Predicting Pedestrian Trajectories for Autonomous Driving Applications

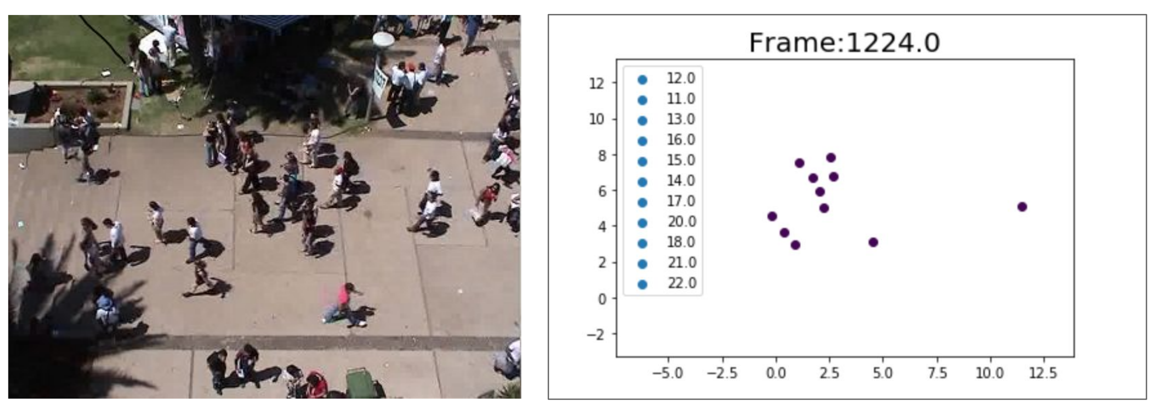
Ashish Roongta, Bryan Zhao, Pranav Narahari, Yayati Jadhav
{aroongta, bryanz, pnarahar, yayatij}@andrew.cmu.edu
Carnegie Mellon University

Motivation

- Modeling the interactions between autonomous vehicles (AVs) and pedestrians is currently a critical issue for public safety [1]. Accurate predictions of pedestrian trajectories would allow AVs to safely navigate around pedestrians in high-risk scenarios.
- The goal of this project is to predict future steps, or trajectories, of civilians in crowds based on their previous states.
- Our approach benchmarks the performance of conventional machine learning methods (**linear regression**, **KNN regression**) against deep learning frameworks, such as **GRU** and **Vanilla LSTM**.

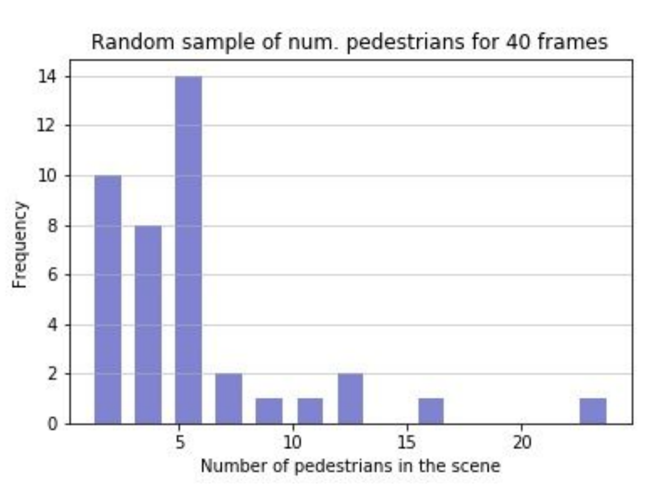
Dataset Overview

Various trajectories were recorded and manually annotated from aerial viewpoints at 25 fps [2].



Video of pedestrians from dataset and processed numerical coordinates with labeled pedestrian IDs [2].

- These datasets of pedestrians from ETH, UCY, and Stanford are publicly available [3, 4].
- The highly stochastic nature of pedestrian trajectories warrants extensive data preprocessing. Sequence length, trajectory curve as well as the number of pedestrians in each frame is highly variable.
- For KNN regression, the data was segregated based on pedestrian ID. Pedestrian trajectories were divided into sequences of 6 steps. The first 5 steps are used to train the model and the 6th acts as the ground truth for the predicted value.
- For RNN, the sequences of pedestrians with trajectory length lesser than observed predicted were ignored, trajectory sequences greater than that were sliced.



Pedestrian trajectories are highly erratic and variable from scene to scene.

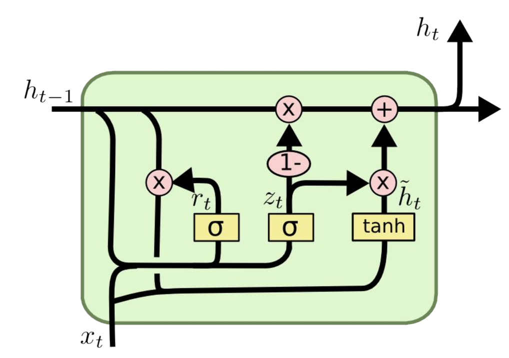
Proposed Methods

After processing the pedestrian datasets for the different learning frameworks, we compare the performance between traditional ML algorithms and specific RNNs.

Linear regression is first used to set a baseline and determine how closely we can predict the pedestrians' trajectories using a simple and naive machine learning algorithm.

We then utilize KNN to find the nearest neighbours of the test set, and then run linear regression for trajectory estimation based on the distance between the point and its neighbours.

Since predicting pedestrian trajectories are better framed as a sequence generation task, scene layout RNNs such as LSTM/GRU are particularly suited for their abilities to capture and generate steps from prior states. GRUs are another variant of LSTM that are more computationally efficient and expose full hidden states without any control [1].



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Layout of GRU networks, a variant of LSTM [5].

Finally, we utilize mean-squared-error (MSE) as the loss metric, and report prediction errors with two primary metrics, similar to Alahi et al. [1]: **average displacement error** (MSE over all estimated and ground-truth trajectory points) and **final displacement error** (Euclidean distance between final estimated and ground-truth points):

$$L(\hat{s}, s) = (\hat{s} - s)^2 = (1/m) * \sum \|\Delta s^i\|_2^2 \, ; \Delta s^i = \begin{pmatrix} x^i \\ y^i \end{pmatrix}$$

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Discussion

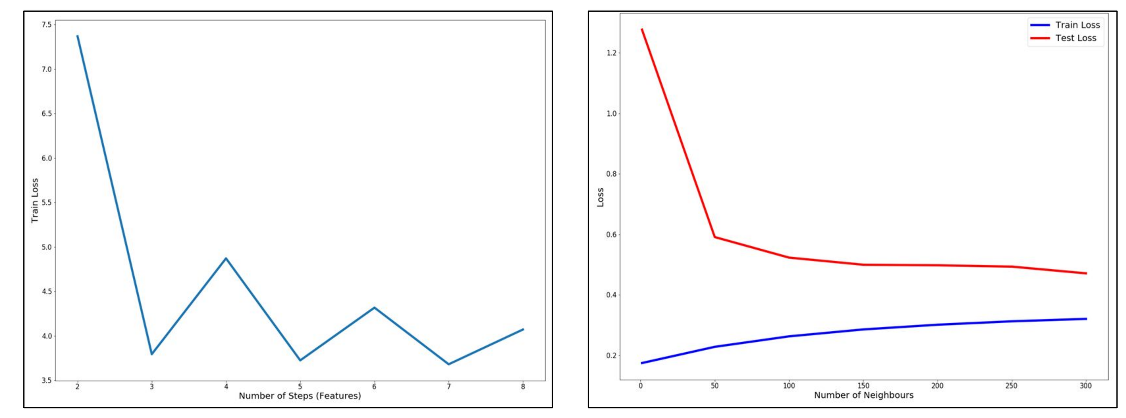
- KNN regression and the deep learning methods far outperformed the baseline, linear regression, for predicting the next 1-2 steps. Intuitively, the trajectories are non-linear and complex, so the poor performance of linear regression was expected.
- Though the GRU and LSTM networks predicted trajectories for at least 2 steps, they outperformed linear regression and KNN regression significantly. GRU's final displacement error was 0.0863, an order of magnitude better than KNN regression.

Future Works

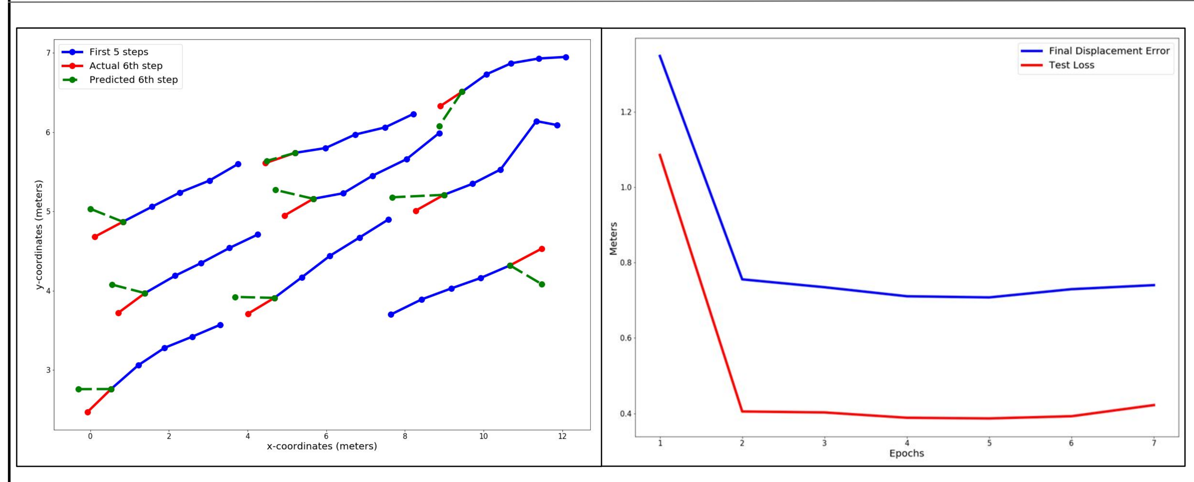
- Incorporate additional parameters to estimate pedestrian intention more accurately, such as gaze direction.
- Capture interaction between neighboring pedestrians to simulate social interaction, allowing much more accurate prediction in highly crowded scenarios.
- Collecting additional data for different crowd scenarios – beyond the given datasets – could enhance the ability for the models to generalize their predictions to other complex pedestrian cases.

Results

KNN regression, predicting 1 step after observing 5 steps.

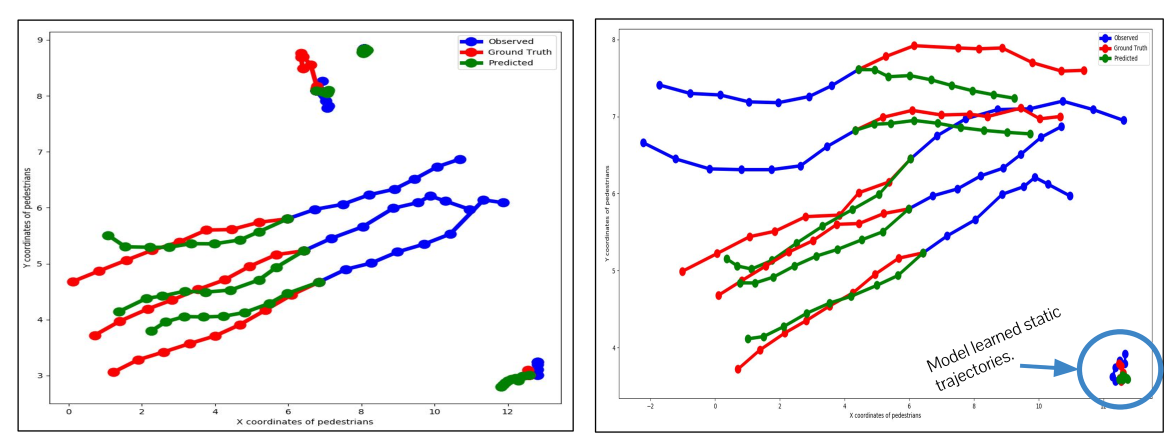


Train loss vs. number of steps (features). *Loss vs number of neighbours.*

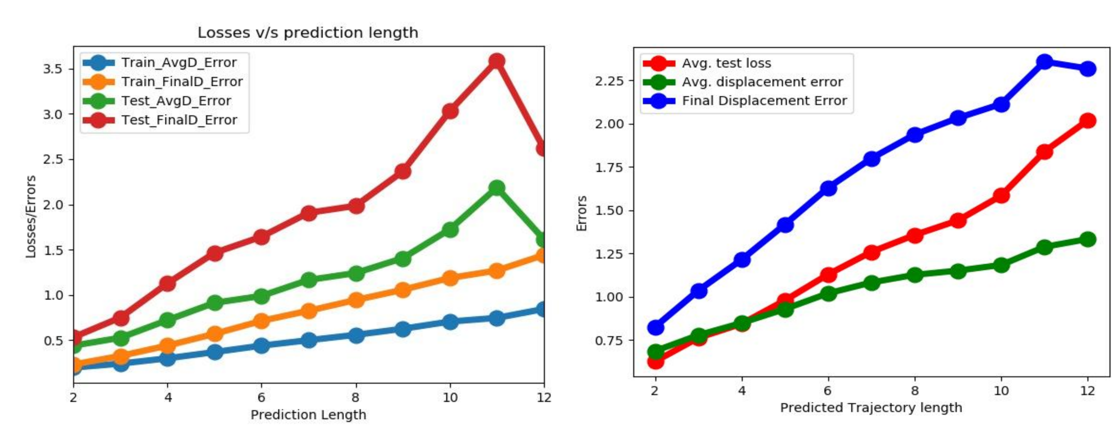


Predicted and actual path of pedestrians. *Displacement error vs. epochs.*

LSTM (*left*) and GRU (*right*) networks, predicting 8 steps after observing 8 steps on ETH.



Errors grow significantly with an increase in predicted trajectory length, especially past 8 steps.



Machine learning algorithms	Number of steps predicted	Test loss (MSE)	Final test displacement error (meters)	Average test displacement error (meters)
		ETH dataset		
Linear Regression	1	8.7331	3.4498	--
KNN Regression	1	0.4182	0.7319	--
Long Short-Term Memory (LSTM)	2	0.0821	0.3591	0.2902
	8	2.5376	2.4367	1.5625
Gated Recurrent Units (GRUs)	2	0.0049	0.0863	0.0689
	8	0.1278	0.6764	0.3670

References

[1]. Alahi, A., et al.: Social LSTM: Human trajectory prediction in crowded spaces. In: CVPR. (2016)
[2]. A. Lerner, Y. Chrysanthou, and D. Lischinski. Crowds by example. Computer Graphics Forum (Proceedings of Eurographics), 26(3), 2007.
[3]. Gupta, A., et al.: Social GAN: Socially acceptable trajectories with generative adversarial networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018).
[4]. S. Pellegrini, A. Ess, K. Schindler, L. Van Gool, "You'll Never Walk Alone: Modeling Social Behavior for Multi-Target Tracking", Proc. IEEE Int'l Conf. Computer Vision, 2009.
[5]. Olah, C. Understanding LSTM Networks.