)$. S_t can thus be considered as a summary of the past context features up to timestep t.At timestep $t = 1$, $\mathcal{X}_1$ is employed for both the input and initial state.

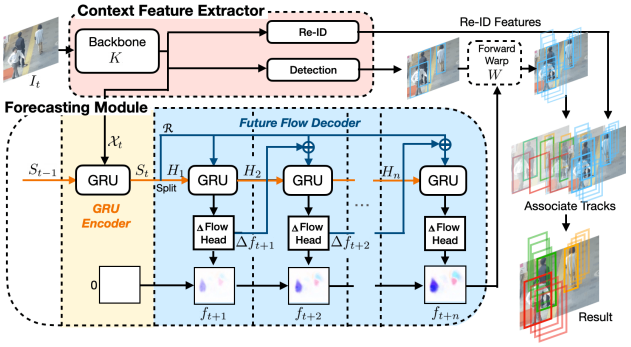

Figure 2. The proposed S2F2 architecture.

2.4.2 Future Flow Decoder Block

The future flow decoder's goal is to predict n residual future flow estimations $\{\Delta f_{t+1}, \Delta f_{t+2}, ..., \Delta f_{t+n}\}, \Delta f \in \mathbb{R}^{2\times w\times h}$, where each estimation is an update direction used to update a fixed flow field initialized with zeros. The main function of this decoder block is to create future flows $F = \{f_{t+1}, f_{t+2}, ..., f_{t+n}\}$, where $f_{t+1} = \Delta f_{t+1} + f_t$. Similar to the encoder block, the decoder also consists of a ConvGRU. It takes the encoded representation S_t and splits it into a hidden state H_1 and an input R. They are fed separately into the ConvGRU to generate the next hidden state $H_2 = GRU(H_1, R)$, which is then utilized by a Δ *flow head* to produce Δf_{t+1}. This, in turn, is used to generate the next input to the ConvGRU by concatenating Δf_{t+1} with R. The process repeats n times, with each iteration stands for a timestep into the future. To train the future flow decoder block, a loss function consisting of two parts is employed. The first part is a supervised loss formulated as:

$$\begin{aligned} F_{center} &= \sum_{f_t \in F} \sum_{i=1}^{N} \|c_i^{t+1} - \hat{c}_i^{t+1}\|_1 \\ &= \sum_{f_t \in F} \sum_{i=1}^{N} \|c_i^{t+1} - (c_i^t + W(c_i^t, f_{t+1})\|_1. \end{aligned} \tag{1}$$

where f_{t+1} is the estimated future flow, $W(\cdot,\cdot)$ is the warping operator, c_i^t and $\hat{c}_i^t$ represent the centers of the annotated bounding box b_i^t and the predicted bounding box $\hat{b}_i^t$ of person i at timestep t, respectively. For each center c^t, the forecasted center $\hat{c}^{t+1}$ can be inferred with the forward warping operation $\hat{c}^{t+1} = c^t + W(c^t, f_{t+1})$. The second part further refines and stabilizes the optical flow with the structural similarity index (SSIM) loss, formulated as:

$$F_{warp} = \sum_{f_t \in F} \sum_{x \in I} SSIM\Big(I_t(x), I_{t+1}\big(x + f_t(x)\big)\Big). \tag{2}$$

2.5. Online Tracking Refinement

We enhance the tracking performance by taking the estimated future bounding boxes $\hat{b}^{t+1}$ into consideration. We modify the original tracking algorithm of [7] by reducing the threshold δ_i of a bounding box if the distance between

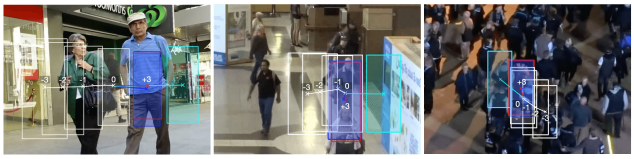

Figure 3. A comparison between the forecasting results made by S2F2 and CV-CS. From the left to the right, a pedestrian (1) walks away from the viewpoint, (2) makes a sharp turn due to the lockers in its way, and (3) makes a right turn to follow the crowd. Bounding boxes are highlighted in different colors to represent the ground truth (red), past locations (white), and the predictions made by CV-CS (aqua) and those made by S2F2 (dark blue). The predictions are one second into the future.

any of $\hat{c}^{t+1}$ and c_i^t is within a predefined range r, given by:

$$\forall i \in N, \delta_i = \begin{cases} \delta_i/2, & \text{if } \exists \hat{c}^{t+1}, \|\hat{c}^{t+1} - c_i^t\|_1 < r \\ \delta_i, & \text{otherwise.} \end{cases} \quad (3)$$

This allows the bounding boxes with lower confidence to be re-considered and associated if their estimated future locations are nearby, so as to further improve the performance.

2.6. Training Objective

We train our network model in an end-to-end manner by minimizing the objective function $L_{all} = \frac{1}{e^{w_1}} L_{det} + \frac{1}{e^{w_2}} L_{id} + \frac{1}{e^{w_3}} L_{fut} + w_1 + w_2 + w_3$, where L_{fut} is the summation of F_{center} and F_{warp}, w_1, w_2 and w_3 are learnable parameters, and L_{det} and L_{id} are the losses for detection and Re-ID. We use the uncertainty loss in [14] to automatically balance the detection, Re-ID, and forecasting tasks.

3. Experimental Results

3.1. Data Curation for Forecasting w.o. Ego Motion

We examine S2F2 on the subset of the MOT17 and MOT20 Challenge Datasets [15,16]. Since S2F2 focuses on the model's capability of encoding trajectories and forecasting, the movements from the camera are beyond the scope of this paper. As a result, we select a subset of videos involving no camera's ego motion from MOT17 and MOT20 to form our dataset, named StaticMOT[1]. We then train and evaluate S2F2 on it, with each video sequence split into halves to form the training and validation sets, respectively.

3.2. Trajectory Forecasting Results

To fairly compare different methods, the pre-processed bounding boxes and the necessary past trajectories of the pedestrians are generated by S2F2 from the validation set of StaticMOT. Tracks that are not continuously detected for six frames are discarded, resulting in around $470,000$ tracks for evaluation. We predict three future frames, corresponding to around one second of forecasting into the future.

[1]StaticMOT contains: MOT17-02, 04, 09, and MOT20-01, 02, 03, 05.

Table 1. The forecasting results on the StaticMOT validation set. The latency reported is evaluated on an NVIDIA Tesla V100 GPU.

Model	ADE(↓)	FDE(↓)	AIOU(↑)	FIOU(↑)	Latency (ms)
CV-CS	14.481	20.196	0.673	0.594	-
LKF [17]	20.635	24.323	0.581	0.512	-
STED [1]	16.928	23.761	0.654	0.570	623.480
Ours	**12.275**	**16.228**	**0.704**	**0.643**	**13.788**

Table 2. The detection results of S2F2 and FairMOT [7]. Those with * are taken from the original paper. The MOT17 test results are from the evaluation server under the private detection protocol.

Method	Dataset	MOTA [18] (↑)	MOTP [18] (↑)	IDs [18] (↓)	IDF1 [19] (↑)
FairMOT*	MOT17 test	69.8	-	**3996**	**69.9**
Ours	MOT17 test	**70.0**	80.15	4590	**69.9**
FairMOT*	MOT17 val	67.5	-	**408**	69.9
Ours	MOT17 val	**67.7**	80.3	513	**71.0**
FairMOT	StaticMOT	73.1	80.5	**2283**	76.4
Ours	StaticMOT	**73.6**	**80.5**	2307	**76.6**

(1) Quantitative Results: Table 1 shows the quantitative results in terms of the ADE/FDE and AIOU/FIOU metrics [1] of all methods on StaticMOT. CV-CS and LKF represent conventional trajectory forecasting algorithms Constant Velocity & Constant Scale linear motion model, and Linear Kalman Filter [17], respectively. STED [1] is a recent two-stage learning-based model with a similar GRU encoder-decoder architecture. The latency for forecasting is also included. Notice that latency is calculated for the forecasting stage only. It can be observed that, the proposed S2F2 outperforms all the baselines, while performing several times faster than STED. The degradation in STED's predictions might be partially due to the imperfect detection results generated by S2F2, as described in Section 3.2. Unlike the ground truth trajectories, the tracks from StaticMOT might have id-switches, occlusions, or miss-detections.

(2) Qualitative Results: Fig. 3 shows three examples of challenging scenarios selected and evaluated from our StaticMOT validation set. From left to right, the scenarios are: (1) a person behind two people walks away from the viewpoint, (2) a person moves to the right and takes a sharp turn due to the lockers in its way, and (3) a person makes a right turn to follow the crowd. In the first scenario, the person's bounding boxes from different timesteps becomes closer to each other due to the increase in their distances from the viewpoint. This can be forecasted by S2F2, but is unable to be correctly predicted by the CV-CS model. In the second scenario, CV-CS also fails to estimate the trajectory of the person. However, S2F2 incorporates features from the whole images, enabling it to anticipate this. In the third scenario, since S2F2 makes predictions for all objects concurrently based on a dense flow field, it is thus capable of capturing the spatial correlations between different objects, allowing it to forecast the future trajectory of the person by taking into account the behavior of the crowd. These three examples thus qualitatively validates S2F2's performance. More visualizations of S2F2's results are shown in Fig 4.

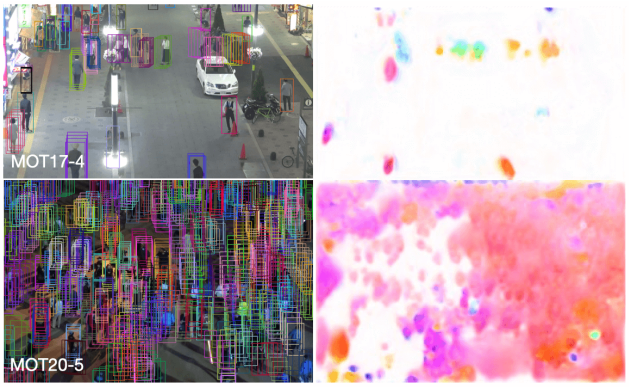

Figure 4. The tracking results and the predicted optical flow of our method on the validation set of StaticMOT.

3.3. Multiple Object Tracking Results

In addition to forecasting, Table 2 further compares the tracking results of S2F2 and FairMOT [7], the framework that S2F2 is based on. From top to bottom, the three categories correspond to the models trained on the whole official MOT17 training dataset [15], the training split of MOT17 from [7], and our StaticMOT, respectively. For each category, S2F2 and FairMOT are trained with the same set of data samples, and do not use any additional fine-tuning. It is observed from the results that our performance is on par or even slightly better than that of FairMOT for certain metrics, implying that the addition of our forecasting module does not affect its tracking capability. Note that S2F2 performs slightly worse than FairMOT in terms of the ID switch (IDS) metric. This might be due to the fact that FairMOT is trained on independent images, while S2F2 is trained on image sequences, thus causing slight overfitting.

4. Conclusion

In this paper, we presented the first single-stage framework, named S2F2 for predicting multiple human trajectories from raw video images. S2F2 performs detection, Re-ID, and forecasting of multiple pedestrians at the same time, with consistent computational burden even if the number of pedestrians grows. S2F2 is able to outperform two conventional trajectory forecasting algorithms, and a recent two-stage learning-based model [1], while maintaining its tracking performance on par with the contemporary MOT models. We hope this sheds light on single-stage pedestrian forecasting, and facilitates future works in this direction.

5. Acknowledgements

This work was supported by the Ministry of Science and Technology (MOST) in Taiwan under grant number MOST 111-2628-E-007-010. The authors acknowledge the financial support from MediaTek Inc., Taiwan, and would also like to acknowledge the donation of the GPUs from NVIDIA Corporation and NVIDIA AI Technology Center used in this research work. Finally, the authors thank the National Center for High-Performance Computing (NCHC) for providing computational and storage resources.

References

[1] O. Styles et al. Multiple object forecasting: Predicting future object locations in diverse environments. In *WACV*, pages 690–699, 2020.

[2] A. Gupta et al. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, pages 2255–2264, 2018.

[3] Ansar et al. Simple means faster: Real-time human motion forecasting in monocular first person videos on cpu. In *IROS*, pages 10319–10326, 2020.

[4] H. Yao et al. End-to-end pedestrian trajectory forecasting with transformer network. *ISPRS International Journal of Geo-Information*, page 44, 2022.

[5] B. Ivanovic et al. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *ICCV*, pages 2375–2384, 2019.

[6] O. Makansi et al. Multimodal future localization and emergence prediction for objects in egocentric view with a reachability prior. In *CVPR*, pages 4354–4363, 2020.

[7] Y. Zhang et al. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *IJCV*, pages 1–19, 2021.

[8] Z. Wang et al. Towards real-time multi-object tracking. In *ECCV*, pages 107–122, 2020.

[9] X. Zhou et al. Tracking objects as points. In *ECCV*, pages 474–490, 2020.

[10] P. Tokmakov et al. Learning to track with object permanence. *arXiv preprint*, 2021.

[11] B. Shuai et al. Multi-object tracking with siamese track-rcnn. *arXiv preprint*, 2020.

[12] Y. Yan et al. Anchor-free person search. In *CVPR*, pages 7690–7699, 2021.

[13] S. Xingjian et al. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, pages 802–810, 2015.

[14] A. Kendall et al. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018.

[15] A. Milan et al. MOT16: A benchmark for multi-object tracking. March 2016.

[16] P. Dendorfer et al. Mot20: A benchmark for multi object tracking in crowded scenes. March 2020.

[17] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. 1960.

[18] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[19] E. Ristani et al. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, pages 17–35, 2016.