

Forecasting Pedestrian Trajectory with Machine-Annotated Training Data

Olly Styles¹, Arun Ross² and Victor Sanchez¹

Abstract—Reliable anticipation of pedestrian trajectory is imperative for the operation of autonomous vehicles and can significantly enhance the functionality of advanced driver assistance systems. While significant progress has been made in the field of pedestrian detection, forecasting pedestrian trajectories remains a challenging problem due to the unpredictable nature of pedestrians and the huge space of potentially useful features. In this work, we present a deep learning approach for pedestrian trajectory forecasting using a single vehicle-mounted camera. Deep learning models that have revolutionized other areas in computer vision have seen limited application to trajectory forecasting, in part due to the lack of richly annotated training data. We address the lack of training data by introducing a scalable machine annotation scheme that enables our model to be trained using a large dataset without human annotation. In addition, we propose Dynamic Trajectory Predictor (DTP), a model for forecasting pedestrian trajectory up to one second into the future. DTP is trained using both human and machine-annotated data, and anticipates dynamic motion that is not captured by linear models. Experimental evaluation confirms the benefits of the proposed model.

I. INTRODUCTION

Interacting with humans in complex urban environments remains a challenging problem for autonomous vehicles (AVs). Unlike highways with well-defined rules for traffic, urban environments necessitate that vehicles interact with other road users, such as pedestrians and cyclists, in a more nuanced manner. For an AV to navigate effectively in such environments, the vehicle must be able to locate and react to pedestrians in order to avoid collisions. The first component of such a navigation system, detecting pedestrians, has seen a tremendous amount of research effort in the past decade [2]. If current trends continue, performance will soon match and even surpass human-level performance on standard evaluation benchmarks [39]. The rapid advancements in this area have led to real-world implementations of advanced driver assistance systems (ADAS) to aid drivers in critical situations. Such systems are capable of providing warnings or initiating braking if a pedestrian is detected in front of the vehicle, but are less reliable in the anticipation of potentially dangerous events before a pedestrian steps into the roadway.

As vehicles move towards ever greater autonomy, the need for accurate pedestrian trajectory forecasting also grows. With a human driver in the loop, ADAS may be designed conservatively as false negatives can be tolerated. For an

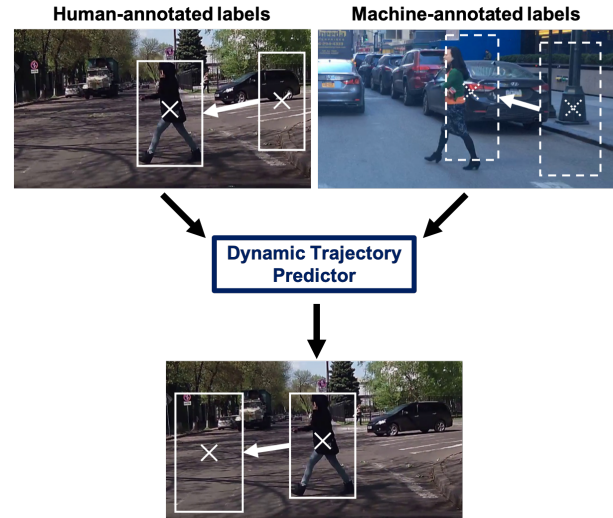


Fig. 1. We propose a model and training regime for pedestrian trajectory forecasting. Due to a lack of annotated training data, our model is trained jointly with human-annotated and machine-annotated pedestrian bounding boxes generated by a pedestrian detection and tracking algorithm.

AV, however, the reliable anticipation of pedestrian *intent* is a critical safety feature but a complex challenge. Although driven by long-term motion goals such as reaching a specific destination [26], pedestrian motion is highly dynamic and may change at a moment's notice, such as a child running rapidly into the street. To deal with this uncertainty, human drivers use heuristics such as pedestrian head pose, gait, and scene dynamics to reason about intent [23]. Without these cues, for example, human drivers find it more challenging to predict if a pedestrian is about to cross the road [28].

Modern vehicles equipped with sensors such as LiDAR and Radar can build an accurate representation of the surrounding environment [41]. Both LiDAR and Radar, however, lack the capability for extracting high-resolution features and are, thus, commonly supplemented with visible spectrum cameras. Manual annotation of features such as pedestrian head pose and body language cues from camera data is challenging and time-consuming. Furthermore, pedestrian behavior varies across different cultures and driving environments [8]. A model trained to anticipate pedestrian behavior in California, USA is unlikely to perform well on the streets of Mumbai, India. Without a practical method for learning from unlabelled data, it is likely that large quantities of data must be manually annotated for deployment in each environment.

Based on the above observations, we present a system

¹Olly Styles and Victor Sanchez are with the Department of Computer Science, University of Warwick, Coventry, UK {o.c.styles | v.f.sanchez-silva}@warwick.ac.uk

²Arun Ross is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, USA rossarun@cse.msu.edu

for pedestrian trajectory forecasting capable of learning from unlabeled data. The two main contributions of this work are as follows:

- 1) Dynamic Trajectory Predictor (DTP), a pedestrian trajectory forecasting deep learning model based on motion features from optical flow.
- 2) A machine annotation scheme for training trajectory forecasting models in the absence of labeled data.

II. RELATED WORK

Our proposed approach builds on the substantial progress made in pedestrian detection and human action recognition. However, in this section, we concentrate on literature more directly relevant to our contributions, that are focused on (a) pedestrian trajectory forecasting and (b) alternative supervision methods for training models in the absence of large-scale human annotated datasets. For pedestrian detection, see recent surveys such as [2], [40]. For action recognition, see recent surveys such as [10], [15].

A. Pedestrian Trajectory Forecasting

Dynamic Systems Approach. Given the absence of large pedestrian trajectory datasets, previous works have modeled the dynamic motion of pedestrian's using linear dynamic systems (LDS) that combine the assumptions of constant velocity (CV) or constant acceleration (CA) with a filtering algorithm such as the Kalman filter [29]. To model non-linear, dynamic motion, a switching linear dynamic system (SLDS) uses a discrete Markov chain to select between multiple LDS at each timestep based on past observations. However, the SLDS is limited to *reacting* to pedestrian motion rather than *anticipating* a change in dynamics. To address this issue, existing works [17], [16] focus on additional cues such as pedestrian head pose, motion state, and road scene context or use a non-linear filtering algorithm such as the unscented Kalman filter [21].

Data-Driven Approach. Data-driven approaches for trajectory forecasting have gained attention in recent years resulting from the success of deep learning models for related problems such as image classification, action recognition, and pedestrian detection. In particular, deep learning models have been applied to trajectory forecasting in a surveillance setting with a fixed overhead camera on datasets such as UCY [18] and ETH [22], or forecasting vehicle trajectories [34]. In [35], pedestrian trajectory is forecast by encoding pedestrian location as a sparse vector which is used directly as input to a convolutional neural network (CNN). In [1], pedestrian trajectory is forecast from a static, overhead camera using a long short-term memory (LSTM) network. The authors introduce social pooling, which models the social interactions between multiple pedestrians.

Trajectory forecasting is considered from a first-person perspective in [33]. The authors propose a model combining features from the pedestrians pose, estimated ego-motion, and past location information. Similarly, in [3], an LSTM is used to predict the future location of pedestrian bounding boxes by first estimating future ego-motion and then using

these estimates with observed bounding boxes to forecast the location of future bounding boxes. All data-driven approaches, however, are limited by the lack of available training data.

B. Alternative supervision

Supervised learning has been a prominent learning paradigm requiring accurate annotation of datasets, which is commonly completed manually through painstaking human effort. Due to the massive quantities of data necessary to effectively train state-of-the-art models, several alternative means of supervision have been proposed. Pre-training neural network models on the large Imagenet dataset [6], before fine-tuning on a target dataset, has become the de facto standard in settings where annotated data is limited. Alternative means of building large annotated datasets for pre-training such as mining social media websites [20] have also been proposed. An alternative learning paradigm is self-supervision. In self-supervision, some subset of a dataset is withheld during the training process, and a model is trained to predict the withheld data. In this way, a model may exploit large-scale datasets without expensive annotation. For example, the authors in [38] convert color images to greyscale and train a model to perform the inverse operation, and the authors in [7] predict the location of image patches in relation to another patch. In an intelligent vehicle setting, existing works have used data collected by one sensor (such as a camera) to predict the data collected by another sensor (such as an inertial measurement unit) [13], [4]. Self-supervision avoids the expensive human annotation component of supervised learning and is, therefore, well-suited to address problems with limited annotated data. Our proposed machine annotation scheme enables us to leverage the power of self-supervision for pedestrian trajectory forecasting.

III. PROPOSED METHOD

A. Problem formulation and baseline

Consider a pedestrian localized in a video with an associated set of bounding box coordinates for the current and past m frames, such as in Fig 1. Our goal is to predict the centroid of future bounding boxes with coordinates x_t and y_t as to anticipate potentially dangerous events, such as a pedestrian stepping into the roadway. The horizontal and vertical components of velocity, v_t^x and v_t^y respectively, at time t of a pedestrian relative to the vehicle in the 2D projection obtained by a camera can be estimated by taking the first order derivative of the past centroids:

$$v_t^x = \frac{x_t - x_{t-m}}{m}, \quad v_t^y = \frac{y_t - y_{t-m}}{m} \quad (1)$$

As a baseline, we consider that the pedestrian maintains their average velocity of the previous m timesteps in the future n timesteps:

$$\tilde{x}_{t+n} = x_t + v_t^x \cdot n, \quad \tilde{y}_{t+n} = y_t + v_t^y \cdot n \quad (2)$$

We denote a pedestrian's centroid location at time t as L_t , comprised of coordinates x_t and y_t . Similarly, we denote

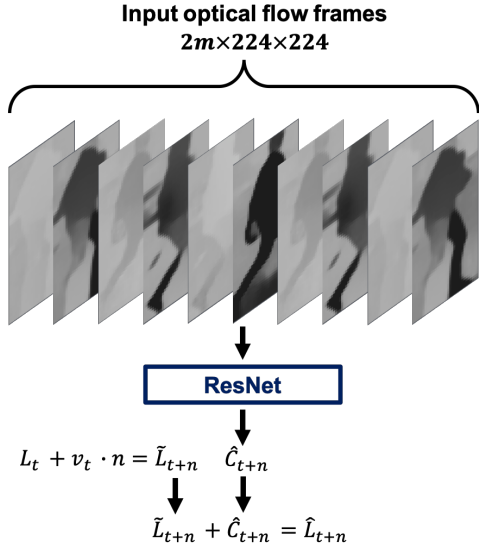


Fig. 2. DTP forecasts pedestrian trajectory relative a constant velocity baseline. We use ResNet [9] with modified input and output layers to compute features from past optical flow. See Section IV-B for details.

velocity at time t as v_t , comprised of vertical and lateral velocities v_t^x and v_t^y . The predicted location of the centroid at time $t + n$ following the constant velocity assumption is denoted as:

$$\tilde{L}_{t+n} = L_t + v_t \cdot n \quad (3)$$

We focus here on predicting the centroid in the 2D coordinate space obtained by a camera, rather than the 3D world coordinates required for full localization by an AV. In practical applications, 2D object detections may be associated with 3D world coordinates using a depth estimation method such as [11].

B. Dynamic Trajectory Predictor

In many scenarios, such as when a pedestrian is stationary or walking at a constant speed, the constant velocity assumption is a reasonable predictor of future location. Challenging situations are instances that deviate significantly from this assumption. An effective model must anticipate a change in velocity and adjusts predictions accordingly. The error resulting from the constant velocity assumption is denoted by:

$$\tilde{e}_{t+n} = |L_{t+n} - \tilde{L}_{t+n}| \quad (4)$$

Rather than directly predicting a location \hat{L}_{t+n} directly, existing works [33], [3] output the location relative to the last observed timestep, $\Delta L_{t+n} = L_{t+n} - L_t$. In contrast, we propose to output a compensation term, $C_t = -\tilde{e}_t$, which corrects for errors in the constant velocity assumption. In this way, our model is first initialised to a strong baseline (in the case where $C_t = 0$, the model's predictions equal constant velocity) and then fine-tunes predictions on training examples for which the constant velocity assumption results in errors. The final predicted coordinates in the original 2D image projection, \hat{L}_{t+n} , are then recovered as follows:

$$\hat{L}_{t+n} = \tilde{L}_{t+n} + \hat{C}_{t+n} \quad (5)$$

Inspired by effective action recognition models [30], [31], DTP uses a stack of optical flow frames as input to a CNN that extracts a compact representation of human motion. From this feature vector, a fully connected layer outputs a prediction \hat{C}_{t+n} representing the estimated correction factor. A vector of large magnitude indicates that the pedestrian velocity will increase or decrease, whereas a vector of magnitude close to 0 indicates that the pedestrian will maintain their current velocity. We use ResNet [9] as our backbone network, owing to its consistently good performance on many vision tasks. A high-level diagram of our model is shown in Fig. 2. Further details of the architecture modifications are outlined in Section IV-B.

C. Machine annotation

The training of trajectory forecasting models in a supervised learning setting requires dense (per-frame) bounding box annotation of pedestrians, which are expensive to obtain by hand. For this reason, the number and size of datasets with densely annotated pedestrian bounding boxes is limited. The size of existing datasets [17], [24] is prohibitive for the training of high-capacity deep learning models, which rely on large quantities of data to learn an effective feature representation. To overcome this issue, we propose to learn from unlabeled data by using an automated pedestrian detection and tracking algorithm to generate bounding boxes without human labor.

Given an input video sequence, pedestrian detection algorithms obtain an estimate of the location L_t for each pedestrian, and a tracking method then links these estimated locations across each timestep t . Given a set of such detections, we adopt the self-supervision learning paradigm by training our model to predict future pedestrian locations, L_{t+n} , given only the current and past locations, viz., $L_{t-m} \dots L_t$.

A similar annotation process is proposed in [33], in which pedestrians are detected and tracked using [5]. However, automated detectors do not perform on par with human annotators, and make different errors to humans, such as false positive detections of vertical structures [39]. Due to this, it is not evident that models trained on machine-annotated data will generalize across datasets and to human-annotated data. To verify our proposed machine-annotation regime, we validate the performance of our model on a human-annotated dataset. We adopt the conventional methodology of pre-training on a large dataset before fine-tuning on a smaller target dataset, intending to improve generalizability on the target dataset [36].

IV. EXPERIMENTS

A. Datasets

We use two datasets in our experiments, JAAD [24] and BDD-100K [37]. Both datasets consist of videos captured by a front-facing camera mounted behind a windshield collected by cars driving on public roads in Europe and



Fig. 3. Example pedestrians with associated optical flow obtained using FlowNet2-CSS. Left 3 images are human-annotated pedestrians from the JAAD dataset, right 3 images are pedestrians detected on the BDD-100K dataset using YOLOv3. Optical flow captures motion resulting from both the camera and pedestrian.

North America. The JAAD dataset contains dense pedestrian bounding box annotation, that is, annotations are provided for each frame. BDD-100K, however, contains sparse bounding box annotation. Only one frame per video is annotated. We do not use the sparsely annotated bounding boxes. Due to the huge number of videos in BDD-100K, we use only the first 10,000 videos. This subset is henceforth referred to as BDD-10K. Example pedestrian images from both datasets are shown in Fig. 3 (first row). Videos from the JAAD dataset are downsampled with bilinear interpolation to match the BDD-10K dataset resolution of 1280×720 . The frame rate of both datasets is downsampled from 30 to 15 frames per second to reduce redundancy between consecutive frames.

B. Dynamic Trajectory Predictor

Implementation. To evaluate DTP, we use the JAAD dataset. Pedestrians smaller than 50 pixels in height, occluded pedestrians, and tracks shorter than 25 frames are discarded. Optical flow is extracted from cropped pedestrians using the provided human-annotated bounding boxes with the FlowNet2-CSS algorithm [12]. Pixel displacements are clipped at ± 50 and scaled to the range $[0, 1]$. Example pedestrian flow images are shown in Fig. 3 (second row).

We use a stack of m horizontal and m vertical optical flow frames at timesteps $t - m$ to t . Features are computed from the $2m$ input channels using the ResNet-18 CNN architecture [9]. We modify the first convolutional layer to use $2m$ input channels rather than 3, keeping other dimensions the same. We replace the 1000-D softmax output layer with a 30-D fully connected layer which produces predictions for the x and y coordinates of the 15 future bounding box centroids. We use cross-modality pre-training and partial batch normalization [31] to initialize our CNN with ImageNet weights. The model is optimized to minimize the \mathcal{L}_2 loss between the true and predicted future locations, $L_{t+1} \dots L_{t+n}$ and $\hat{L}_{t+1} \dots \hat{L}_{t+n}$, and is trained until convergence using the Adam [14] optimizer with an initial learning rate of 10^{-5} , which is reduced to 10^{-6} once performance saturates. We use a batch size of 64 and a weight decay of 10^{-2} . Each pedestrian is resized to 256×256 pixels. For data augmen-

tation, a randomly cropped sub-image of size 224×224 is taken.

We split the JAAD dataset into training (videos 0-250) and testing (videos 251-346) sets. We perform 5 fold cross-validation on the JAAD training set to tune hyperparameters. Once hyperparameters are fixed, we obtain an estimate of the model’s generalizability by training on each of the 5 folds until performance on the respective validation set saturates. We then evaluate the model on the test set. We report the mean performance on the test set with associated 95% confidence intervals for the 5 folds.

Evaluation. We use two metrics to evaluate model performance, mean squared error (MSE) and displacement error (DE@ t) at timesteps up to 15, following [3], [33]. The MSE is the mean of the squared errors of the predicted centroid in pixels from all timesteps 1 to n and across all samples in the test set. The DE@ t is the mean Euclidean distance in pixels of the predicted and ground truth centroid for timestep t only. Both metrics are relative to an image resolution of 1280×720 .

We evaluate our proposed approach with 4 different inputs: a single RGB frame at time t , a single optical flow frame at time t , a stack of 5 optical flow frames at times $t - 4$ to t , and a stack of 9 optical flow frames at times $t - 8$ to t . We use 9 as our maximum value of m rather than the 10 frames commonly used for action recognition [30], [31] for a fair comparison with Future Person Localization (FPL) [33], which uses 10 frames as input. As each optical flow frame requires two consecutive RGB frames to be computed, using 10 input frames results in 9 optical flow frames. Following prior works [33], [1] we adopt constant velocity (CV) and constant acceleration (CA) as baselines. For the CV baseline, we compute the average velocity in the image space using the previous locations and predict the future location assuming the pedestrian maintains a linear velocity. Similarly, for the CA baseline, we compute the average acceleration using the previous locations and extrapolate these values into the future timesteps assuming linear acceleration. Using 4 previous locations resulted in the best cross-validation performance.

Results. Table I shows the performance of each model with different input modalities in comparison with the CV and CA baselines. Due to the relatively poor performance of the RGB input, we do not fuse RGB and optical flow models as in the two-stream model [30]. Example outputs of our model using a stack of 9 optical flow frames compared to baselines are shown in Fig. 4. DTP performs particularly well in situations where a pedestrian first begins walking, and when the ego-vehicle begins to turn sharply (top two rows). DTP performs less well under conditions of significant background motion such as those due to other vehicles (bottom left image) or upper body motion in the counter walking direction (bottom right image).

We compare our method using a stack of 9 optical flow frames with linear baselines and FPL [33] in Table II. We modify FPL to output 15 timesteps into the future rather than the 10 as in the original architecture and use optical flow for ego-motion estimation as described in [33]. Both DTP and

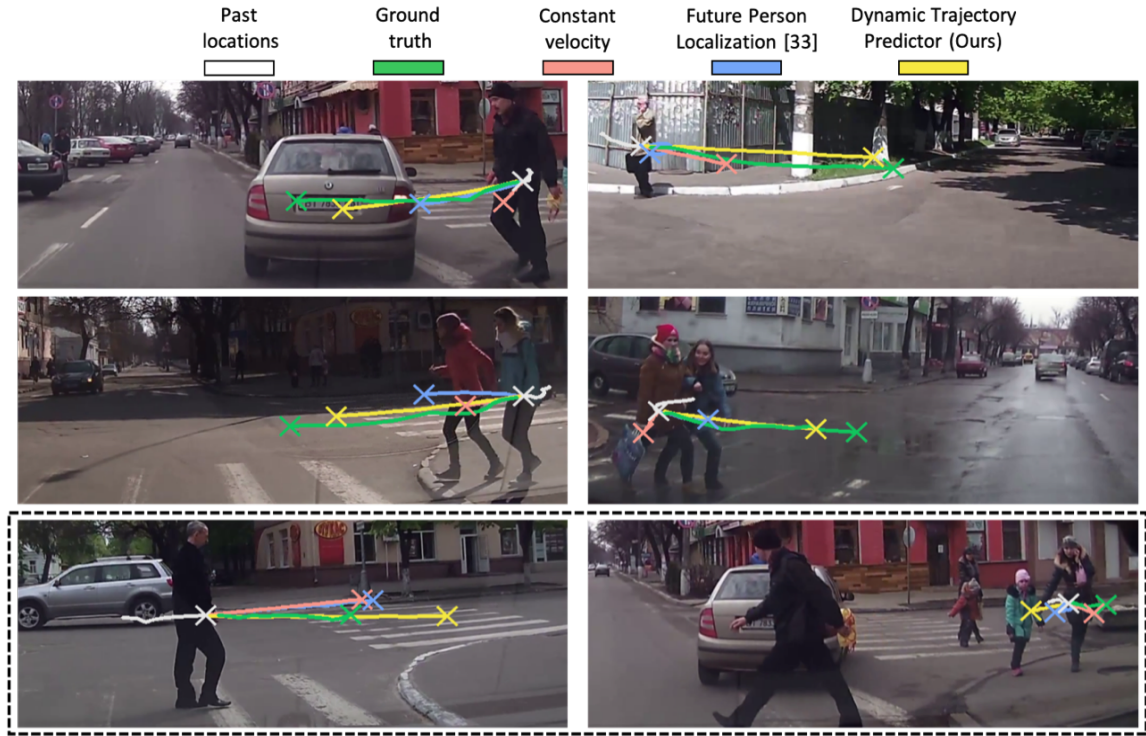


Fig. 4. Example successful (top 2 rows) and unsuccessful (bottom row) trajectory forecasts on the JAAD test set. See main text for discussion. Best viewed in color.

FPL see a reduction in error with our proposed CV correction term C_t (rather than directly predicting the location displacement ΔL_{t+n}). DTP attains the best performance.

TABLE I
INPUT MODALITY COMPARISON.

| Input modality | MSE | DE@5 | DE@10 | DE@15 |
|------------------------------|------------|------------|-------------|-------------|
| CA | 1426 | 15.3 | 28.3 | 52.8 |
| CV | 1148 | 16.0 | 26.4 | 47.5 |
| RGB frame | 1042 | 11.6 | 24.9 | 45.2 |
| Optical flow frame | 873 | 11.1 | 23.0 | 41.2 |
| 5 optical flow frames | 651 | 9.4 | 19.3 | 35.6 |
| 9 optical flow frames | 610 | 9.2 | 18.7 | 34.6 |

TABLE II
MODEL COMPARISON.

| Model | CV correction term | MSE | DE@15 |
|------------|--------------------|-----------------|-------------------|
| FPL [33] | ✗ | 1405 ± 182 | 49.5 ± 2.9 |
| FPL [33] | ✓ | 881 ± 44 | 41.3 ± 1.2 |
| DTP | ✗ | 1404 ± 94 | 54.6 ± 2.6 |
| DTP | ✓ | 610 ± 21 | 34.6 ± 0.5 |

C. Machine annotation

Implementation. We annotate pedestrian bounding boxes in the BDD-10K dataset using two popular off-the-shelf object detectors, YOLOv3 [25] and Faster-RCNN [27]. Although the detectors are capable of detecting a variety of objects, we use the pedestrian class only. Our aim here is to evaluate the robustness of our proposed system to multiple automated detectors, rather than to compare detector performance directly. Nonetheless, for consistency, we train both detectors on the same dataset (MS-COCO [19]) and threshold confidence scores at 0.6.

Once frame-wise detections are obtained, detections are associated across frames using the Deepsort [32] tracking-by-detection algorithm resulting in a series of bounding boxes and tracking identifiers. We use the same setup as the JAAD dataset and discard detections with height fewer than 50 pixels, and tracks shorter than 25 frames. Using this annotation scheme, we find a total of 16,900 valid non-overlapping pedestrian tracks using YOLOv3 and 13,200 using Faster-RCNN.

Evaluation. We use an 80%-20% training-validation split for BDD-10K. We pre-train DTP on BDD-10K using the same hyperparameters as outlined in Section IV-B. Once performance on the validation set saturates, the model is fine-tuned on the JAAD training set. We evaluate the trajectory forecasting performance with and without pre-training rather than the pedestrian detection quality, owing to the lack of human-annotated bounding boxes.

Results. The impact of machine-annotated pre-training using the YOLOv3 detector before fine-tuning on the human-annotated JAAD dataset is shown in Table III.

TABLE III

IMPACT OF PRE-TRAINING ON BDD-10K WITH YOLOV3.

| Model | Pre-training with machine annotation | MSE | DE@15 |
|------------|--------------------------------------|-----------------|-------------------|
| FPL [33] | ✗ | 881 ± 44 | 41.3 ± 1.2 |
| FPL [33] | ✓ | 805 ± 46 | 40.1 ± 1.2 |
| DTP | ✗ | 610 ± 21 | 34.6 ± 0.5 |
| DTP | ✓ | 539 ± 13 | 32.7 ± 0.4 |

We evaluate the impact of pre-training dataset size and pedestrian detector by training on subsets of BDD-10K ranging from 20% to 100% of the total dataset size. Fig.

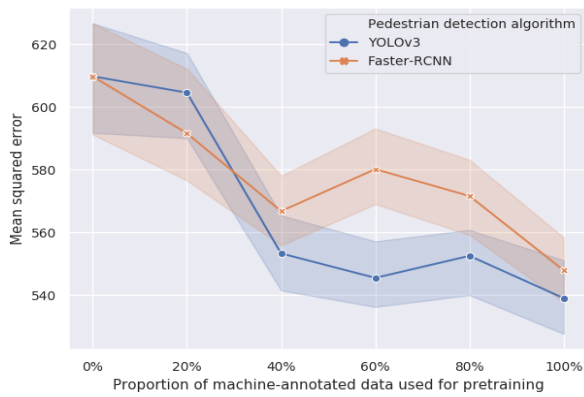


Fig. 5. Impact of pre-training dataset size and pedestrian detection algorithm on the performance on JAAD test set. Shaded areas show the 95% confidence interval.

5 shows the MSE on the JAAD test set for both YOLOv3 and Faster-RCNN. In general, the error on the JAAD test set reduces as larger subsets of our machine-annotated dataset, BDD-10k, is used for pre-training. The reduction in error may be due to the model's ability to learn the motion patterns of under-represented classes, such as children or the elderly, from a larger dataset.

V. CONCLUSION

We have presented a model and complementary machine annotation scheme for pedestrian trajectory forecasting from onboard a moving vehicle. Our model, DTP, forecasts trajectory for time horizons up to one second by anticipating a change in velocity using optical flow information. By introducing a method for annotating data without human labor, DTP and other similar models may leverage large-scale datasets for learning effective feature representations.

ACKNOWLEDGMENT

This work is funded by the UK EPSRC (grant no. EP/L016400/1) and the EU Horizon 2020 project IDENTITY (Project No. 690907). Portions of this work were done when Styles was at MSU. Our thanks to NVIDIA for supporting this research with their generous hardware donation.

REFERENCES

- [1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*, 2016.
- [2] R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In *ECCV*, 2014.
- [3] A. Bhattacharyya, M. Fritz, and B. Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *CVPR*, 2018.
- [4] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv:1604.07316*, 2016.
- [5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.
- [6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [7] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.
- [8] B. Färber. Communication and communication problems between autonomous vehicles and human drivers. In *Autonomous Driving*, pages 125–144. Springer, 2016.

- [9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [10] S. Herath, M. Harandi, and F. Porikli. Going deeper into action recognition: A survey. *Image and vision computing*, 60:4–21, 2017.
- [11] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *T-PAMI*, 30(2), 2008.
- [12] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.
- [13] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *ICCV*, 2015.
- [14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.
- [15] Y. Kong and Y. Fu. Human action recognition and prediction: A survey. *arXiv:1806.11230*, 2018.
- [16] J. F. Kooij, F. Flohr, E. A. Pool, and D. M. Gavrila. Context-based path prediction for targets with switching dynamics. *IJCV*, 2018.
- [17] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila. Context-based pedestrian path prediction. In *ECCV*, 2014.
- [18] A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. In *Computer Graphics Forum*, 2007.
- [19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.
- [20] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the limits of weakly supervised pretraining. *arXiv:1805.00932*, 2018.
- [21] M. Meuter, U. Irgel, S.-B. Park, and A. Kummert. The unscented kalman filter for pedestrian tracking from a moving host. In *Intelligent Vehicles Symposium*. IEEE, 2008.
- [22] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009.
- [23] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. Agreeing to cross: How drivers and pedestrians communicate. In *Intelligent Vehicles Symposium*, 2017.
- [24] A. Rasouli, I. Kotseruba, and J. K. Tsotsos. Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior. In *ICCV Workshop*, 2017.
- [25] J. Redmon and A. Farhadi. Yolo3: An incremental improvement. *arXiv:1804.02767*, 2018.
- [26] E. Rehder and H. Kloeden. Goal-directed pedestrian prediction. In *CVPR workshop*, 2015.
- [27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.
- [28] S. Schmidt and B. Färber. Pedestrians at the kerb—recognising the action intentions of humans. *Transportation research part F: traffic psychology and behaviour*, 12(4), 2009.
- [29] N. Schneider and D. M. Gavrila. Pedestrian path prediction with recursive bayesian filters: A comparative study. In *German Conference on Pattern Recognition*, 2013.
- [30] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.
- [31] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*. Springer, 2016.
- [32] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, 2017.
- [33] T. Yagi, K. Mangalam, R. Yonetani, and Y. Sato. Future person localization in first-person videos. In *CVPR*, 2018.
- [34] Y. Yao, M. Xu, C. Choi, D. J. Crandall, E. M. Atkins, and B. Darius. Egocentric vision-based future vehicle localization for intelligent driving assistance systems. *arXiv:1809.07408*, 2018.
- [35] S. Yi, H. Li, and X. Wang. Pedestrian behavior understanding and prediction with deep neural networks. In *ECCV*, 2016.
- [36] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, pages 3320–3328, 2014.
- [37] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv:1805.04687*, 2018.
- [38] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *ECCV*. Springer, 2016.
- [39] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele. How far are we from solving pedestrian detection? In *CVPR*, 2016.
- [40] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele. Towards reaching human performance in pedestrian detection. *T-PAMI*, 2018.
- [41] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung. Overview of environment perception for intelligent vehicles. *ITS*, 18(10), 2017.