Classifying Pedestrian Actions In Advance Using Predicted Video of Urban Driving Scenes

Pratik Gujjar and Richard Vaughan[1]

(a) (b) (c)

Fig. 1: Generating predictions of a future for a pedestrian attempting to cross the street. We pick out two key frames from the (a) input sequence and the (b) ground truth sequence, 16 frames apart. Image (c) shows our prediction at the same time instant as the ground truth.

***Abstract*— We explore prediction of urban pedestrian actions by generating a video future of the traffic scene, and show promising results in classifying pedestrian behaviour before it is observed. We compare several encoder-decoder network models that predict 16 frames (400-600 milliseconds of video) from the preceding 16 frames. Our main contribution is a method for learning a sequence of representations to iteratively transform features learnt from the input to the future. Then we use a binary action classifier network for determining a pedestrian's crossing intent from predicted video. Our results show an average precision of 81%, significantly higher than previous methods. The model with the best classification performance runs for 117 ms on commodity GPU, giving an effective look-ahead of 416 ms.**

I. Introduction

Automated vehicles must react very quickly to the actions of pedestrians for safety. For maximum responsiveness, we would like to predict dangerous pedestrian behaviours and react to them *before* they are observed. In this work we predict behaviours of pedestrians seen from a car dashboard video while crossing, in a variety of street configurations and weather conditions. Achieving this robustly would have obvious applications in autonomous vehicles. We investigate two data-driven methods to learn traffic activity as a scene phenomenon: an autoregressive model of motion of traffic participants for video prediction, and an action recognition algorithm that detects crossing intent in pedestrians.

Most pedestrian scenarios explored in existing literature are concerned with pedestrians who are approaching the street orthogonally and in constrained settings. However, many other configurations occur in real world scenes. A traffic scene is an interplay between various agents besides pedestrians. Cars driving on and off an intersection, the nature of the intersection (stop controlled or not), and the speed and direction of other pedestrians on the crosswalk are among many factors that influence a pedestrian's decision to cross a road. Here we manage the complexity of the full multi-agent situation and its context by modelling it as a scene phenomenon from the point of view of the vehicle camera. From a history of 16 video frames, we predict the evolution of scene by predicting the video into the future. We propose and compare alternative neural network models for predicting 16-frame video futures of driving clips: a fully convolutional net, a decoding strategy employing recurrent nets and a network that encourages iterative transformation and refinement of learnt representations using residual connections. We compare these models for their qualitative and quantitative strengths.

The main novelty and contribution of this work is the use of a sequence of learnt representations from which we can *decode* a future, given past video, and thus potentially take robot control actions. Our inventive supposition is that using the video scene to summarize the situation can contain useful context that for e.g. a list of bounding volumes with kinematic forward models would not. Below we demonstrate empirically that we can correctly detect changes in pedestrian behaviour in the predicted video.

II. Related Work

In a survey, Ohn-Bar and Trivedi highlight the roles of humans in intelligent vehicle interactions [1]. Human driver factors such as gestures, distraction, and gaze analysis have been widely studied [2]–[7]. Direct observation of pedestrians and occupants of nearby vehicles is less well studied, though this is an important area of current interest [1].

We are concerned with modeling pedestrian intent. Hidden Markov Model (HMM) based approaches [8, 9], in which the hidden state is a pedestrian's intent, have been extended to partially-observable Markov Decision Processes (POMDP) to learn a distribution over pedestrian intent [10]. Although a Markovian process allows the model to quickly adapt to changes in its observations, the Markov assumption itself can be overly restrictive, owing to insufficient prior conditioning.

[1]School of Computing Science, Simon Fraser University pgujjar@sfu.ca

Preprint submitted to 2019 International Conference on Robotics and Automation (ICRA). Received September 15, 2018.

Other approaches for time-series prediction are to assume the samples come from a linear process driven by a white, zero-mean, Gaussian input [11]. Schneider and Gavrilla [12] provide a comparative study of recursive Bayesian filters for pedestrian path prediction. Gaussian processes are slower to detect a change than Markov models because the entire observed trajectory is used to predict the future state [13]. Additionally, they are too slow to learn previously unobserved behaviour patterns. Switching linear dynamical models as extensions to these models are shown to work in constrained environments [12, 14]. Karasev et al. [11] account for more accurate long-term predictions by postulating goals of a pedestrian's trajectory, that they navigate to, by approximately following stochastic shortest paths. Kooij et al. [14] incorporate a Dynamic Bayesian Network (DBN) to supplement a Switching Linear Dynamical System (SLDS) with environment parameters like a pedestrian's head orientation, their situational awareness and distance between the car and the pedestrians to predict their paths.

All these motion models assume accurate segmentation and tracking of pedestrians in the scene, yet this is challenging due to the difficulty of choosing reliable and efficient image features for tracking [15]. The solution is to enable pedestrian activity models to work in the image space by directly analysing videos from a car's dashboard camera. Hasan et al. [16] treat the prediction of adverse pedestrian actions as an anomaly detection problem. They first track handcrafted HOG features to create a spatio-temporal appearance feature descriptor, then earn temporal regularity across video frames. Existing generative architectures struggle with the dimensionality of the input image space and are in many instances not scalable to natural images. Consequently, they suffer from short prediction time horizons. Vondrick et al. [17] use a Generative Adversarial Network to predict future frames using only one frame as the conditioning frame. Lotter et al. [18] use a weighted mean-squared-error and adversarial loss to predict future frames in the bouncing balls dataset [19]. However, extension of this work to natural image sequences is not presented. Lotter et al., subsequently study the future prediction problem with the idea that each layer of prediction only accounts for local changes and forward deviations higher up [20]. Jastrzebski et al. [21] provide observational and analytical evidence that residual connections encourage iterative inference. We incorporate a similar iterative strategy by using residual connections across different scales of frame generation.

III. Future Generation

Our objective is to predict the future positions of salient objects like vehicles and pedestrians by learning their motion. Functionally, an encoder reads a sequence of frames $\boldsymbol{x} = \{x_T, ..., x_1\}$ to yield dense representations $\boldsymbol{z} = \{z_1, ..., z_T\}$. Conditioned on $\boldsymbol{z}$, a decoder will then auto-regressively predict an image sequence $\boldsymbol{y'} = \{y'_{T+1}, ..., y'_{2T}\}$ by minimizing a pixel-wise loss between $\boldsymbol{y'}$ and ground truth frames $\boldsymbol{y} = \{y_{T+1}, ..., y_{2T}\}$. Each generated frame is of the same resolution as the input. We reverse the temporal ordering of input data to condition the latent space with spatial information from the latest frame. The most recent frame carries forward the closest contextual resemblance. Recursively learning representations from each input frame, we expect to first learn a *temporal regularity* in the early representations and parametrize a *temporal variance* in the later representations. We later visualize the learned representations to validate our intuition.

A. Encoder

The encoder is a spatio-temporal neural network composed of three-dimensional convolutional layers. In contrast to the results of Tran et al. [22] we found that in our case, kernels with decreasing sizes in the spatial dimension (11x11 $\rightarrow$ 5x5 $\rightarrow$ 3x3) and constant size in the time dimension capture the input scene and temporal variations in greater detail. A large spatial kernel (11x11) along with a stride with much overlap between them (4) in the first layer was observed to produce sharper images. We believe that this allows the network to account for more spatial features per time frame. Residual connections are introduced at two image resolutions in our downsampling pipeline: 32×54 and 16×26. Two 3D convolutional layers feed into these residual blocks, where the filters are time dilated for larger temporal reception. We forward the features learnt at the first residual block to the decoder, building another residual connection. Each hidden representation z_j, $j \in \{1, ..., T\}$ is a function of all input frames, with the learnt weights determining the contribution of each frame towards each z_j. The learnt $\boldsymbol{z}$ is 16x26 dimensional. We abstract the mathematical formulation for 3D convolutions in Equation (1) to show the temporal order of processing. The equations presented are for an l-th residual block in the encoder with two 3D convolutional layers, a and b in each block. k is the kernel size in the time dimension and the equations do not show dilations in the kernel. We use time distributed 1×1, 2D convolution operations for dimensionality matching in addition operations for residual connections.

$$\begin{aligned} h_t^{al} &= f_a(\boldsymbol{W}_a(r_{t:t-k}^{l-1}) + \boldsymbol{b}_a) \\ h_t^{bl} &= f_a(\boldsymbol{W}_b(h_{t:t-k}^{al}) + \boldsymbol{b}_b) \\ r_t^l &= r_t^{l-1} + h_t^{bl} \end{aligned} \tag{1}$$

B. Decoder

The decoder is recurrent, containing convolutional LSTM layers [23]. ConvLSTM layers interspersed with up-sampling layers go from the low-dimensional representation space of $\boldsymbol{z}$ to the image space of $\boldsymbol{y'}$. Unlike the encoder, the decoder layers up-sample steadily to facilitate fluid transforms. We found that a fixed kernel size of 3×3 provided an appropriate balance between training time and quality of generation.

We introduce residual connections at three image scales of the upsampling pipeline: ($16 \times 26, 32 \times 52$ and 64×104), following the intuition that each block would optimize for mutually different visual features. We introduce another residual connection to factor in the first convolutional level image features forwarded from the encoder. This is added

at the 32×52 image resolution, or the second level of decoding. We only add the first feature vector corresponding to the last few input frames for a balance between keyframe retention and over-conditioning of the decoder. We choose a greater number of filters in the earlier stages of the decoder, reducing them rapidly towards the end ($128 \rightarrow 64 \rightarrow 16 \rightarrow 3$) to generate a 3-channel colour image. Our interpretation is that the greater number of filters early on offers more opportunities for a structural transformation due to smaller image resolutions. The reduced number of filters in the later stages eases the network's optimizing efforts. The final transformations are encouraged to be more refining than compositional because of iterative refinement in residual networks [21]. Each decoding layer's function can be elementarily defined as in Equation (2). We abstract the hidden state dependences from less relevant convolution operations in the recurrence formulation of the ConvLSTM.

$$\begin{aligned} h_t^{al} &= f_a(\boldsymbol{W}_a(r_t^{l-1})^\dagger + \boldsymbol{U}_a h_{t-1}^{al} + \boldsymbol{b}_a) \\ h_t^{bl} &= f_b(\boldsymbol{W}_b(h_t^{al})^\dagger + \boldsymbol{U}_b h_{t-1}^{bl} + \boldsymbol{b}_b) \\ r_t^l &= r_t^{l-1} + h_t^{bl} \\ (r_t^l)^\dagger &= Upsamp(r_t^l) \\ r_t^0 &= z_t \end{aligned} \tag{2}$$

IV. Action Recognition

The task of action recognition is motivated by the idea that by looking ahead in time, we could react to a hazardous pedestrian interaction a little earlier, with safety benefits. We do this end-to-end by appending a binary action classifier to our future video generator. In this task, we want to learn to predict a pedestrian's crossing intent across a multitude of crossing scenarios and behaviours. Prediction accuracy also serves as an evaluation metric to compare the quality of videos generated by various models. Formally, a classifier parametrized by θ, predicts the probability of a crossing event $P(a)$ in a scene as a sigmoidal function of the generated frames $\boldsymbol{y}' = \{y'_{T+1}, ..., y'_{2T}\}$ as shown in Equation (3).

$$P(a|\boldsymbol{y'}_{T+1:2T}) = \sigma\{f_\theta(\boldsymbol{y'}_{T+1:2T})\} \tag{3}$$

As most occurrences of the beginning and end of a crossing event take place towards either the left or right edges of the frame (usually the curbsides), we slice our input frames into two with an overlapping region between them (Column slices $0:112$ and $96:208$). Two instantiations of the same classifier network are used to extract motion and visual features from these image slices. Features from the C3D pair are concatenated before transforming into the probability of crossing through two newly trained fully connected layers. The last layer uses a sigmoidal activation, with the classifier trained on binary-crossentropy loss. Performance scores and training strategy are detailed in section V-F.

V. Experiments

A. Experimental Setup

We use the JAAD dataset [24] consisting of 346 high resolution videos in pedestrian interaction scenarios. The clips were collected from approximately 240 hours of driving videos recorded at 30 fps during different times of day and under varying lighting conditions. We split 30% of the data into our test set and 10% as validation set. We omit 8 videos sampled at a higher frame rate of 60 fps, leaving 60% of the data as the training set. The recordings across the sets are mutually exclusive, meaning we do not split the same video across any of the sets. To augment the training set, we stride a window of 32 frames by one, over the videos and pack them in randomly shuffled batches. We train the encoder-decoder stack described in the previous sections to optimize for a combination of l_1 and l_2 losses as shown in Equation (4). The losses are calculated between the N pixels of T predicted frames y' and ground truth frames y. For video prediction experiments we set $N = 128 \times 208$ and $T = 16$ frames. The generator is first trained with the *RMSprop* optimizer for 30 epochs to minimize l_2 loss. Learning rate is set at 10^{-3} for 7 epochs and then reduced by a tenth after 14 epochs and again after 20 epochs. We then train the network for 10 more epochs to minimize l_1 loss for visually smoother images and sharper edges with the learning rate set at 10^{-5}. In order to visualize relative consistency across frames in the generated sequence, we plot l_1 loss between the prediction and the ground truth per frame in a *Temporal Variation Graph* (TVG) as a function of time. We experiment with various architectures for the 16-frame generator which we evaluate qualitatively and quantitatively in the following sections. All training and experiments are run on Nvidia GTX 1080Ti GPU. We arrived at these designs and procedures after extensive iterated experiments. For lack of space we describe only the most successful designs.

$$L = \frac{1}{N}\sum_{t=T+1}^{2T}\sum_{i=1}^{N}(y_{t,i} - y'_{t,i})^2 + \lambda\frac{1}{N}\sum_{t=T+1}^{2T}\sum_{i=1}^{N}|y_{t,i} - y'_{t,i}| \tag{4}$$

B. Frame-wise Accuracy: Qualitative Analysis

We train three kinds of models for future prediction: a fully convolutional model (*Conv3D*), a recurrent decoder model (*Segment*) and a residual encoder-decoder model (*Res-EnDec*). We then perform ablation studies on the residual encoder-decoder model described earlier. Figures 1, 4 and 5 present examples with action labels inscribed within. Figure 4 shows image sequences predicted by various models and are stacked one below the other for comparison. We only show some key-frames for brevity. See accompanying predicted videos.[1]

C. Frame-wise Accuracy: Quantitative Analysis

Table I lists some frame generation metrics for various models. The average pixel-wise prediction l_1 error is $(1.37 \pm 0.37) \times 10^{-1}$ for the Res-EnDec model. The errorbar l_1 loss TVG for this model, shown in Figure 2a displays an increase in prediction error, as the errors can be expected to accumulate as one goes deeper in time. We suspect the

[1] http://autonomy.cs.sfu.ca/deep_intent/

Fig. 4: A group of pedestrians are about to cross the street from right to left. Row 1: Input frames - pedestrians are not yet crossing; Row 2: Ground-truth future frames for reference - a pedestrian steps into the street at Column 3; Row 3: Res-EnDec model predicted frames; Row 4: Segment model predicted frames; Row 5: Conv3D model predicted frames. Frames are labelled with the classified action. All models except the Segment model predict the action change correctly.

Fig. 5: Row 1: Ground Truth. Rows 2: Future predicted by Res-EnDec model; A pedestrian can be seen to cross the intersection from behind the car ahead from the ground truth sequence. All models are unable to capture this movement. We believe that the car's motion dominates the transformation for objects in the scene, hindering the pedestrian's forward movement prediction.

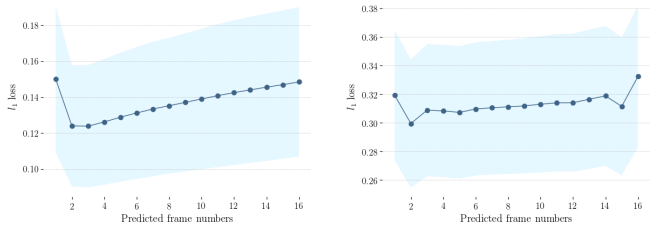

Fig. 2: TVG for Res-EnDec model (left) and Conv3D model (right)

high initial error in the TVG to be caused by the substantial influence the last input frame has on the generation due to our reverse ordering of the input. An observed drop in the prediction loss from the second frame onwards empirically suggests that the model appears to configure the spatiality of the scene first and then successively transform objects to project motion.

In comparison, the TVG for the *Conv3D* model shows a much higher mean prediction error of $(3.13 \pm 0.47) \times 10^{-1}$. The decoder in the Conv3D model uses transposed convolutions or deconvolutions, to increase image sizes [25]. All kernel sizes are set to $3 \times 3 \times 3$ as is common with

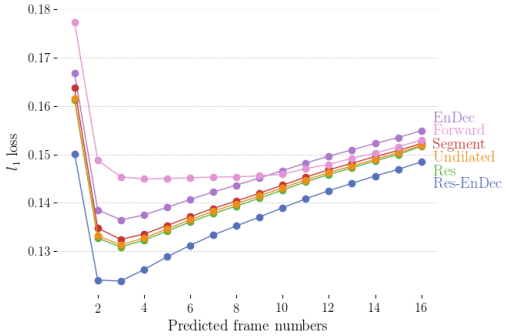

Fig. 3: Temporal Variation of the l_1 prediction loss for various models. *Conv3D* model omitted due to large values.

Model	l_1 loss ($\times 10^{-1}$)	Appreciation (%)
Conv3D	$\mathbf{3.13 \pm 0.47}$	**7.61**
Segment	1.43 ± 0.37	14.94
EnDec	1.46 ± 0.37	13.48
Res	1.42 ± 0.36	15.88
Res-EnDec	$\mathbf{1.37 \pm 0.37}$	**19.86**
Undilated	1.42 ± 0.36	15.60
Unreversed	$\mathbf{1.49 \pm 0.39}$	**5.22**

TABLE I: Analysis of loss variation in time for various models

3D convolutional architectures [22, 26, 27]. The trend over time, as can be noted from Figure 2b is unstable and the error does not appreciate as much as in the case of the Res-EnDec model. We believe the instability arises due to our reverse ordering of input frames on an architecture that is more parallel than the others with recurrent decoders.

The sequential nature of the video frames can be benefit from recurrent layers over 3D-convolutions. We change the decoder from the Conv3D model to adopt recurrent convolutional layers to build our *Segment* model. The convolutional layers help preserve the spatiality of the data relative to standard vector LSTM layers. The Segment model is designed with kernel sizes and number of filters drawn from the common image segmentation models [28]. The mean error for this model is $(1.43 \pm 0.36) \times 10^{-1}$, $\sim 54\%$ better than the Conv3D model and $\sim 4\%$ worse than the Res-EnDec model.

D. Ablation Studies

We perform ablation studies on our Res-EnDec model to determine the importance of the residual connections, dilated convolutions and reversal of image data. In our first *Res* model, we remove the residual connection between the encoder and decoder, but let the connections within the sub-modules remain. Next we remove all residual connections in the *EnDec* model. In the *Undilated* model, all convolutions are undilated and we retain the residual connections within the encoder-decoder pair. In our last ablation study, we study the effect of reversing the input data. In the *Unreversed* model, the input data ordering is not reversed. The graph in Figure 3 compares the temporal error variance across all models. We do not include the TVG for the Conv3D model so as to not skew the $y-$axis with its relatively large prediction errors.

Prediction error can be seen to appreciate across all models over time. These results suggest that both dilated convolutions and residual connections have improved loss performance. We omit discussion for space.

E. Latent Space Visualization

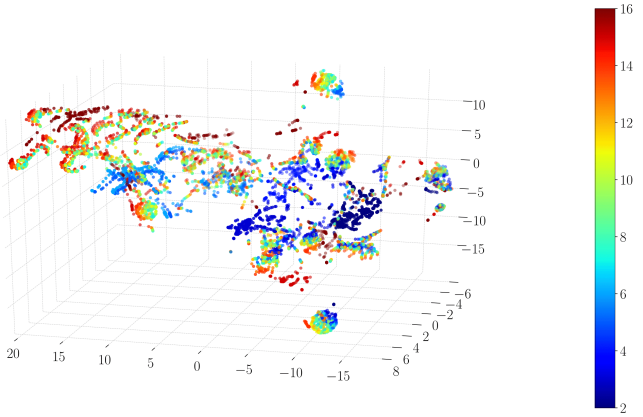

Fig. 6: Scatter plot of t-sne minimized representations learnt by the encoder. A gradual trend from deep blue to red suggests that the representations are different from each other. Autoregressively processing them encourages iterative transformation.

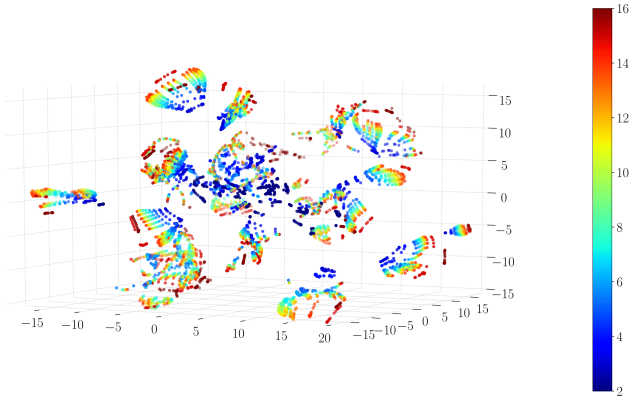

Fig. 7: Scatter plot of t-sne minimized tensors obtained from the first residual connection in the decoder. Close association of samples in the form of blue-red streaks suggests that the recurrent convolutions assist in reforming the representation space into a hypersphere with motion projecting the transformations outwards.

We visualize the $16 \times 26 \times 64$ dimensional representation space z using the t-distributed Stochastic Neighbour Embedding algorithm (t-SNE) [29]. Visualizing each z_i, $1 \leq i \leq 16$ directly is inconclusive because they correspond to a variety of traffic scenes. Subtracting the first representation from each of the predictions to retain temporal variances borne out of object motion, we project them into 3D in Figure 6. We observe almost no visible motion across predicted video when we replace learned representations $z_{1:16}$ with the first representation z_1. Thus we choose the first frame for subtraction because we deem it to transfer the most spatial information. The embedding graph is a scatter plot of 625 representations for each of the 16 temporal slices. The slices are shown with each colour corresponding to a slice of prediction time. The scatter plot shows a trend

Preprint submitted to 2019 International Conference on Robotics and Automation (ICRA). Received September 15, 2018.

going from deep blue to red from the second predicted frame to the last. The clustering of like colours indicates a similarity in decoding a given time slice of prediction. Note representations appear to transition sequentially to a later point in time, suggesting that each representation iteratively adds information to the generation process.

We perform the same experiment as previously discussed but now extract new representations r_i, $1 \leq i \leq 16$ from the first residual connection in the decoder. The scatter plot for t-SNE minimized $16 \times 26 \times 64$ dimensional space is presented in Figure 7. The plot shows streaks of colour transitions flowing from r_1 to r_{16}. The representation space is rendered with more structure, with the r's earlier in the sequence appearing to spawn from within a virtual volume outwards as seen in the 3D rendition. The effect of the recurrent and residual connections together is that of an Iterative Refinement.

F. Crossing Intent

We fine-tune the pre-trained C3D network presented by Tran et. al., [22] on the JAAD dataset of pedestrian behaviours. The C3D model is pre-trained on the Sports 1M dataset [30]. Our model is trained on the same data training, validation and test splits of the JAAD Dataset as described in [31] for comparison. We organise training videos in strides of 8 and test videos in strides of 16. We augment training videos by choosing every alternating frame in the sequence. This also helps simulate faster moving traffic. We use *RMSprop* as the optimizer starting with an initial learning rate of 10^{-5} for 30 epochs with early stopping based on validation loss. The learning rate is reduced by a factor of 10 at epochs 7 and 16. We also regularize the fully connected layers to lessen overfitting.

Model	Acc	Prec	Recall	F_1	Time (ms)
Conv3D	61.81	**83.75**	46.74	60.00	68.84 $\pm$ 26.34
Segment	65.23	81.80	56.83	67.07	96.10 $\pm$ 28.89
Res	66.59	79.23	62.84	70.09	116.11 $\pm$ 42.22
Res-EnDec	**67.38**	74.77	**71.90**	**73.31**	116.39 $\pm$ 38.78

TABLE II: Crossing intent classification performance

Model	AP
Action [31]	39.24 $\pm$ 16.23
Action + Context [31]	$\mathbf{62.73 \pm 13.16}$
Conv3D	78.61
Segment	80.85
Res	80.42
Res-EnDec	**81.14**

TABLE III: Average precision in predicting crossing intent

The classifier is input with futures predicted by the models listed in Table II. The table compares accuracy in recognizing a crossing action from predicted videos alone. The test set contains 1257 image sequences of 16 frames each, of which 474 are labelled *crossing* and 783 as *not crossing*. The Res-EnDec model with the lowest reconstruction loss outperforms the Conv3D model by around 6%. We compare average precision scores with the results presented by Rasouli et al. in [31], in Table III. We outperform their Action+Context model consistently across all our models and by about 18% in the case of our best result.

VI. Discussion

Predicting adverse pedestrian crossing actions has the potential to save lives. Conversely, it would be undesirable for an autonomous driving system to brake every time a pedestrian is detected on the curbside. Such a system could tolerate a non-crossing action mislabelled as crossing or false positives, but false negatives need to be penalized. From Table II, we find that the Res-EnDec model has similar precision and recall scores on our test set. On the other hand, the Conv3D model has a precision $\sim$10% higher than the Res-EnDec, but has the lowest recall. The F_1 scores better summarize this trend.

Predicting 16 frames of future at a frame rate of 30 fps corresponds to looking ahead 533 ms in time. Any advantage gained from this is reduced by the run time of the prediction method. The last column in table II lists the run time to load a 16-length image, predict the next 16 frames and classify each predicted frame as an action for four of our test models on our implementation running on an Nvidia GTX 1080 Ti GPU. Conv3D is the fastest model at $69ms$, for an effective maximum look-ahead time of 463 ms.

VII. Conclusion

In this paper, we proposed and demonstrated three broad categories of neural network algorithms tasked with generating video predictions of the future. We then introduced a Temporal Variation Graph for all models, to measure their contributions in per-frame visual reproducibility and a temporal coherence. Our results suggest that the residual connections encourage learnt intermediate representations to be mutually different. Along with multi-stage recurrent decoding, iterative refinement can be seen. The novelty of our approach is that we learn a sequence of representations from an encoder rather than a comprehensive vector as done in many sequence generation approaches.

We also proposed and demonstrated a classifier algorithm based on the C3D action classifier model. The network was tasked with recognizing a crossing action by looking at a video of a predicted future, thereby being able to predict a pedestrian's crossing intent. We showed that our best model with an average precision of 81.14% is 18% higher than the model introduced by Rasouli et al. [31] for videos from the JAAD Dataset. This demonstrates the contribution of context in the overall gain in performance without explicitly having to detect scene elements such as traffic signs to interpret as context. Our best performing model predicts a future and a crossing intent with an effective look-ahead of $416ms$.

We propose this system and results as a proof-of-concept that predicting the future via video could potentially provide useful early input to action selection for mobile robots including safety critical systems like urban driving.

VIII. Acknowledgements

Supported by the NSERC Canadian Field Robotics Network.

REFERENCES

[1] E. Ohn-Bar and M. M. Trivedi, "Looking at Humans in the Age of Self-Driving and Highly Automated Vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 90–104, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7501845/

[2] K. Seshadri, F. Juefei-Xu, D. K. Pal, M. Savvides, and C. P. Thor, "Driver cell phone usage detection on strategic highway research program (shrp2) face view videos," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2015, pp. 35–43.

[3] E. Ohn-Bar, S. Martin, A. Tawari, and M. M. Trivedi, "Head, eye, and hand patterns for driver activity recognition," in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 660–665.

[4] F. Parada-Loira, E. Gonzalez-Agulla, and J. L. Alba-Castro, "Hand gestures to control infotainment equipment in cars," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 1–6.

[5] A. Rangesh, E. Ohn-Bar, and M. M. Trivedi, "Hidden hands: Tracking hands with an occlusion aware tracker," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2016, pp. 1224–1231.

[6] A. Tawari, K. H. Chen, and M. M. Trivedi, "Where is the driver looking: Analysis of head, eye and iris for robust gaze zone estimation," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Oct 2014, pp. 988–994.

[7] C. Tran, A. Doshi, and M. M. Trivedi, "Modeling and prediction of driver behavior by foot gesture analysis," *Computer Vision and Image Understanding*, vol. 116, pp. 435–445, 2012.

[8] R. Kelley, M. Nicolescu, A. Tavakkoli, M. Nicolescu, C. King, and G. Bebis, "Understanding human intentions via hidden markov models in autonomous mobile robots," in *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2008, pp. 367–374.

[9] Q. Zhu, "Hidden markov model for dynamic obstacle avoidance of mobile robot navigation," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 390–397, Jun 1991.

[10] T. Bandyopadhyay, C. Z. Jie, D. Hsu, M. H. Ang, D. Rus, and E. Frazzoli, *Intention-Aware Pedestrian Avoidance*. Heidelberg: Springer International Publishing, 2013, pp. 963–977. [Online]. Available: https://doi.org/10.1007/978-3-319-00065-7_64

[11] V. Karasev, A. Ayvaci, B. Heisele, and S. Soatto, "Intent-aware long-term prediction of pedestrian motion," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2543–2549.

[12] N. Schneider and D. M. Gavrila, "Pedestrian Path Prediction with Recursive Bayesian Filters: A Comparative Study," in *Pattern Recognition*, J. Weickert, M. Hein, and B. Schiele, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 174–183.

[13] D. Ellis, E. Sommerlade, and I. Reid, "Modelling pedestrian trajectory patterns with gaussian processes," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, Sept 2009, pp. 1229–1234.

[14] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8694 LNCS, no. PART 6, pp. 618–633, 2014.

[15] B. Vlz, K. Behrendt, H. Mielenz, I. Gilitschenski, R. Siegwart, and J. Nieto, "A data-driven approach for pedestrian intention estimation," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 2607–2612.

[16] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," *CoRR*, vol. abs/1604.04574, 2016. [Online]. Available: http://arxiv.org/abs/1604.04574

[17] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," *CoRR*, vol. abs/1609.02612, 2016. [Online]. Available: http://arxiv.org/abs/1609.02612

[18] W. Lotter, G. Kreiman, and D. D. Cox, "Unsupervised learning of visual structure using predictive generative networks," *CoRR*, vol. abs/1511.06380, 2015. [Online]. Available: http://arxiv.org/abs/1511.06380

[19] I. Sutskever, G. E. Hinton, and G. W. Taylor, "The recurrent temporal restricted boltzmann machine," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 1601–1608. [Online]. Available: http://papers.nips.cc/paper/3567-the-recurrent-temporal-restricted-boltzmann-machine.pdf

[20] W. Lotter, G. Kreiman, and D. D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," *CoRR*, vol. abs/1605.08104, 2016. [Online]. Available: http://arxiv.org/abs/1605.08104

[21] S. Jastrzebski, D. Arpit, N. Ballas, V. Verma, T. Che, and Y. Bengio, "Residual connections encourage iterative inference," *CoRR*, vol. abs/1710.04773, 2017. [Online]. Available: http://arxiv.org/abs/1710.04773

[22] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: generic features for video analysis," *CoRR*, vol. abs/1412.0767, 2014. [Online]. Available: http://arxiv.org/abs/1412.0767

[23] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *CoRR*, vol. abs/1506.04214, 2015. [Online]. Available: http://arxiv.org/abs/1506.04214

[24] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Joint attention in autonomous driving (JAAD)," *CoRR*, vol. abs/1609.04741, 2016. [Online]. Available: http://arxiv.org/abs/1609.04741

[25] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 2528–2535.

[26] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1933–1941, 2016.

[27] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 1510–1517, 2018.

[28] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, 2017.

[29] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," 2008.

[30] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 1725–1732. [Online]. Available: https://doi.org/10.1109/CVPR.2014.223

[31] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Oct 2017, pp. 206–213.