Reasoning About Pedestrian Intent by Future Video Prediction

by

Pratik Gujjar

B.Tech, National Institute of Technology
Karnataka, India, 2015

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Computing Science
Faculty of Applied Science

SIMON FRASER UNIVERSITY
Fall 2018

Approval

Name:	**Pratik Gujjar**
Degree:	**Master of Science (Computing Science)**
Title:	**Reasoning About Pedestrian Intent by Future Video Prediction**
Examining Committee:	**Chair:** Dr. Mo Chen Professor
	Dr. Richard Vaughan Senior Supervisor Professor
	Dr. Anoop Sarkar Supervisor Professor
	Dr. Yasutaka Furukawa Examiner Professor
Date Defended:	**November 20, 2018**

Abstract

Automated vehicles must react very quickly to pedestrians for safety. We explore analysis and prediction of pedestrian movements around urban roads by generating a video future of the traffic scene, and show promising results in classifying pedestrian behaviour before it is observed. Our first method consists of a decoding algorithm that is autoregressive of representations that an encoder learns from input video. We compare many neural network based encoder-decoder models to predict 16 frames (400-600 milliseconds) of video. We present the contributions of time-dilated causal convolutions and additive residual connections in our recurrent decoding algorithm. Furthermore, we show that these connections encourage representations at various decoding stages to be mutually different. Our main contribution is learning a sequence of representations to iteratively transform features learnt from the input to the future. Our second method presents a binary action classifier network for determining a pedestrian's crossing intent from videos predicted by our first method. Our results show an average precision of 81%, significantly higher than previous methods. Our best model in terms of classification performance has a run time of 117 ms on a commodity GPU with an effective look-ahead of 416 ms.

Keywords: autonomous driving, human-robot interaction, pedestrian prediction, video prediction, spatio-temporal reasoning

To my mum and my dad
For giving me every moment of your lives

Acknowledgements

A friend suggested that I begin my acknowledgements with *Once upon a time.* Given my many tempestuous days and nights, I felt it was only befitting. So here goes!

Once upon a time, a fear that working a corporate job would mean no two days are different urged me to quit and go back to the familiar pens, papers and exams (yep, the exams too). I would fail miserably to not acknowledge the role of my supervisor Dr. Richard Vaughan for what has followed. I thank you for listening to the incessant rate at which I spew my ideas every time I read a paper. I thank you for being hard and not accept anything below quality. This has pushed me to weigh every word I write, think or say. The questions you raised sparked a more structured debate in me to evaluate the claims and ramifications of our work. I would also like to thank Dr. Anoop Sarkar, for the many discussions we had on encoder-decoder architectures. As tedious though it was, I thank you for your insistence on writing the math down for every layer. It made me seek out new Greek alphabets.

My lab life has been no less bratty than my school life. There are no doubts that everyone in the Autonomy Lab has conceded to me. Geoff does try the daily revolution, with Sepehr joining forces a couple times when he's not saving the world. Silence has taken over this rebel faction since they've heard how regular I am at the gym. *Ha ha ha ha* *evil conquering laughs*. I must not forget to thank Jack for listening to the conversations I forced on him. Perhaps it'd do me good to apologize instead. Coming now to Faraz who I care nothing about. He's not a friend. And he's not helped me with anything. We have never discussed my projects at length at the gym, on the whiteboards, by our desks, over calls, when he's on a date and so on. We've definitely not taken a bus ride that will live in history with infamy for him. I do thank Bita for bringing in Darchin. Although I have seen the little kitten two times, I have had dreams protecting him from Faraz's indifference. More like thank me then eh?! Also, let me set the record straight here, I want to absolve myself of the comment that I know you through Faraz. Knowing you directly has brought me greater happiness (and money, cuz, modo) than knowing Faraz and Payam and Rakesh. Together. Speaking of Payam, my first friend at SFU from the orientation. Thank you for pushing me onto the many rsvp's. I owe you many a free food. I thank Rakesh from his profound enquiries of the method, his envious coding ethics and voracious appetite. Rakesh, I will miss you for when you are banished to the GruVi lab for gross treason. For the little time Adam was here, he helped me pick out the best colour gamut for the graphs in this thesis.

I thank Srikanth for his evaluations and checks on my status with the thesis, Akash for his help with sit-down-and-watch-the-graph-debate, Nazanin for pointing me to the C3D pre-trained model and Amir for introducing me to the heavenly Hibye hazefult cream filled little pockets of delight.

I thank Nishant for inciting a hunger towards neural networks with his borderline stalking of community stalwarts. Thank you for being there the many times I have come home disappointed, lost, exasperated and exhausted. I thank Aniket for telling me a rock is a rock when I saw it a mountain. Thank you for always sharing your food and picking up mine. I thank Spoorthi and Ajith for being my family away from home. Lougheed is not so far away! They have heard me cry, be brave, be a coward and broke. I thank Meet for constantly, definitely not to the point of annoyance, asking me for my date of defence. I thank Lavanya for being so jovial and free! I thank Cassandra, my gracious salsa dance partner who made me realize that two left feet can still *tango(?)*. There were days in those dreaded months where dancing was possibly the only good feeling I had. Thank you also for the, *"Once upon a time"*, narrative. Gives me a *unique* opening to say the least.

My parents have taught me that in life it does you good to chase proficiency over success. My dad has a flair for patience and of course the a-jog-in-the-morning-cures-all attitude. My mum is my academic driver. She insisted on me learning to write in cursive, bought me big books of many colours and prided at everything I have ever done. The books only got bigger over time and then suddenly changed into sheets of papers. Throughout this thesis, they have seen some more sides of me. I thank today's advanced communication and transportation strategies that have kept my family at home with me. I often remark to my mum, *"I have known you my whole life!"*, knowing fully well that everyday of my life is but a shadow of theirs.

This thesis is grand covenant of many individuals who did not necessarily agree to participate or perhaps weren't even aware. Their contributions flowed in for they were and are family and friends. A Master's degree could be mine because of all your hard work and dedication. The least I could do is put your name here and thank you all.

And then I lived happily ever after...

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

(a) (b) (c)

Figure 1.1: 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 future sequence, 16 frames apart. Image (c) shows our prediction of the future ground truth frame.

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 review pedestrians who are approaching the street orthogonally and in constrained settings. However, a typical 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 speed and direction of other pedestrians on the crosswalk, are some of the 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 the 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 inference and refinement of learnt representations using residual connections. We compare these models for their qualitative and quantitative strengths. We plot per-frame l_1 losses between predicted frames and the ground truth in a *Temporal Variation Graph* to visualize relative time consistency across the models. We also report loss metrics from ablation studies to analyze incremental improvements afforded by network style changes such as *time-dilated causal convolutions* and *additive residual connections* in Sections 5.1.2 and 5.1.3.

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 e.g. a list of bounding volumes with kinematic forward models would not. In this thesis, we demonstrate empirically that we can correctly detect changes in pedestrian behaviour in the predicted video.

Our second contribution is an algorithm that exploits C3D [6], a 3D convolutional action recognition network for determining a pedestrian's crossing intent by looking solely at predicted videos. Along with measuring standard binary classification metrics such as average precision and recall, we define a new metric called *mean time to correct prediction* to measure the future generator's responsiveness to input cues. We then discuss that for a crossing intent predictor in the real world, it is imperative that such a model possess good recall as false negatives are very expensive. On the other hand, a low false positive rate would mean reduced unnecessary braking and a potentially smoother driving experience for passengers in the car. We also perform a timing analysis of the various models end-to-end; from loading input sequence of frames, to generating a future, to predicting intention of crossing.

Chapter 2

Predicting a Future

2.1 The Problem

Autonomous driving has been the subject of intense research over the last few decades. Although there are cars with limited autonomy, a ubiquitous transit system is still in the making. Full autonomy requires that robot-cars interact with both their immediate environment and humans. Current robot-cars are impeded by their assumptions of rational interactions with their surroundings. A human driver, on the other hand, would infer the intent of other drivers and pedestrians from their behaviours. For example, in the case of a pedestrian crossing a road, a driver may decide to stop at a cross-walk or slow down, consequent to the appearance and apparent urgency of the pedestrian, amongst other non-verbal cues.

The Insurance Corporation of British Columbia in Canada prescribes *See-Think-Do* as an essential driving strategy [2]. Each unit of the strategy can be elaborated as, *See:* scan for hazards and pay attention to other road occupants for potential hazards, *Think:* decide which hazards are dangerous and rank possible evasive actions, *Do:* manoeuvre to keep yourself and others safe. Typically, three-quarters of a second are needed to *see* a hazard and to *decide* to stop. Three-quarters more of a second are needed to actuate the brakes to stop a vehicle [2]. An early prediction of a potentially hazardous action, could add precious time before one decides to act. By modelling pedestrian intent $400 - 500ms$ in the future, a car travelling at 60 $kmph$ gains between $6.67 - 8.34m$ of stopping distance over the suggested $45m$ [7, 8]. This could make a significant difference in reducing accidents and saving human lives.

Currently available autonomous driving datasets cater to the applications of 3D mapping, navigation, and car and pedestrian detection. Since the goal of our work is centred around human behaviour analysis, we opted to learn from the Joint-Attention in Autonomous Driving (JAAD) dataset [9]. The JAAD dataset was created to facilitate behavioural studies of traffic participants. More details of the dataset are presented in Section 2.2. Perhaps one of the most widely used publicly available datasets, KITTI [10] contains

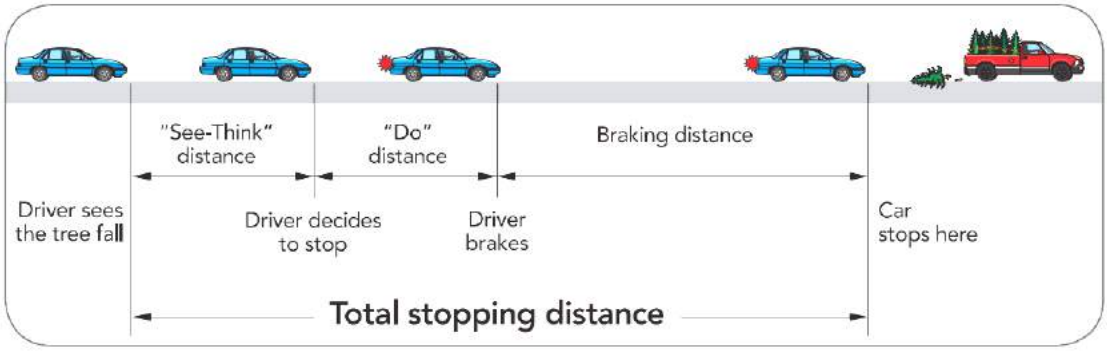
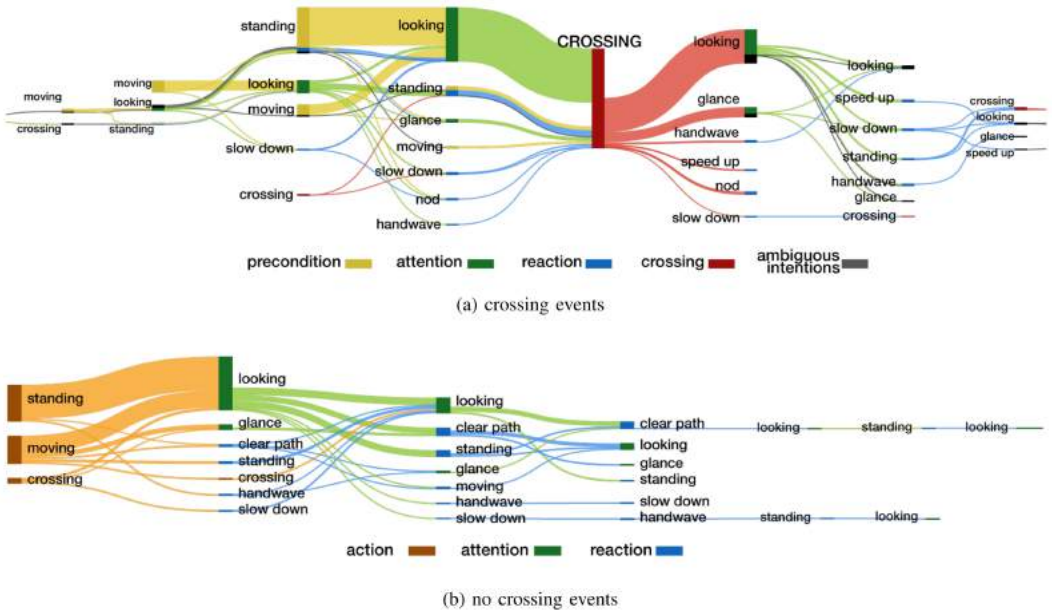


Figure 2.1: Total stopping distance is the distance a vehicle will travel from the moment a hazard is noticed by the driver to the moment the vehicle stops. A driver needs time to see, think and do before the brakes even begin to slow the vehicle. (Picture source: ICBC Learn to Drive Smart Manual [2])

driving data collected from various locations such as residential areas and highways. However, the data has no annotation for pedestrian behaviours. The Caltech pedestrian detection benchmark [11] on the other hand has 10 hours of driving video clips in regular traffic in urban environments. The data contains pedestrian bounding boxes and detailed occlusion labels. The Berkeley pedestrian dataset [12] and the Daimler pedestrian benchmark dataset [13] are both formulated with pedestrian detection, tracking and scene segmentation as intended applications and contain pedestrian bounding box labels. All this data, although rich in pedestrian annotations, does not include any labels for behavioural analysis such as crossing or stopped [9], ignoring the role of context in influencing a pedestrian's intent.

2.1.1 High Variability

Rasouli et al. [3] while presenting their analysis of the JAAD dataset observe high variability in the behaviour of pedestrians. At a point of crossing, they observe more than a 100 distinct patterns of actions. Typically expected crossing behaviour *Standing-Looking-Crossing* and *Crossing-Looking* only account for half the situations observed in the dataset. In one-third of the non-crossing scenarios, the pedestrians are waiting at the curb and looking at the traffic - a strong indication of a possible crossing intention. An individual action performed before and after a crossing event may mean different things. For example, a person may be standing before beginning to cross or as a response to the oncoming vehicle. Figure 2.2 is a visualization of the action transitions for all crossing events in the JAAD dataset. Such a high variability presents a formidable challenge for generalization and hand-crafting features to model behavioural sequences.



(a) crossing events

(b) no crossing events

Figure 2.2: A summary of sequences of pedestrian actions before and after crossing. The thickness of lines represents the frequency of actions in the "crossing" or "non-crossing" subset [3].

2.1.2 Non-Verbal Cues

In more than 90% of the JAAD dataset, pedestrians are observed to use some form of non-verbal communication in their crossing behaviour. An observable attention behaviour is to look at oncoming traffic (Figure 2.3). Other communication forms such as nodding and hand gestures are less frequent. Spatial resolutions of cues such as *looking*, however, are almost inconspicuous as we resize the frames in order to conform to computational limitations. Also, although a pedestrian's head orientation and attentive behaviour are strong indicators of crossing intent, they are not always followed by a crossing event. In designated crosswalks, a pedestrian may cross the street with no attention expressed towards the traffic. In such events, a pedestrian's gait, posture and approach to the curb (a cue as simple as raising a leg to step forward), could lead to more informed inferences about their crossing intent. The cues as observed, are not explicit, but are embedded in the behaviour of the pedestrian in the presence of and in response to traffic movement.

2.1.3 Context Matters

The context in which a crossing event takes place may strongly affect a pedestrian's behaviour. The context can be identified as a set of factors like the weather, nature of the intersections (stop controlled or not), street structure, presence of other vehicles at the intersection and size of the crowd attempting to cross the street. The idea of context also

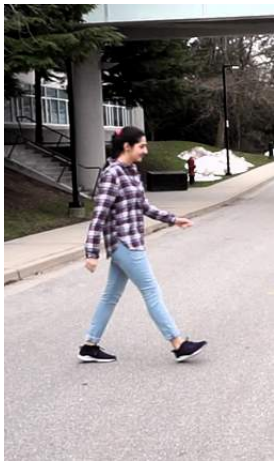
(*a*) (*b*)

Figure 2.3: Some non-verbal cues exhibited by crossing pedestrians. Left: Looking onto oncoming traffic. Right: Leaning forward and preparing to take a step.

extends to target specifics such as demographics (gender and age) or cultural attributes, spatio-temporal variations (velocity and trajectory) or environmental context (traffic signs). As can be seen in Figure 2.4 (a), cropping out the context makes it impossible for a model to predict that the pedestrian will stop before the stationary car seen in Figure 2.4 (b). Pedestrians in the JAAD dataset were observed to cross without attention to the vehicles at non-designated cross-walks when *time-to-collision* (TTC) [14] was greater than 6 seconds. When TTC is below 3 seconds, there is no occurrence of crossing without attention in streets wider than 2 lanes. By adding context to a crossing event classifier, Rasouli et al., [1] achieved more than 20% improvement in prediction performance (more details in Section **??**). Changing the speed of the vehicle leads to varying views of the same intersection on different approaches. Considering the vehicle's own ego-motion is important in further resolving prediction ambiguity.

2.2 JAAD Dataset

The Joint Attention in Autonomous Driving (JAAD) dataset [9] consists of 346 high-resolution video clips (1920 x 1080 and 1280 x 720) created to study the behaviour of traffic participants. The clips were collected from approximately 240 hours of driving videos recorded at 30 FPS (except for a few recorded at 60 FPS) during different times of the day and under varying lighting conditions. Some videos contain a challenging sun glare. The videos collected also cover heavy snow conditions and with people wearing hooded jackets or carrying umbrellas. Two vehicles equipped with wide-angle video cameras were used for data collection. The cameras were mounted inside the cars in the centre of the windshield and below the rear-view mirror. The clips cover a wide spectrum of urban and suburban



(*a*)

(*b*)

Figure 2.4: Top: Without contextual knowledge, the pedestrian is likely to be predicted as continuing to walk across. Bottom: The knowledge of the stationary car adds the information that the pedestrian is very likely to stop before it.

locations and only a few in rural locations. The samples cover a variety of situations like pedestrians crossing individually or as a group, pedestrians occluded by objects, pedestrians pushing objects like a stroller while crossing and pedestrians walking along the street and not crossing. Many crossing instances resemble the commonly seen pedestrians waiting at intersections. The data also contains samples of the usually less recorded situations like a pedestrian walking along the road looks back to see if there is a gap in traffic or peeks from behind an obstacle to see if it is safe to cross. Pedestrian interaction locations range from controlled intersections to parking lots.

All video data is annotated for pedestrian behavioural analysis and is provided with bounding boxes and textual annotations. The bounding boxes are provided for cars and pedestrians that are active in an interaction or require the attention of the driver. The

textual annotations provide behavioural tags for each pedestrian and driver engaged in a traffic correspondence. Each pedestrian is assigned a distinguishing label and tagged within a range of actions: *clear path, crossing, hand-wave, looking, moving fast, moving slow, walking, nod, slow down, speed up, standing* and *stopped*. The data includes 654 unique pedestrian samples with behavioural tags to inform the type and duration of the pedestrian's action.

Pedestrians are annotated such that their behaviours can be categorized into three groups: *Precondition:*- the state of the pedestrian prior to crossing (standing, moving slow or fast), *Attention:* the way the pedestrian becomes aware of the approaching vehicle (looking), *Response:* behaviours the pedestrians respond with to the approaching vehicle (stop, clear path). Although this method of annotation is behaviourally consistent, in our experiments, we found it to be underspecifying. Consider for example the task of recognizing the action being performed in an n-length sequence of frames where the pedestrian might exhibit a mixture of actions. The pedestrian might be standing on the curb and looking at the person standing next to them. The data only provides the standing label as the behavioural annotation. A model that has learnt to recognize both the standing and looking actions to a reasonable accuracy might be penalized in the evaluation for predicting the looking label instead of the standing label. Irrespective of the seemingly erratic labelling, learning the inherent temporal structure in the labels themselves could help adhere to the *precondition-attention-response* strategy of tagging. Another point to note about the provided tags is the skew in the number of labels for each class. The dataset contains about 88K unoccluded pedestrian samples with around 14K samples for people standing and 17K for people looking. Temporally, as the duration of a looking action is shorter than a standing action, standing labels per frame vastly outnumber looking labels per frame. In our work, we identify that the crossing label amongst all other labels, is visually sufficiently well-represented. Also, a crossing pedestrian is the most urgent behaviour to be responded to. For these reasons, we choose to predict a pedestrian's crossing intent and ignore all other labels. We approach the problem as a binary classification of the video sequence into either the crossing or not crossing classes.

2.2.1 Baselines

Rasouli et al. [1] in their follow-up work from 2018, examine approaches using Convolutional Neural Networks (CNNs) to detect and analyze the context of the scenes as well as pedestrians' behavioural cues. In this section, we present the results that are relevant to our work.

They train separate models for pedestrian gait and attention estimation. A randomly initialized AlexNet [15] is trained end-to-end on cropped images of pedestrians from the JAAD dataset with minor occlusions up to 25% allowed in the image crops. They then try transfer learning by fine-tuning AlexNet pre-trained on ImageNet. The networks are also trained with full-body poses to improve classification performance instead of the using the

Method	walking	looking
AlexNet-full	78.34	67.45
AlexNet-cropped	74.23	74.98
AlexNet-Imagenet-full	80.45	75.23
AlexNet-Imagenet-cropped	83.45	80.23

Table 2.1: The average precision (AP%) of classification results for pedestrians' walking and looking actions [1].

Method	AP
Action	39.24 ± 16.23
Action + Context	$\mathbf{62.73 \pm 13.16}$

Table 2.2: Prediction accuracy (%) of pedestrians' crossing. Adding context information significantly improves the prediction results [1].

crops of the head to estimate orientation or lower body to estimate gait. The crops are roughly obtained with the top third of the pedestrian bounding box for attention and the bottom half for gait estimation. Table 2.1 (from their paper) lists the average precision of the classification results for two classes: walking and looking.

In another experiment similar to our work, Rasouli et al. examine the contribution of gait and attention to determine if a pedestrian is going to cross the street. They select between 10-15 frames and corresponding pedestrian bounding boxes where the pedestrians are either walking or standing and looking or not looking. This results in 81 non-crossing and 234 crossing scenarios with a total of 3324 frames. Visual features from a modified variant of AlexNet trained for attention and gait estimation are used as a compact representation to further train a linear SVM to classify the frames as crossing or not-crossing. A second experiment is run to study the importance of context in this task. A fully-convolutional variant of AlexNet is trained to detect contextual elements: *narrow street, wide street, pedestrian crossing sign, zebra crossing, stop sign , traffic light, parking lot.* Output of the last layer from this network is used as a context vector to supplement crossing intent classification. As can be seen in the Table 2.2, the addition of context significantly improves classification performance by 20%.

Chapter 3

Related Work

A large number of human-centric vehicle studies emphasize humans inside the vehicle. Ohn-Bar and Trivedi in their survey highlight the roles of humans in intelligent vehicle interactions [16]. Human activity analyses, covering driver gestures, distraction and manoeuvre classification and pre-condition, and eye and gaze analysis have been widely conducted [17–22]. On the other hand, direct observation of pedestrians and occupants of nearby vehicles is less well studied though this is an important area of current interest [16].

3.1 Humans and Crossing

Observing humans in the vicinity of intelligent vehicles is essential for smooth and safe navigation. An intelligent vehicle should be able to sense and predict human intent because the road is shared with pedestrians. Studies on pedestrian behaviour, styles, skill, attention, distraction and social-forces are still in the early stages [16]. Schmidt and Farber [14] in their work conclude that at least one part of the human body, either the head, the upper-body or the legs must be visible for a human driver to accurately predict a pedestrian's future movements. Koehler et al. [23] model a pedestrian's contour, particularly the spread of the legs and bending of the body to infer crossing intent. Keller and Gavrilla [24] explore motion features using dense optical flow to capture upper-body and leg movements. They use a low-dimensional flow-based histogram to create a special trajectory representation by linking the motion features to the pedestrian's positions. Quintero et al. [25] study a larger variety of body-parts including arm-movements to create sparse geometrical representations for them. They detect 3D-poses of the body parts and joints with stereo cameras and use them to learn pedestrian dynamics in latent space. Age, gender and other human properties of the interacting pedestrians influence driving behaviour. Modelling social relationships among agents is another method employed by drivers to recognize and communicate intent [26, 27].

3.2 Time Series Predictions

Starting from an identical initial position, two pedestrians will rarely follow the same trajectories in order to cross a road. Their trajectories depend on their own preferences, on the environment and on the presence of other pedestrians. An accurate prediction of gait and positions of a crossing pedestrian would imply that the learnt model understands dynamics of motion to a reasonable extent. Most common approaches for future trajectory predictions are based on the Markov property. A *Markov Decision Process* (MDP) is used to express the dynamics of a decision-making process and is defined by an initial state distribution $p(s_0)$, a transition model $p(s'|s, a)$ and a cost function $r(s)$ [28]. Given these parameters, optimal control can be solved for by learning a policy $\pi(a|s)$ which encodes the distribution of the decision a to be taken in state s. *Hidden Markov Model* (HMM) based approaches, in which the hidden state is the pedestrian's intent [29, 30] have been extended to partially-observable Markov Decision Processes (POMDP) to learn a distribution over pedestrian intent [31]. Kitani et al. propose a hidden variable Markov decision process (hMDP), assuming that the observer has noisy observations of an actor, where the actor is fully aware of its own state. In contrast, in a POMDP, the actor is uncertain about its own state and the observer is not modelled. Although a Markovian process allows for the model to quickly adapt to changes in its observations, the Markov assumption itself can be overly restrictive, owing to insufficient prior conditioning.

Other approaches for a time-series prediction are to assume the samples come from a linear process driven by a white, zero-mean, Gaussian input [32]. Gaussian processes perform well with noisy observations and have a closed-form predictive uncertainty [33, 34]. *Gaussian process* mixture models enable predictions for both intent and trajectory uncertainty. Schneider and Gavrilla [13], in their work provide a comparative study of recursive Bayesian filters for pedestrian path prediction. They consider Extended Kalman Filters based on single dynamical models and Interacting Multiple Models (IMM) combining several basic models such as constant velocity or constant acceleration. These are applied to four typical pedestrian scenarios: Crossing, Stopping, Bending in and Starting. Gaussian processes are slower to detect a change than Markovian models because the entire observed trajectory is used to predict the future state [35]. Additionally they are too slow to learn previously unobserved behaviour patterns. Ferguson et al. propose a novel changepoint detection and clustering algorithm to combat the slower approaches in existing Gaussian process methods [36]. They show that the resulting long-term movement predictions demonstrate improved accuracy in both intent and trajectory predictions.

Restricting the order of an autoregressive time-series model yields predictions that are accurate at shorter time scales, but inefficient beyond a few steps, as the processes are stationary. It is also difficult to embed context, such as current traffic state, into these models. Switching linear dynamical models as extensions to these models are shown to

work in constrained environments [13, 37]. Karasev et al. [32] account for more accurate long-term predictions by postulating goals to a pedestrian's trajectory that they navigate to by approximately following stochastic shortest paths. Kooij et al. in their work incorporate a *Dynamic Bayesian Network* 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. A non-parametric approach based on Gaussian processes proposed by Ellis et al. [35], although accurate on longer time-scales, is computationally expensive. Batkovic et al. present a computationally efficient mathematical model for pedestrian motion over a finite horizon up to 10 s [38]. However they assume a rational pedestrian behaviour and the model is based on a pre-conditioned road map structure.

3.3 Data-driven Modelling

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 [39]. The solution is to enable pedestrian activity models to work in the image space by directly analysing videos from a car's dashboard camera. Together with the availability of large-scale annotated datasets, convolutional deep learning architectures are breaking new ground in learning generalizable features due to supervision provided in an end-to-end fashion [4, 40–42]. These models can leverage a multitude of information available across video frames, including behavioural cues from crossing pedestrians. Inferring, say a crossing intent in a pedestrian requires a time-series model of their actions. Both Long Short Term Memory (LSTM) [43] and 3D convolutional networks are popular algorithms used to study temporal variations of features learnt from images. Hasan et al. [44] treat the prediction of adverse pedestrian actions as an anomaly detection problem. They first leverage hand-crafted features like the Histogram of Oriented Gradients (HOG) and Histogram of Optical Flows (HOF) in a temporal cuboid as a spatio-temporal appearance feature descriptor and learn a fully-connected feed-forward autoencoder to learn temporal regularity across video frames. For comparison, they build a fully-convolutional autoencoder and anomaly classifier in an end-to-end fashion. Their deep-learnt model consistently outperforms hand-crafted feature model across datasets such as the Avenue [45] and UCSD pedestrian dataset [46].

Our work is based on the hypothesis that by re-generating a scene we learn a visual structure of the scene itself. And by generating a next-frame future conditioned upon the past, our model builds representations that also address motion in the scene. Although generative video models can impact many applications in video understanding and simulations, not much work has been done in this regard. Existing generative architectures struggle with the dimensionality of the input image space and are in many instances not scalable to nat-

ural images. Consequently, they also suffer from shorter prediction time horizons. Vondrick et al. [47] use a Generative Adversarial Network to predict future frames using only one frame as the conditioning frame. Their model can sometimes predict a plausible, but incorrect video. Lotter et al. [48] use a weighted mean-squared-error and adversarial loss to predict future frames in the bouncing balls dataset [49]. However, extension of this work to natural image sequences is not presented. Lotter et al. in their subsequent work [50] study the future prediction problem with the idea that each layer of prediction only accounts for local changes and forward deviations higher up. We incorporate a similar strategy by using residual connections across different scales of upsampling based on the idea that residual connections encourage iterative inference from the latent space. Jastrzebski et al. [51] in their work further provide observational and analytical evidence for such this.

Typical neural machine translation architectures use recurrent networks in an encoder-decoder stack to translate an n length sequence in the source language to an m length sequence in the target language [52–54]. We follow a similar network structure philosophy of translating past frames to the future. Unlike typical NMT architectures, our encoder is 3D-convolutional in operation and our decoder is a multi-stage recurrent up-sampling network. In similar work Kalchbrenner and Blunsom [55] use 1-D convolutions to learn hierarchical representations for English input sentences and recurrent layers conditioned on this representation to derive French translations.

Chapter 4

Technical Description

4.1 Predictive Networks

We formulate the prediction of crossing intent in pedestrians as a multi-objective problem. In stage one, we learn a sequence of hidden representations for the input such that they encode a regularity in spatial layout as well as variance and rate of variance in temporality. Stage two is about an autoregressive consumption of these representations to predict an image sequence. Appearance cues such as posture are important to model in order to avoid blindly predicting a motion continuing from the past frames. The third stage is the analysis of the predicted video to determine the probability of a crossing event in the scene. A video presents an action classifier with a more defined *future* to reason from, than working from a latent space. This way we leverage the continuity of visual context also factoring the camera's own motion (egomotion) in the predictions. We approach the three stages with neural networks mainly because of the data-driven generalizability of deep networks alongside the capability of learning hierarchical features from error feedback.

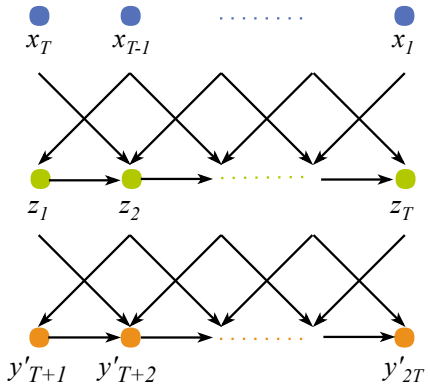


Figure 4.1: Abstract representation of the future generation process. A sequence of learnt representations iteratively condition the decoder.

Recurrent learning of dense representations along with low perplexity language models have lead to ground-breaking research in Neural Machine Translation (NMT). The general

structure of NMT architectures contain an encoder network that recursively learns a dense representation z for a sentence in the source language. Conditioned on this z, a decoder further recursively predicts a variable length sequence of words in the target language [56] [52]. In the context of predicting a sequence of frames, we propose that the conditioning representation $\boldsymbol{z}$ plays a critical role in influencing the plausibility of the sequence. Frames from both the past and the future share a significant visual structure. We hypothesize that learning T representations, $\boldsymbol{z} = \{z_1, ..., z_T\}$ for an input sequence of frames $\boldsymbol{x} = \{x_1, ..., x_T\}$ offers more transfer of information than that of a comprehensive z for the entire input sequence. Learning a sequence of representation vectors affords *Iterative Inference* to the future generator. The earlier vectors contribute more towards the spatial reasoning of the predictions. The later vectors successively transform predictions to a later time.

A relative prominence of input frames in the learnt z will arise because of their temporal ordering. In other words, x_T should contribute more to z_1 than x_1. Reversing the input sequence from $\boldsymbol{x} = \{x_1, ..., x_T\}$ to $\boldsymbol{x} = \{x_T, ..., x_1\}$ implements this sort of conditioning. However, as a result, the earlier frames have diminishing influence on predictions. Combined with lower pixel-level variations between the consecutive input frames for objects in motion, small resolutions for pedestrians at a distance lead to motion itself not being recognized. We address this particular issue by recognizing that feeding a convolutional kernel with frames deeper in time would activate more neurons to the now more apparent motion. We implement this idea by time dilating the convolution operations.

4.2 Structuring Predictions

4.2.1 Dilated Convolutions

Dilated convolutions, also referred in literature as *à trous* convolutions, is the method where the convolution filter is applied to an area larger than the specified filter size by overlooking certain input values. In practice, it can be seen as applying a larger filter over the original size and dilated with zeros. Yu and Koltun [57] mathematically formalize a dilated convolution operation in equation 4.1; where $F : \mathbb{Z}^2 \to \mathbb{R}$ is a discrete function representing the input. Let $\Omega_r = [-r, r]^2 \cap \mathbb{Z}^2$ and $k : \Omega_r \to \mathbb{R}$ be the filter of size $(2r + 1)^2$. A discrete convolution operation $*_l$ is defined with the dilation factor l, or simply an l-dilated convolution. As a special case, a dilated convolution with dilation 1 is the same as the standard convolution. Stacking dilated convolutions enables a network to have very large receptive fields with just a few layers while preserving the input resolution for the same number of trainable parameters, thus also being computationally efficient.

$$(F *_l k)(\boldsymbol{p}) = \sum_{\boldsymbol{s}+l\boldsymbol{t}=\boldsymbol{p}} F(\boldsymbol{s})k(\boldsymbol{t}) \tag{4.1}$$

Reversing input data allows spatial conditioning of the decoder on the most recent frame as described in Section 4.1. However, the compositional differences between subsequent frames may not be enough for the network to understand motion. This occurs because of the relatively small pixel boundaries for pedestrians in resized frames. In this work we dilate our convolutions in the temporal dimension in order to combat this problem. The goal of using *time-dilated causal convolutions* is to expand the temporal depth of the 3D convolutional filter, retaining the resolution and size of the network. This also aids the network in learning the differential rate of motion for the various objects in the scene (pedestrian and vehicles). Figure 4.2 visually represents how stacking multiple dilated filters achieves a greater temporal receptive field compared to an undilated filter hierarchy.

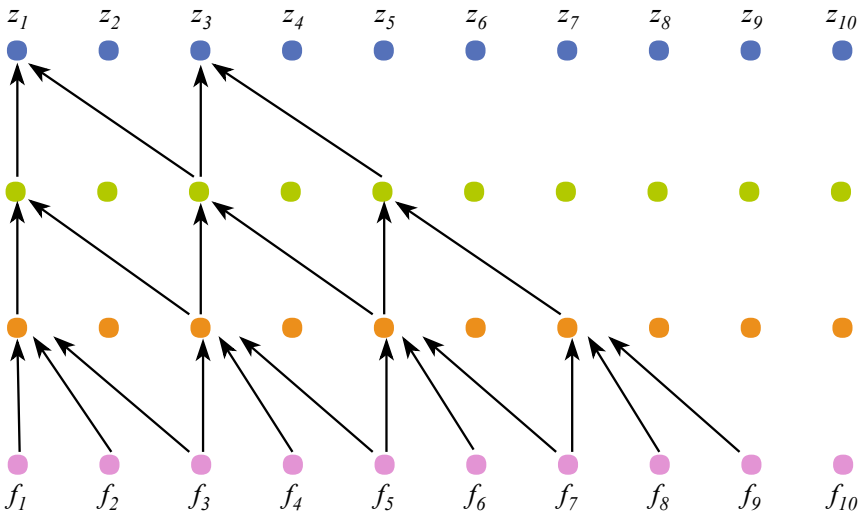


Figure 4.2: An abstract representation of the dilated convolution operations. The first level of transformations are undilated with a kernel 3-deep in time. Levels two and three are each 2-dilated.

4.2.2 Residual Connections

With increasingly deep neural networks, the accuracy of the network is known to saturate and then degrade rapidly. This goes against the idea that a deeper model should produce no higher a training error than its shallower counterpart. The degradation itself is not caused by overfitting but due to vanishing or exploding gradients. Residual connections, introduced by He et al. [4] were originally formulated to tackle this problem with respect to the trainability of deep neural networks. Specifically, training compositional neural networks deeper than 15-20 layers. One treatment of its internal working is that the later layers of a residual block as shown in Figure 4.3 optimize a residual mapping leftover from the earlier layers. The additive framework used to transform representations allows an exponential number of

paths between the input and the prediction layer. The resulting network can be seen as an ensemble of many shallower units making it easier for the deep network to be optimized.

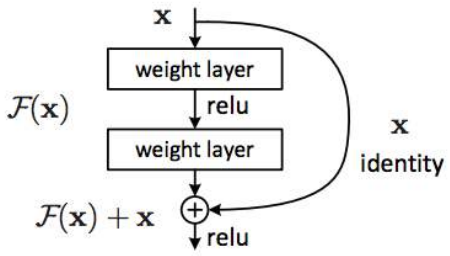


Figure 4.3: A residual block [4].

In our work, we hypothesized that these residual blocks at varying scales of decoding hidden representations, can lead to what Jastrzebski et al. [51] formalize as *Iterative Refinement*. The insight we develop is that at different scales of image generation, each residual block will learn hidden representations that are different from each other. Also, by upsampling an earlier residual block's output, the subsequent block does not have to start from scratch. The later blocks only have to refine the images pixel-wise rather than compositionally, as in the earlier blocks. Jastrzebski et al. in their recent work from 2018, formally present this idea concluding that ResNets perform both representation learning and iterative refinement.

4.2.3 Convolutional LSTM

Recurrent neural layers are standard in many sequence modelling problems. The Long Short Term Memory layer [43] is perhaps the most used recurrent layer to learn long-range dependencies in sequences. A major drawback of the LSTM is that it has to unfold its input vectors to 1D, thereby losing spatial information of the data. We use a modification of the LSTM with dense layer connections replaced with convolution operations called the Convolutional LSTM [5]. The idea is represented pictorially in Figure 4.4. Key equations of the ConvLSTM are shown in (4.2) where '$*$' represents a convolution operation and '$\circ$' represents Hadamard product, $\{X_1, ..., X_t\}$ represent inputs, cell outputs are $\{C_1, ..., C_t\}$, hidden states are $\{H_1, ..., H_t\}$, and the gates i_t, f_t, o_t of the ConvLSTM are 3D tensors whose last two dimensions are the spatial dimensions. In our work, we leverage the recurrent learning ease of the ConvLSTM layers to decode learnt representations to images.

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * x_t + W_{hf} * x_{t-1} + W_{cf} \circ c_{t-1} + b_f) \\
c_t &= f_t \circ c_{t-1} + i_t \circ tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} \circ c_t + b_o) \\
h_t &= o_t \circ tanh(c_t)
\end{aligned} \tag{4.2}
$$

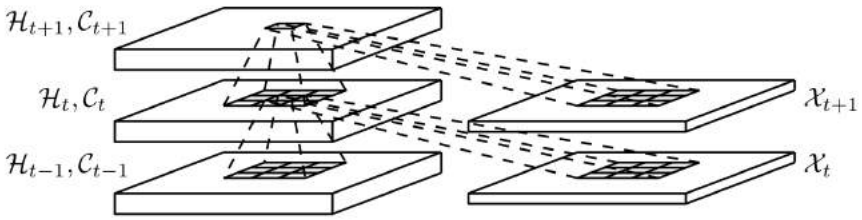


Figure 4.4: The convolutional structure of ConvLSTM [5].

4.3 Future Generation

The task of generating image frames falls into the general category of *dense prediction* problems like semantic segmentation. Modern image classification networks integrate multi-scale contextual information via successive pooling and subsampling until a global prediction is obtained as a weighted measure of the learnt features. In contrast, dense prediction calls for multi-scale compositional reasoning with pixel-level full-resolution output. Although, classification networks have been successfully repurposed for semantic segmentation, the two problem are structurally different [57]. Our method is to employ an encoder-decoder architecture which we describe later.

The objective of our model is to predict the future positions of objects like vehicles and pedestrians in the world by learning their motion. In order to do so, we build a decoder that transforms hidden representations of the input, learnt by an encoder. Representations are decoded to a sequence of frames that minimize a pixel-wise error between the predicted frames and the ground truth. Functionally, the encoder reads a sequence of frames $\boldsymbol{x} = \{x_T, ..., x_1\}$ to yield dense representation $\boldsymbol{z} = \{z_1, ..., z_T\}$. Conditioned on $\boldsymbol{z}$, the decoder will then auto-regressively predict an image sequence $\boldsymbol{y'} = \{y'_{T+1}, ..., y'_1\}$ by minimizing a loss between $\boldsymbol{y'}$ and ground truth frames $\boldsymbol{y} = \{y_{T+1}, ..., y_1\}$. Each generated frame is of the same resolution as the input (128×208 in our case). Intuitively, the encoder's job is to comprehend the visual content in input frames. Recursively learning representations from each frame, we expect to first learn a *temporal regularity* in the early representations and parametrize a *temporal variance* in the later representations. The decoder, on the other

hand, is an autoregressive consumer of these hidden representations. We arrived at these designs and procedures after extensive iterated experiments. The following sections will provide details for our *Res-EnDec* model, our most successful one.

4.3.1 Encoder

The encoder is a spatio-temporal neural network composed of three-dimensional convolutional layers. 3D convolutions are good feature learning machines that can model motion and spatial appearance simultaneously. Tran et al. [6] in their paper on learning spatiotemporal features from 3D convolutions, empirically find that $3 \times 3 \times 3$ convolutional kernels for all layers work best among the set of architectures explored. Their learnt features with a linear classifier outperformed the state-of-the-art on the Sports-1M action classification challenge [58]. In contrast we found that 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 better. A large spatial kernel (11x11 in our case) along with a stride with much overlap between them in the first layer was observed to produce more detailed representations. We believe that this allows the network to account for more spatial features per time frame. We use strided convolutions to down-sample the output instead of max-pooling layers. Residual connections are introduced at two image resolutions: 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 first residual block (Layer 3 in the Table 4.1) 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 as represented in equation (4.3). The learnt $\boldsymbol{z}$ is 16x26 dimensional. We abstract the mathematical formulation for 3D convolutions in the equation 4.3 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. The Table 4.1 presents more architectural details of the encoder. We use time distributed 1×1 2D convolution operations for dimensionality matching in addition operations for residual connections.

$$
\begin{aligned}
h_t^{al} &= f_a(\boldsymbol{W}_a(r_{t:t-k}^{l-1}) + \boldsymbol{b}_a) \\
h_t^{bl} &= f_a(\boldsymbol{W}_b(h_{t:t-k}^{al}) + \boldsymbol{b}_b) \\
r_t^l &= r_t^{l-1} + h_t^{bl}
\end{aligned}
\tag{4.3}
$$

4.3.2 Decoder

The decoder is recurrent in nature, containing convolutional LSTM layers. ConvLSTM layers, interspersed with up-sampling layers, go from the low-dimensional representation

	Type	Kernel	Stride	Dilation	Input	Output
E1	Conv3D	$3 \times 11 \times 11$	$1 \times 4 \times 4$	$1 \times 1 \times 1$	$128 \times 208 \times (3 \times 16)$	$32 \times 52 \times 128$
E2a	Conv3D	$2 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 1 \times 1$	$32 \times 52 \times 128$	$32 \times 52 \times 64$
E2b	Conv3D	$2 \times 5 \times 5$	$1 \times 1 \times 1$	$2 \times 1 \times 1$	$32 \times 52 \times 64$	$32 \times 52 \times 64$
E2c	Conv2D	1×1	1×1	1×1	$32 \times 52 \times 128$ (E1)	$32 \times 52 \times 64$
E3	Add				$32 \times 52 \times 64$ (E2b)	$32 \times 52 \times 64$
					$32 \times 52 \times 64$ (E2c)	
E4	Conv3D	$3 \times 5 \times 5$	$1 \times 2 \times 2$	$1 \times 1 \times 1$	$32 \times 52 \times 64$	$16 \times 26 \times 64$
E5a	Conv3D	$2 \times 3 \times 3$	$1 \times 1 \times 1$	$2 \times 1 \times 1$	$16 \times 26 \times 64$	$16 \times 26 \times 64$
E5b	Conv3D	$2 \times 3 \times 3$	$1 \times 1 \times 1$	$2 \times 1 \times 1$	$16 \times 26 \times 64$	$16 \times 26 \times 64$
E6	Add				$16 \times 26 \times 64$ (E4)	$16 \times 26 \times 64$
					$16 \times 26 \times 64$ (E5b)	

Table 4.1: Architectural details of the encoder. Input and output sizes are indicated per time slice of a 16-frame video.

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.

$$\begin{aligned} h_t^{al} &= f_a(\boldsymbol{W}_a(r_t^{l-1})^\dagger + \boldsymbol{U}_a h_{t-1}^{al} + \boldsymbol{b}_a) \\ h_t^{bl} &= f_b(\boldsymbol{W}_b(h_t^{al})^\dagger + \boldsymbol{U}_b h_{t-1}^{bl} + \boldsymbol{b}_b) \\ r_t^l &= r_t^{l-1} + h_t^{bl} \\ (r_t^l)^\dagger &= Upsamp(r_t^l) \\ r_t^0 &= z_t \end{aligned} \tag{4.4}$$

Residual blocks play an important role in the generation of image sequences as described in Section 4.2.2. We introduce residual connections at three image scales ($16 \times 26, 32 \times 52$ 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 (out of 16) 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. Such a sharp reduction in the number of filters is an extension to the compositional feature learning and iterative refinement capability of the residual blocks. Our interpretation is that the greater number of filters early on offer more opportunities for a structural transformation due to smaller image resolutions. The reduced number of filters in the later stages, afford the network a lesser optimizing effort. The final transformations are encouraged to be more refining than compositional because of iterative refinement in

	Type	Kernel	Stride	Input	Output
In1	Input			16 × 26 × 64 (E6)	
In2	Input			32 × 52 × 64 (E3)	
D1a	ConvLSTM2D	3 × 3	1 × 1	16 × 26 × 64	16 × 26 × 128
D1b	ConvLSTM2D	3 × 3	1 × 1	16 × 26 × 128	16 × 26 × 128
D2	Add			16 × 26 × 128 (In1)	16 × 26 × 128
				16 × 26 × 128 (D1b)	
D3	Upsamp3D	1 × 2 × 2		16 × 26 × 128	32 × 52 × 128
D4a	ConvLSTM2D	3 × 3	1 × 1	32 × 52 × 128	32 × 52 × 64
D4b	ConvLSTM2D	3 × 3	1 × 1	32 × 52 × 64	32 × 52 × 64
D5	Add			32 × 52 × 64 (In2)	32 × 52 × 64
				32 × 52 × 64 (D3)	
				32 × 52 × 64 (D4b)	
D6	Upsamp3D	1 × 2 × 2		32 × 52 × 64	64 × 104 × 64
D7a	ConvLSTM2D	3 × 3	1 × 1	64 × 104 × 16	64 × 104 × 16
D7b	ConvLSTM2D	3 × 3	1 × 1	64 × 104 × 16	64 × 104 × 16
D7c	Conv2D	1 × 1	1 × 1	64 × 104 × 64 (D6)	64 × 104 × 16
D8	Add			64 × 104 × 16 (D7b)	64 × 104 × 16
				64 × 104 × 16 (D7c)	
D9	Upsamp3D	1 × 2 × 2		64 × 104 × 16	128 × 208 × 16
D10	ConvLSTM2D	3 × 3	1 × 1	128 × 208 × 16	128 × 208 × 3

Table 4.2: Architectural details of the decoder. Input and output sizes are indicated per time slice of a 16-frame video.

residual networks (Section 4.2.2). Each decoder layer's function can be elementarily defined as in equation (4.4). We abstract the hidden state dependences from less relevant convolution operations in the recurrence formulation of the ConvLSTM. More architectural details of the decoder are present in the Table 4.2. A 1×1 time distributed convolution is used to project inputs to match dimensions in residual connections.

4.4 Action Recognition

The task of action recognition is motivated by the idea that by looking ahead in time, we could counter a potentially hazardous pedestrian action. To do this in an end-to-end fashion, we append a binary action classifier to our 16-frame generator. In this task, we want to learn to predict a pedestrian's crossing intent across a multitude of scenarios and behavioural sequences. Accuracy of prediction serves as an evaluation metric to gauge the quality of generated images. Formally, the classifier network parametrized by θ predicts the probability of a crossing event $P(a)$ in a scene as a sigmoidal function of the generated frames $\boldsymbol{y}' = \{y'_T, ..., y'_{2T}\}$ as shown in equation 4.5. Figure 4.5 pictorially represents the same.

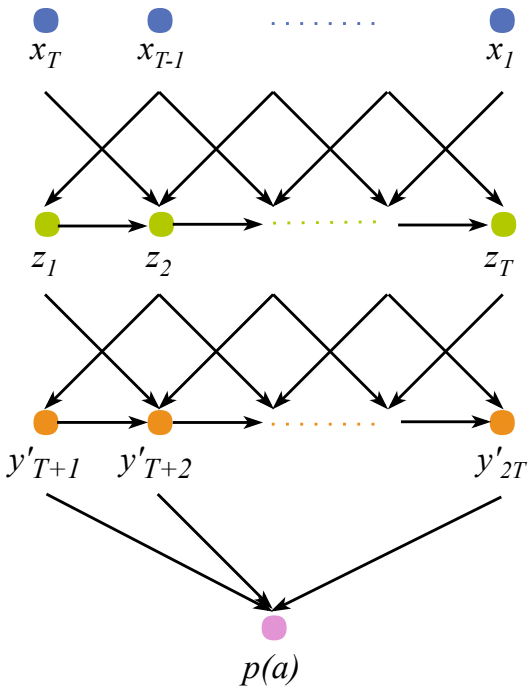


Figure 4.5: Abstract representation of the action recognition process. The generated future is analyzed to predict a pedestrian's crossing intent.

$$P(a|\boldsymbol{y'}_{T:2T}) = \sigma\{f_\theta(\boldsymbol{y'}_{T:2T})\} \tag{4.5}$$

	Type	Input	Units	Output
C1a	C3D	$112 \times 112 \times 3$		4096
C1b	C3D	$112 \times 112 \times 3$		4096
C2	Concatenate	4096 (C1a)		8192
		4096 (C1b)		
C3	Dense	8192	1024	1024
C4	Dense	1024	1	1

Table 4.3: Architectural details of the classifier. Input and output sizes are indicated per time slice of a 16-frame video.

We fine-tune the C3D network presented by Tran et al. [6] pre-trained on the Sports 1M dataset [58] as our classifier. The pretrained model accepts 16 frames of size 112×112 whereas our model generates frames of size 128×208. As most occurrences of the beginning and end of a crossing event take place towards either the left or right edges of the frame (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 new 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 5.2. More information on the architecture can be found in the Table 4.3.

Chapter 5

Experiments and Discussion

5.1 Video Prediction

5.1.1 Experimental Setup

The JAAD dataset consists of 346 traffic videos of pedestrian interaction scenarios. We use 95 (∼30%) of them in our test set and 35 (∼10%) for validation. We omit 8 videos sampled at a higher frame rate of 60 FPS, leaving 208 videos (∼60%) in 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 training videos and pack them in randomly shuffled batches. We train the encoder-decoder stack described in the previous chapter to optimize for a combination of l_1 and l_2 losses as shown in equation 5.1. 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 input frames are normalized across the three RGB channels to fall within the range [-1, 1]. We use *BatchNormalization* [59] layers after every convolutional layer. The outputs in the encoder are activated with *LeakyReLU* functions with $\alpha = 0.2$ whereas we use *tanh* activations in the decoder. We also use *Dropout* layers to avoid overfitting. 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 and 20 epochs. We then train the network for 10 more epochs to minimize l_1 loss for 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). We experimented with various architectures for the 16-frame generator that lead to lower losses and enhanced visual coherence that we evaluate qualitatively and quantitatively in the following sections. All training and experiments are run on an Nvidia GTX 1080Ti GPU.

$$\mathcal{L} = \frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{N}(y_{t,i} - y'_{t,i})^2 + \lambda\frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{N}|y_{t,i} - y'_{t,i}| \tag{5.1}$$

5.1.2 Qualitative Analysis

We train three kinds of models for future prediction: 1. Fully convolutional model (*Conv3D*) 2. Recurrent decoder model (*Segment*) and 3. Residual encoder-decoder model (*Res-EnDec*). In Figure 5.1, we show a block diagram view of network architectures experimented with. We then perform ablation studies on the residual encoder-decoder model described in the previous chapter. Figures 5.2 - 5.6 present examples of videos predicted by each of these models as a sequence of frames stacked one below the other for comparison. The sequences progress in time towards the right in every row. We only show some key-frames for brevity. Predicted videos in the graphics interchange format are posted online [1].

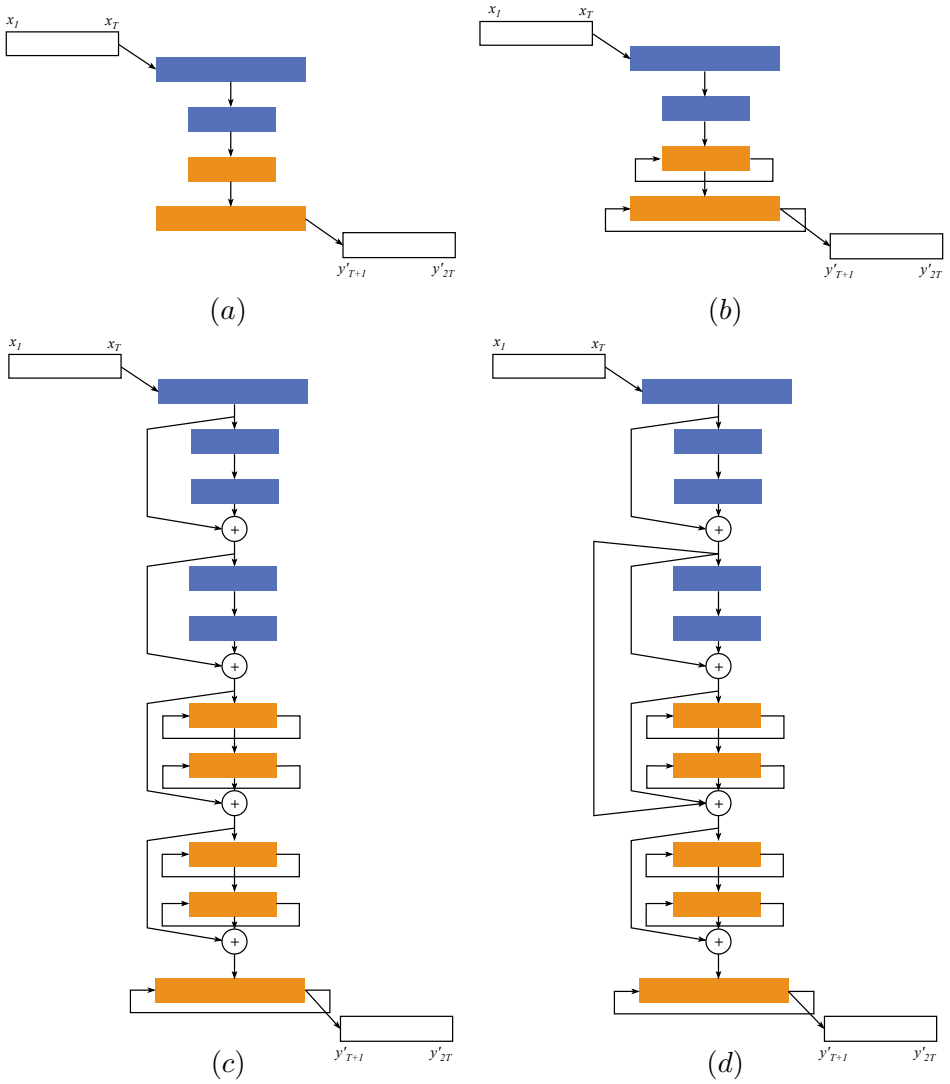


Figure 5.1: Block diagram for (a) Conv3D model (b) Segment model (c) Res model (d) Res-EnDec model

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

Figure 5.2: Example predictions by various models. Every third frame shown for brevity. Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: EnDec model; Row 6: Undilated model; Row 7: Unreversed model; Row 8: Segment model; Row 9: Conv3D model. The car is turning right towards the pedestrian who is crossing the street leftwards. Visually, the Res-EnDec model seems to be able to define the background better than all other models.

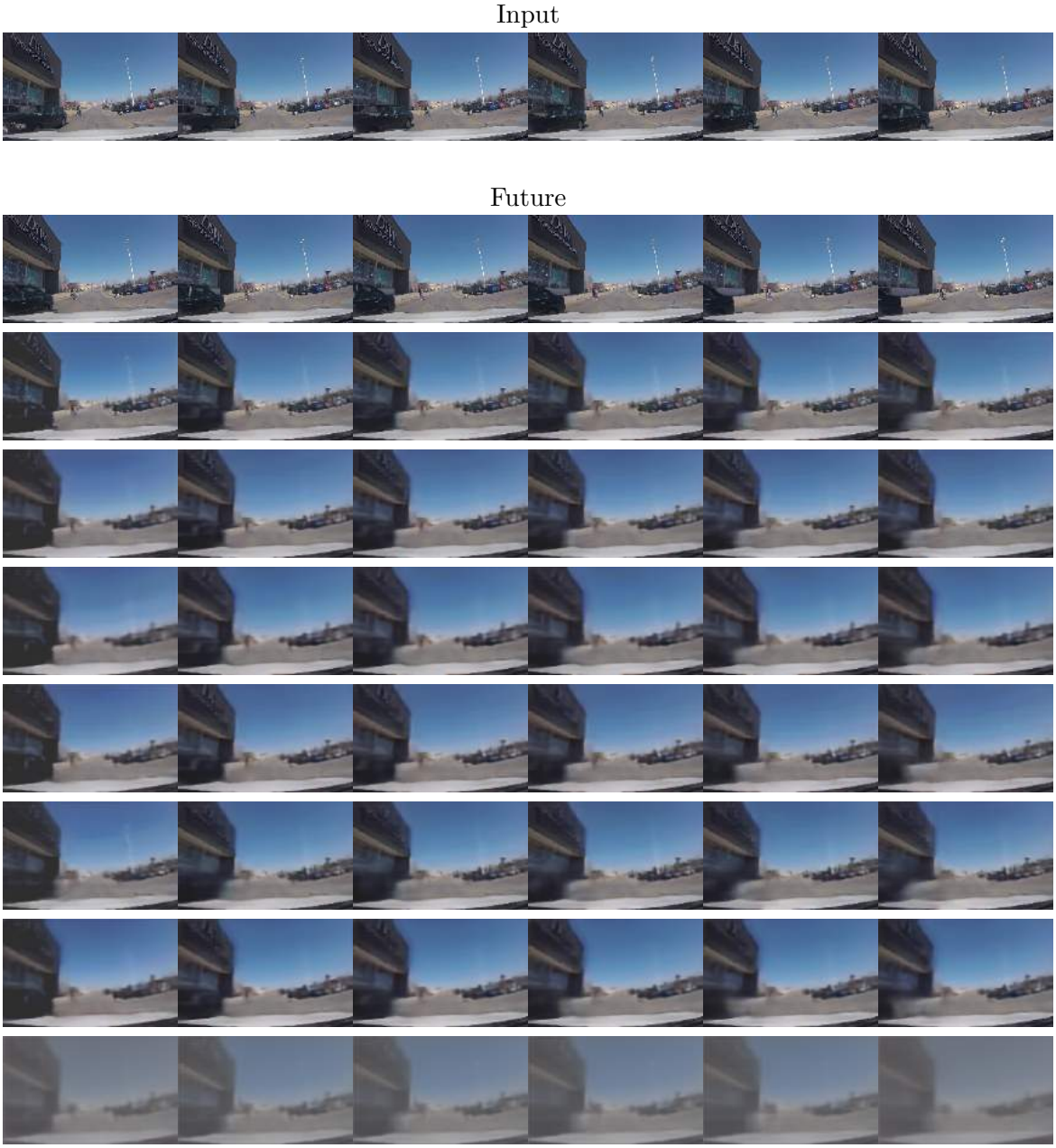


Figure 5.3: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: EnDec model; Row 6: Undilated model; Row 7: Unreversed model; Row 8: Segment model; Row 9: Conv3D model.

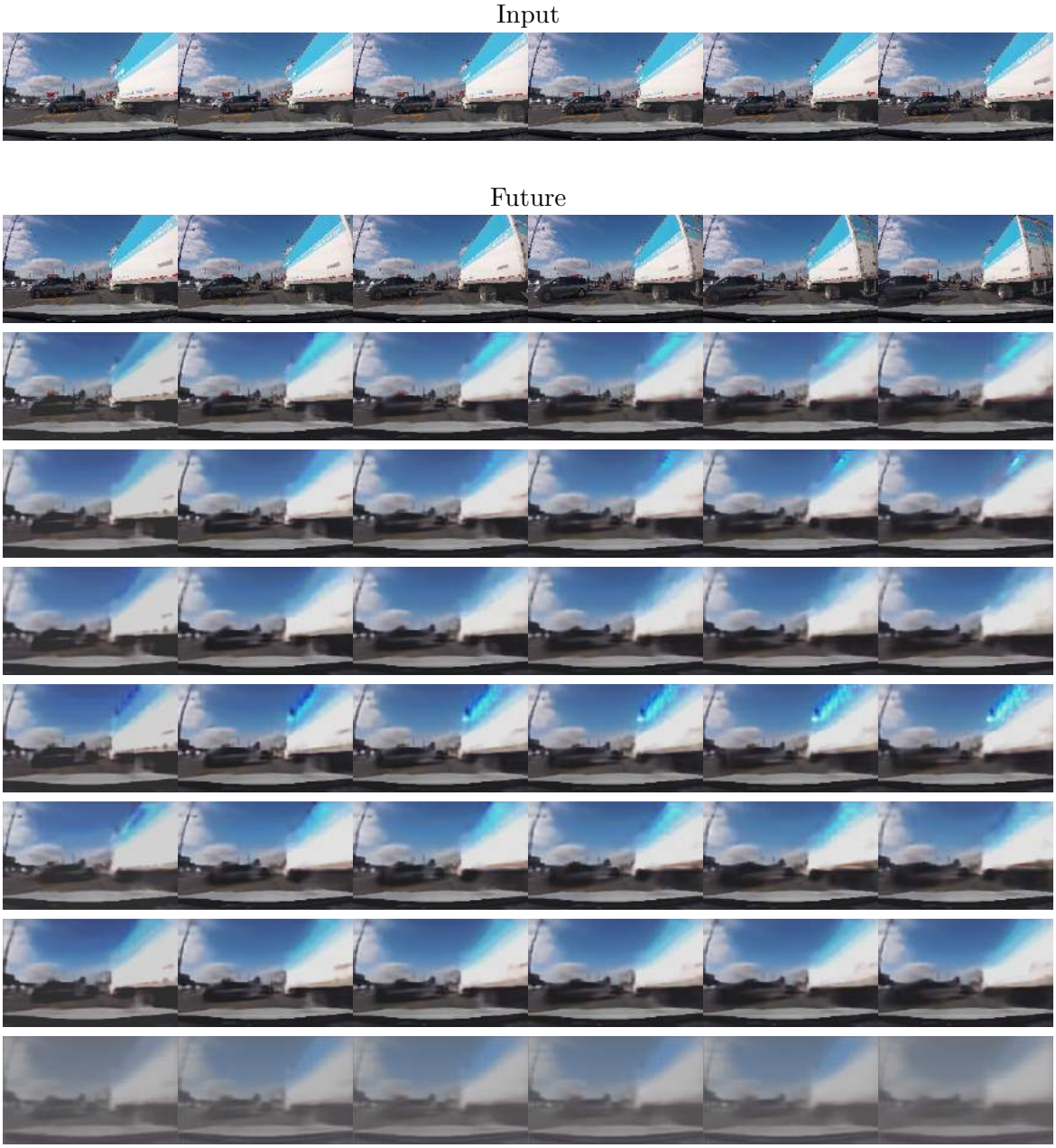


Figure 5.4: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: EnDec model; Row 6: Undilated model; Row 7: Unreversed model; Row 8: Segment model; Row 9: Conv3D model.

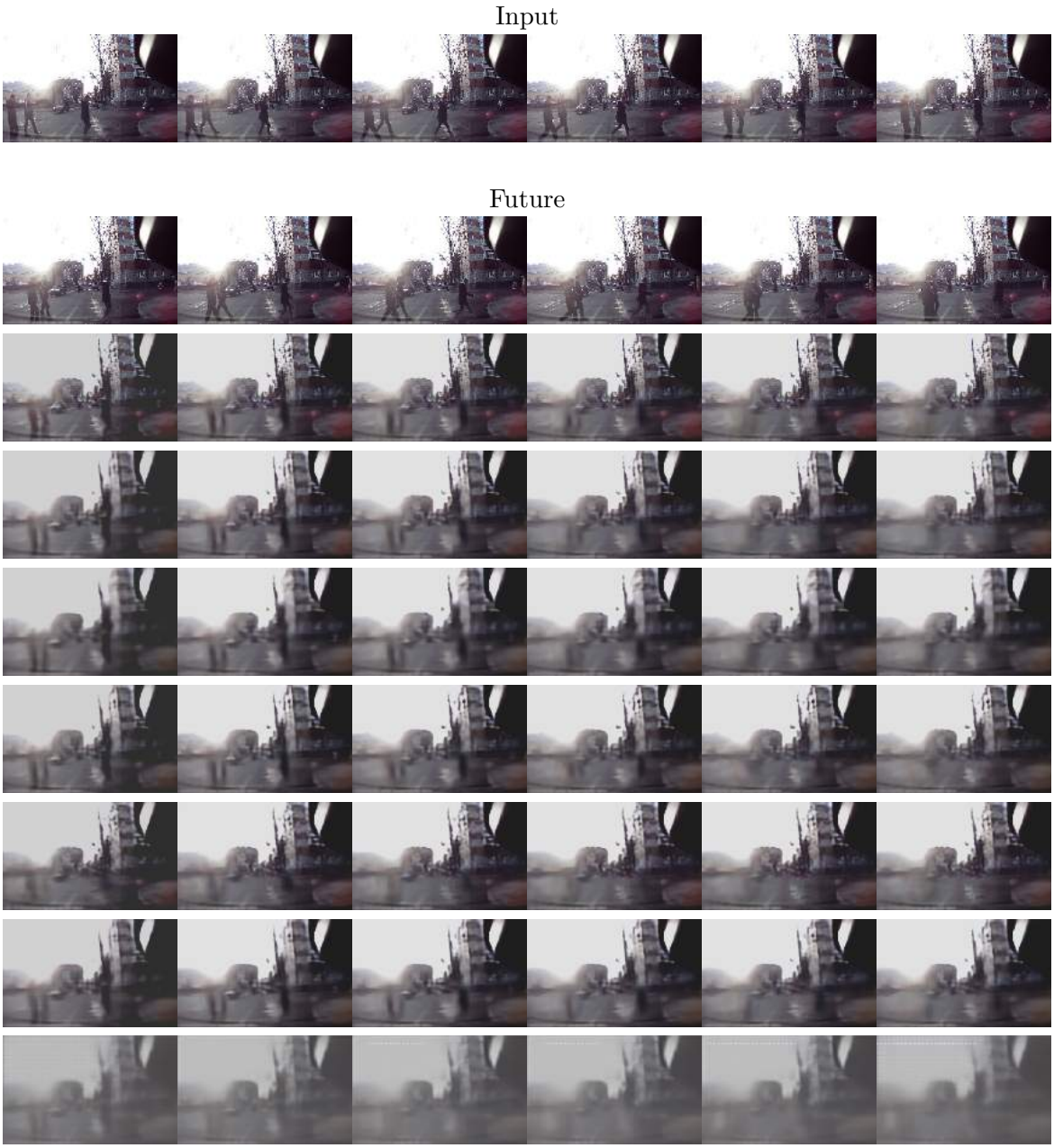


Figure 5.5: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: EnDec model; Row 6: Undilated model; Row 7: Unreversed model; Row 8: Segment model; Row 9: Conv3D model. Both Res-Endec and Res models falter more than other models towards the end of the sequence with a poor definition of the pedestrians.

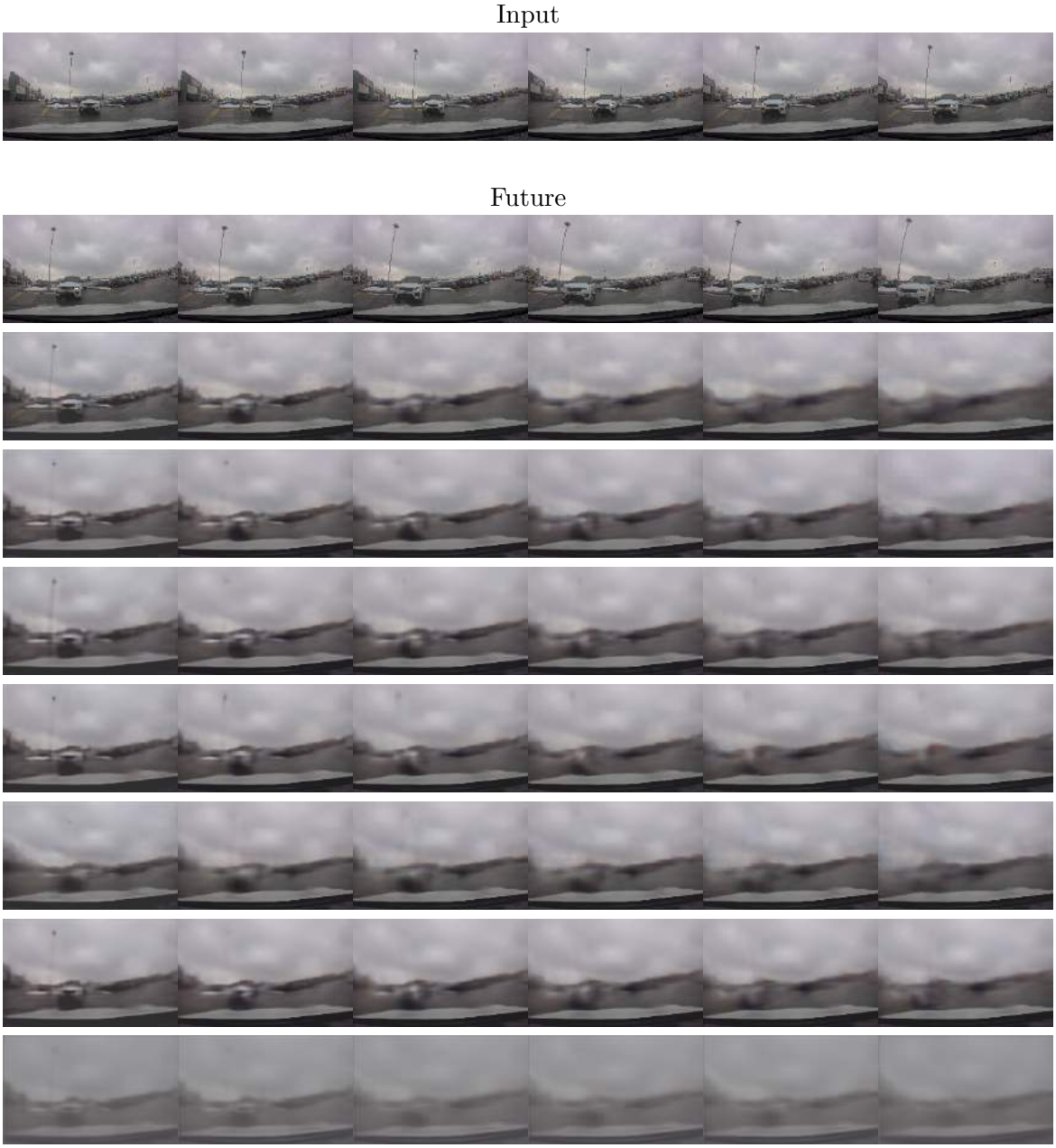


Figure 5.6: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: EnDec model; Row 6: Undilated model; Row 7: Unreversed model; Row 8: Segment model; Row 9: Conv3D model. This example is a case of a featureless scene with the car turning rightwards in the parking lot. Almost all predictions quickly become inaccurate.

5.1.3 Quantitative Analysis

Figure 5.7 shows the training progress for the *Res-EnDec* model. As can be seen, l_2 loss saturates around the eighth epoch and l_1 loss varies in the fourth decimal in the graph on the right, though at first glance it would not appear to saturate. The average pixel-wise prediction l_1 error is $(1.37 \pm 0.37) \times 10^{-1}$ for this model. The errorbar l_1 loss TVG for the model (Figure 5.8) showcases an increase in prediction error, as the errors can be expected to accumulate as one goes further in time. We suspect the high initial error in the TVG to be because of the substantial influence the last input frame has on the generation because of our reverse ordering of the input. Also, the model appears to configure the spatial manifestation of the scene first and then successively transform objects to project in motion, due to the observed drop in prediction loss from the second frame. The shaded blue area in the TVG shows standard deviation in l_1 loss for $k-$th predicted frame. Standard deviation in error over consecutive frames remains fairly consistent. We believe these errors primarily originate from prediction uncertainties for objects in motion and because motion itself remains fairly uniform, the deviation remains consistent with an almost linear increase.

Model	l_1 loss (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
Undilated	1.42 ± 0.36	15.60
Unreversed	$\mathbf{1.49 \pm 0.39}$	**5.22**
Res	1.42 ± 0.36	15.88
Res-EnDec	$\mathbf{1.37 \pm 0.37}$	**19.86**

Table 5.1: Analysis of loss variation in time for various models

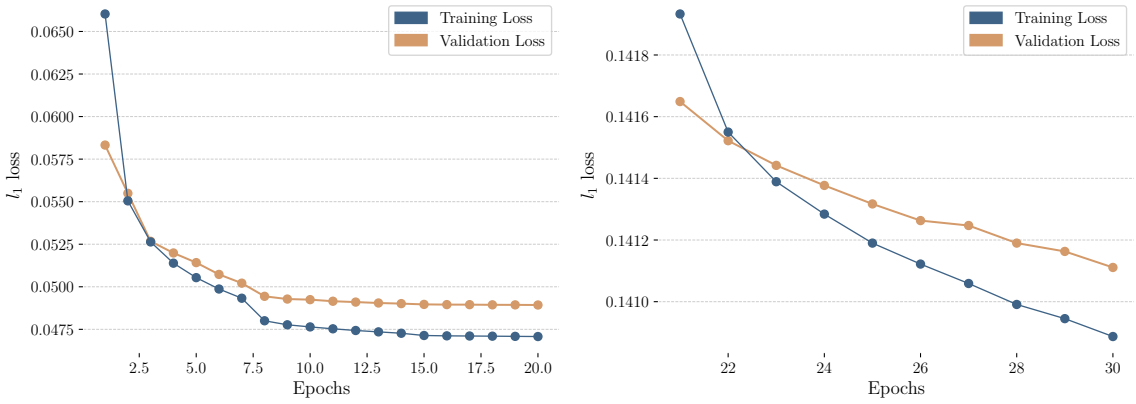


Figure 5.7: Training and validation curves for Res-EnDec model. Left: l_2 loss for 20 epochs and Right: l_1 loss for ten epochs of training.

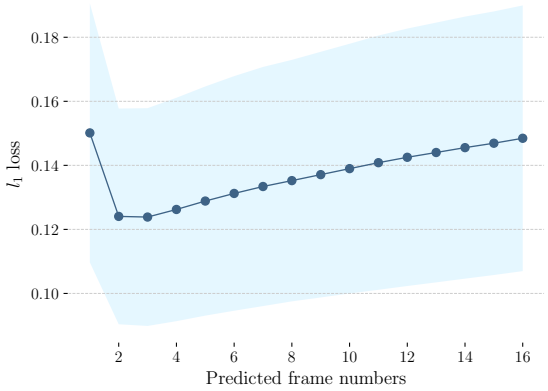


Figure 5.8: TVG for Res-Endec model

By comparison, the TVG for the *Conv3D* model shows a much higher mean prediction error of $(3.13 \pm 0.47) \times 10^{-1}$. The Conv3D model consists of 3D convolutions in the encoder and decoder. The encoder employs striding to reduce image sizes to representations. The decoder on the other hand uses transposed convolutions, referred to as deconvolutions in some literature, to increase image sizes [60]. All kernel sizes are set to $3 \times 3 \times 3$ as is common with 3D convolutional architectures. The trend over time, as can be noted from Figure 5.9 is unstable and the error does not appreciate as much as in the case of the Res-EnDec model.

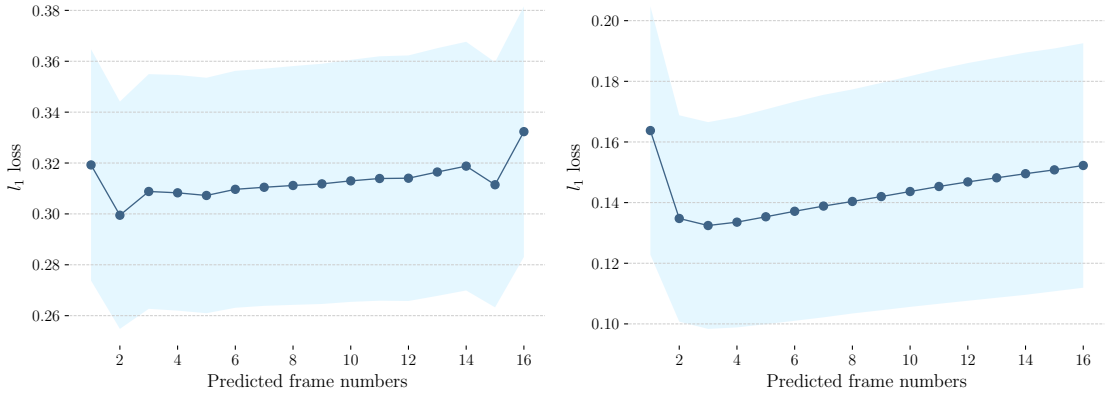


Figure 5.9: Left: TVG for Conv3D model and Right: TVG for Segment model

The sequential nature of the video frames can be better modelled with recurrent layers than with 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 standard vector LSTM layers. The Segment model is designed with kernel sizes and number of filters drawn from the common image segmentation models. The encoder is 3D-convolutional with $3 \times 3 \times 3$ filters as kernels and the number of

filters increases from 32 to 64. The mean error is now $(1.43 \pm 0.36) \times 10^{-1}$, $\sim 54\%$ better than the Conv3D model and $\sim 4\%$ worse than the Res-EnDec model.

Ablation Studies

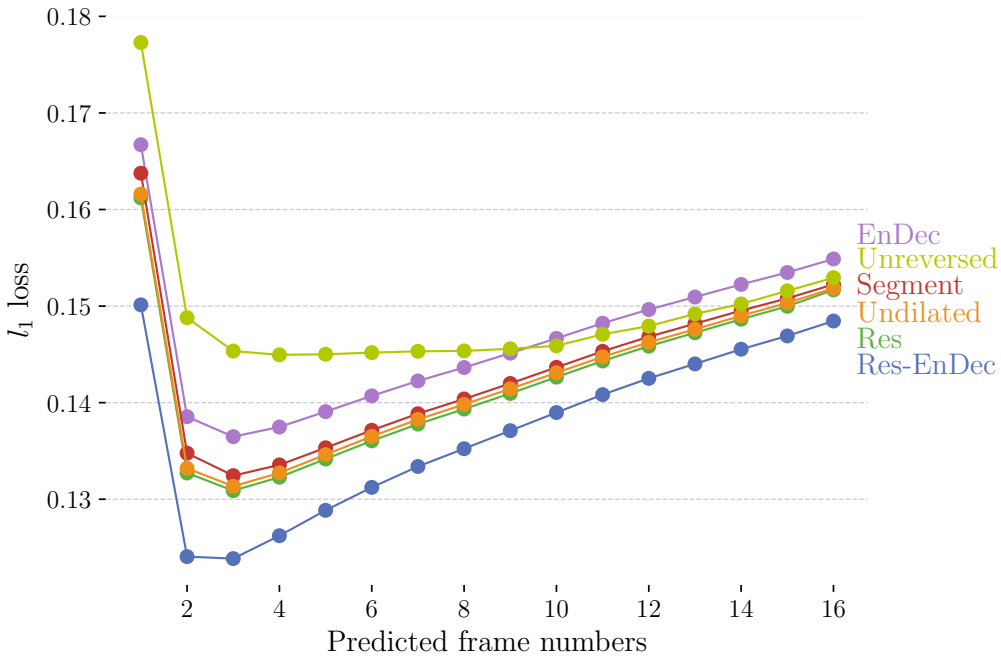


Figure 5.10: Temporal Variation of the l_1 prediction loss for various models. TVG for the *Conv3D* model is not shown to avoid graph skew due to the relatively large error values.

We perform ablation studies on our Res-Endec model to determine the importance of the residual connections, dilated convolutions and reversing the input 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 5.10 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 appreciates across all models over time. We will quantify this appreciation for comparison across all models here. As is evident from the graph some losses increase at a faster rate than others, particularly after the third-frame prediction. We tabulate this rate of increase as a percentage over the third-frame prediction loss in the Table 5.1 as *Appreciation*. Our best performing Res-EnDec model also shows the most error appreciation of 19.86% over the least appreciating Unreserved model at 5.22%. However, the mean loss

for the Res-EnDec model is $\sim 8\%$ lower. For the case of the unreversed T-frame input data making T-frame predictions, we observe that although an empirical distance between frames x_k and x_{T+k} is higher than between x_T and x_{T+1}, the distance remains fairly uniform for short-length sequences. As a result, it can be seen that the Unreserved model makes higher errors, but consistently so. From the Table 5.1, it can be seen that the error for the Conv3D model appreciates lower than most other models from frame 3 onwards, but the error itself is $\sim 56\%$ worse than the best. It is important to note that though the Res-EnDec model shows the highest appreciation, the error is consistently lower than the other models for the duration of 16 frames. All other ablations increase the mean prediction error across frames, with the removal of all residual connections causing the most decline at 6%. This suggests that both dilated convolutions and residual connections have improved loss performance.

5.1.4 Latent Space Visualization

Consider this toy example of a person called Enco whose only job is to describe 16-length traffic videos in English. On the other hand, Deco's job is to paint 16 new images that look like what the future would be after they read these English descriptions. Enco could describe the entire video in one sentence or many. Further we could ask Enco to use a definite number of words in a sentence. It can be argued that multiple sentence descriptions are better as they present more opportunity for including details, given each sentence is different from the other. In our work, as is obvious, Enco is the Encoder, Deco is the Decoder. We choose to represent the encoder's definition of the traffic video as 16 representations $z_{1:16}$, similar to 16 equal length sentences from Enco.

If Enco said, "The red car is driving from the left to the right", 16 times instead of saying new things, Deco would perhaps ignore a pedestrian stepping on to the street in their painting of the future. In these experiments we study how different each of the z are to each other. In our first experiment, we repeat the first learnt representation z_1 across the decoder's input in our Res-EnDec model and show the predicted future in figure 5.11. We continue to repeat the rest of the learnt $z_{2:16}$ in the same fashion and show the predictions to compare their relative contribution. The predictions are presented in Figure 5.11. Repeating z_1 results in virtually no visible motion across predicted frames as seen in the figure. However, repeating the later representations, z_{16} for example, shows the most aberration along the edges suggesting movement in the objects. From these comparisons, it can be inferred that every representation z learnt by the encoder is encouraged to be different from the others. This is because of the recurrent nature of the decoder and the additive framework of residual connections. Our treatment of the future generation stage is that the first few frames offer a transfer of spatial layout from the past to the future. Then, motion variances across frames are imparted to the predictions as multi-stage autoregression over the trailing representations.

In our second experiment, we visualize the $16 \times 26 \times 64$ dimensional representation space z using the t-distributed Stochastic Neighbour Embedding algorithm (t-SNE) [61]. Or in other words, we are reprojecting Enco's 16 sentential descriptions into say, three dimensions to study if and how they are different from each other. If they are different, we can conclude that they are contributing new information to aid Deco in painting the future. Visualizing each z_i, $1 \leq i \leq 16$ directly is clearly inconclusive because they correspond to a variety of traffic videos. We proceed to subtract the first representation, z_1 from each of the rest to retain temporal variances and project them onto a 3-dimensional space in Figure 5.12. We choose z_1 for subtraction from the other representation because we deem it to transfer the most spatial information. We do not plot the now normalized z_1 as they are all zero vectors following the subtraction. For each video in the test set, we gather $z_{1:16}$ that are used to decode the future. We then randomly choose 625 instances for each of the 16 representations and plot their t-sne minimized vectors in Figure 5.12 and 5.13. The slices going from time $t = 2$ to $t = 16$ are shown using a spectrum variation in colour with the legend on the right edge of every graph. The scatter plot shows a trend going from deep blue to maroon, equivalently from the second predicted frame to the last. The clustering of like colours indicates a similarity in decoding a given time slice of prediction. It is interesting to note that the representations appear to transition sequentially, suggesting *Iterative Inference* as discussed in Section 4.1.

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 5.14. The plot shows many 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 pseudo-spherical nature of the volume is clearer in 2D form in Figure 5.15. The effect of the recurrent and residual connections together is that of *Iterative Inference*, as intuited.

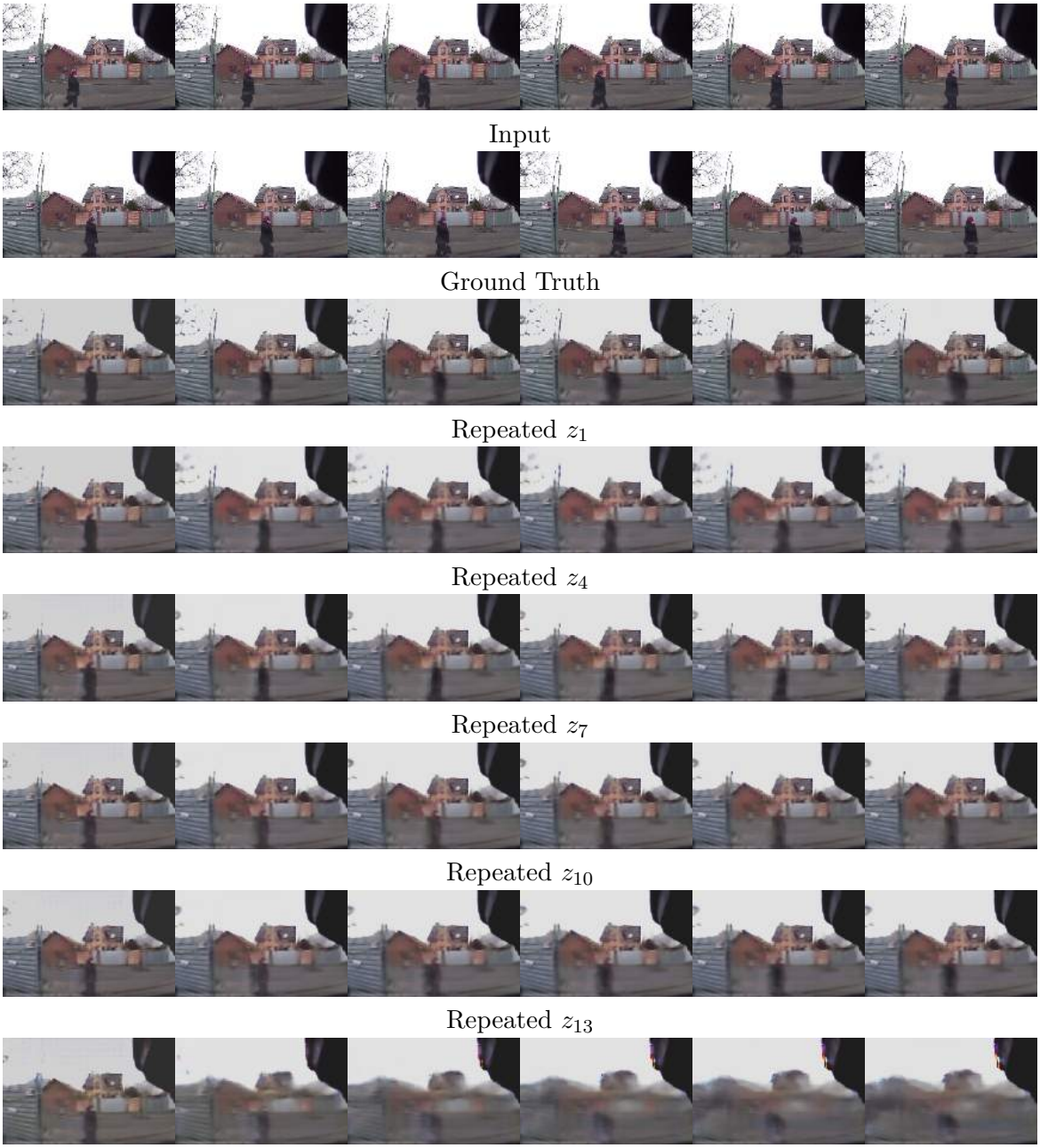


Figure 5.11: Studying the effect of sequential z

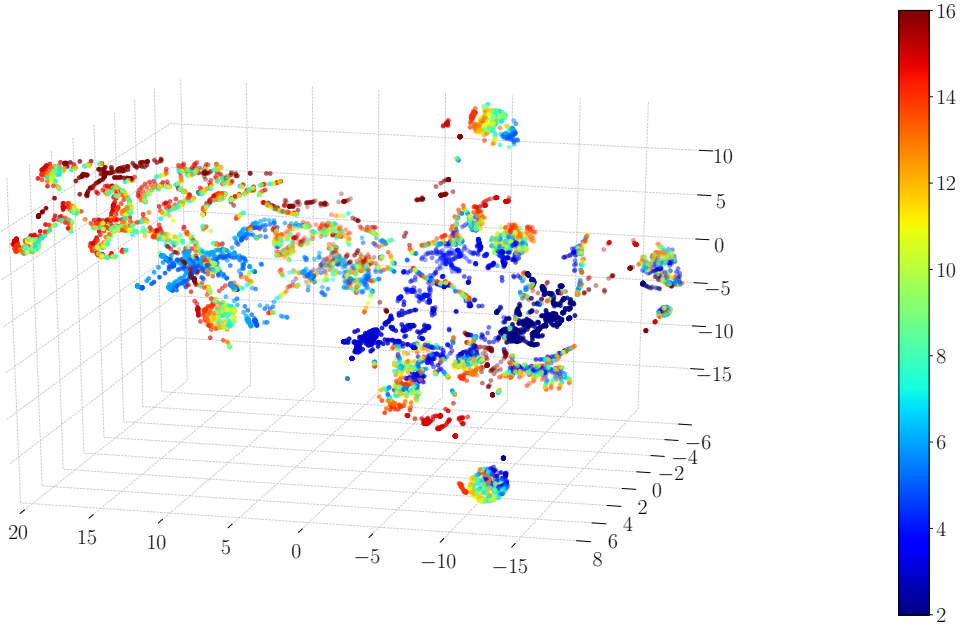


Figure 5.12: Scatter plot of 15 representations $z_{2:16}$ learnt by the encoder. 625 instances for each of the 15 tensors have been reduced to 3 dimensions using the t-sne algorithm. The reprojected tensors are shown as a colour gradient from 2 to 16. A gradual trend from deep blue to maroon suggests that the representations are different from each other. Autoregressively processing them encourages iterative inference.

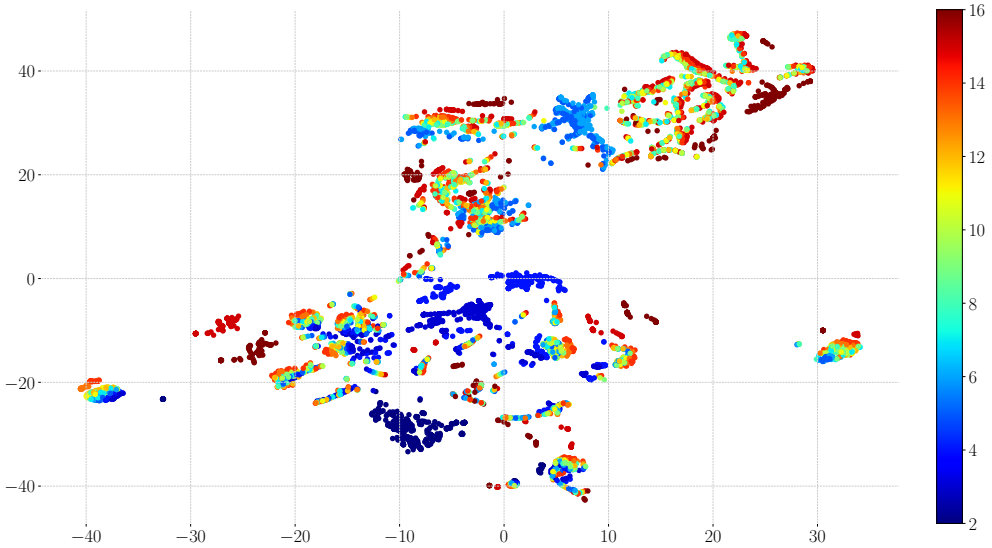


Figure 5.13: Scatter plot of 15 representations $z_{2:16}$ reduced to 2 dimensions using the t-sne algorithm. The reprojected tensors are shown as a colour gradient from 2 to 16.

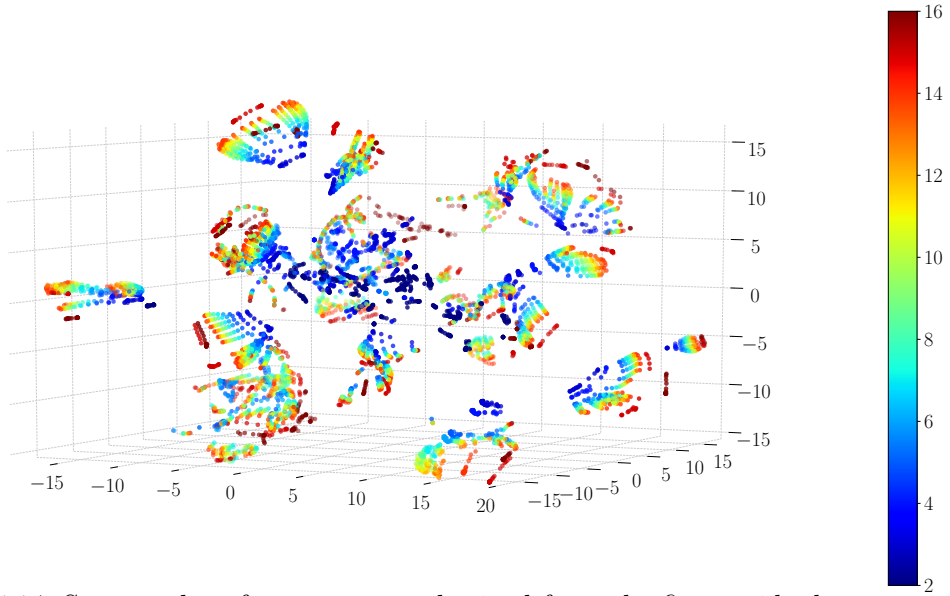


Figure 5.14: Scatter plot of tensors $r_{2:16}$ obtained from the first residual connection in the decoder. 500 instances for each of the 15 tensors have been reduced to 3 dimensions. The reprojected tensors are shown as a colour gradient from 2 to 16. Close association of samples in the form of blue-maroon streaks suggests that the recurrent convolutions assist in reforming the latent space into a virtual sphere with motion projecting the transformations outwards. The distinction between the samples suggests that the residual connections encourage mutually different representations and encourage iterative refinement.

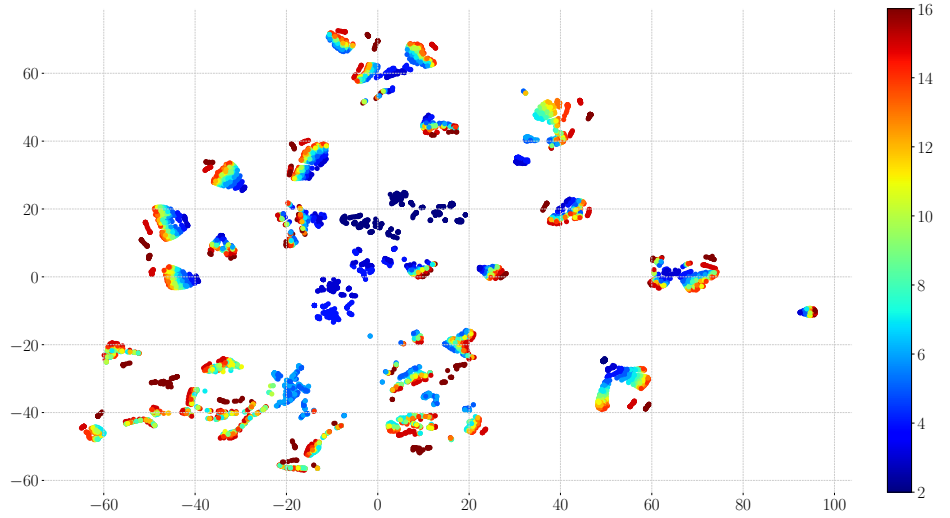


Figure 5.15: 2D scatter plot of 500 instances each for the 15-length tensor sequence $r_{2:16}$ output at the first residual connection in the decoder. The representations are shown as a colour gradient from 2 to 16.

5.1.5 Discussion

A significant challenge to generalization is posed by high variability in how a traffic activity transpires over time. This work is motivated by an observation of human behaviour that short-term predictions of their immediate surroundings are based based on prior experiences. A prediction of the future aids humans in taking more-informed control decisions. In this section we discuss some observations we make in our video prediction experiments.

Working in the image space helps the model understand motion as a scene phenomenon. This results in implicit modelling of individuals' motion, relaxing the need for expensive annotations needed to supervise a network. Multiple object and pedestrian motions, even in orthogonal directions, are also tractable, as seen in Figure 5.16(a). However, such a scene interpretation leads to cases where the largest moving object inadvertently dominates the motion transform. Figure 5.16(b) shows a case where the relatively large motion content of the car impedes the network's ability to perceive the pedestrians in the background.

(a)

(b)

Figure 5.16: The effects of learning motion as a scene phenomenon. (a) Multiple object and pedestrian motions, even in orthogonal directions, are tractable. (b) The relatively large motion content of the car impedes the network's ability to perceive the pedestrians in the background.

Object motions, unseen in the input frames are not predicted in the future. Although this is intuitive, the 16-frame window our model looks at the world through makes it impossible to generate an otherwise periodic motion. Figure 5.17 shows the periodic motion of a windshield wiper in row 1 corresponding to the past, a window prior to the current input

to the model, which itself is shown in row 2. The model does not reproduce this motion as seen in row 3, the predicted sequence.

The approach of data-driven modelling makes it possible to build more generalizable models than ones with hand-crafted features. However, a bias in sampling the training data could lead to flawed predictions for previously unseen or less frequently seen cases. Figure 5.18 shows one such case. Figure 5.18 (a), row 1, is the ground truth video of a car backing up onto the street. The generated future, row 2, predicts that the car is stationary. We suspect this could be because of how rarely such a scenario is encountered in our training data. Ground truth set 5.18 (b) shows the car now showing a movement laterally to the right in each frame while backing up. As this motion is very common, lateral variations in the car-shape can be seen in the prediction, while still failing to embody the car's reverse travel.

Figure 5.19 depicts a test sequence of pedestrians moving forward in the direction of the camera's travel. It can be seen that the predictions (row 2), represent visually inconspicuous movement in the pedestrians, although the car's egomotion borne transitions are reproduced. We believe that these instances can be better approached by considering motion as a three-dimensional phenomenon. Inclusion of depth supplements the model with the needed information to reason longitudinal motion from.

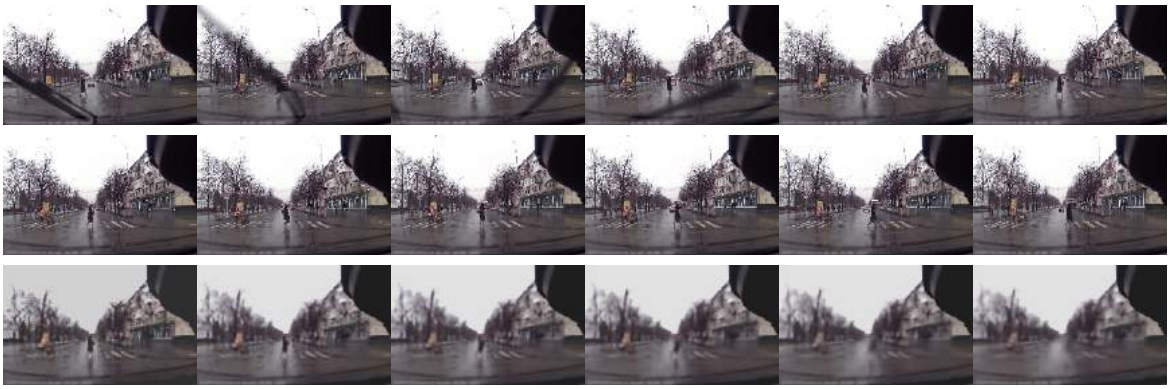


Figure 5.17: Row 1: The past, a window prior to the current input to the model; Row 2: Current input to the model; Row 3: The predicted sequence. The model does not reproduce the periodic motion of the windshield wiper even though it has seen it in an earlier input.

(*a*)

(*b*)

Figure 5.18: Training data contains very few examples of a car backing up on to a street, making such predictions inaccurate.

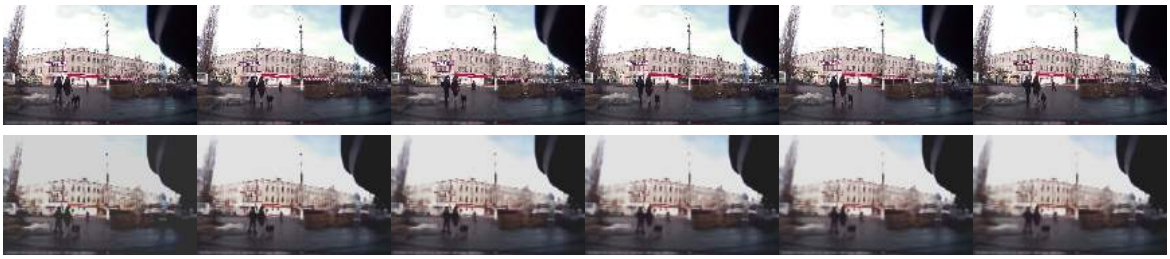


Figure 5.19: Non-lateral motion cannot be predicted accurately. Such instances can be better approached by considering motion in 3-dimensions.

5.2 Crossing Intent

5.2.1 Experiments and Results

We fine tune the classifier model described in Section 4.4 on the JAAD dataset. Our model is trained on the similar data train, validation and test splits as described in [1] so that we can compare with them in terms of Average Precision (AP) for classifying a video into crossing or not crossing. Their *Action + Context* model has an AP of 62.73% as shown in Table 5.2 for classifying 10-15 frames of video. Details of this work are presented in section 2.2.1. We organise training videos in strides of 8 and test videos in strides of 16. We also augment image sequences with every alternate frame chosen to train on, to simulate faster moving traffic. We use *RMSprop* as the optimizer starting with an initial learning rate of 10^{-5} for 30 epochs. 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. The most frequent label in a sequence is considered as the target label for binary classification. Although our usage of l_1 loss over l_2 renders sharper edges in the images, we pose that these losses do not evaluate the compositional consistency of the predictions. Recognizing an action using a classifier network, we can reason about motion and visual content in the video by building hierarchical representations.

Model	AP
Action [1]	39.24 ± 16.23
Action + Context [1]	$\mathbf{62.73 \pm 13.16}$
Conv3D	78.61
Segment	80.85
Res	80.42
Res-EnDec	**81.14**

Table 5.2: Average precision in predicting crossing intent

Model	Accuracy
Conv3D	61.18
Segment	65.23
Res	66.59
Res-EnDec	**67.38**

Table 5.3: Crossing intent prediction accuracy across various models

We fine tune our classifier model for 30 epochs with early stopping based on validation loss and on the futures predicted by the models listed in the Table 5.3 as input. 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 [1], in the Table 5.2. We outperform their Action+Context model consistently across all our models and by about 18% in the case of our best results. Figures 5.20 - 5.28 present some video sequences with predicted and ground truth labels on the bottom left corner of each frame. The image descriptions underneath provide more observations from the sequences.

Model	AP	Precision	Recall	F_1 score	Accuracy
Conv3D	69.91	**78.26**	39.13	52.17	52.17
Segment	**72.99**	67.86	41.30	51.35	47.83
Res	71.09	67.74	45.65	54.55	49.28
Res-EnDec	72.56	68.29	**60.87**	**64.37**	**55.07**

Table 5.4: Crossing intent prediction statistics across various models for videos with an action change between input and ground truth.

Model	AP	Precision	Recall	F_1 score	Accuracy	MTCP
Conv3D	74.86	**92.86**	63.93	75.73	76.85	$\mathbf{1.52 \pm 1.85}$
Segment	76.92	91.49	70.49	79.63	79.63	2.06 ± 2.87
Res	77.82	86.89	80.30	83.46	80.56	2.37 ± 3.48
Res-EnDec	**81.93**	89.61	**89.61**	**89.61**	**86.09**	2.01 ± 2.83

Table 5.5: The performance of various models from MTCP experiments employing sliding for test cases where the ground truth labels change from *crossing* to *not crossing* and vice-versa.

In a second experiment, we only consider those test cases where there is a change in action labels between the current and the future ground truth frames. The ground truth data is labelled with actions per frame. We consider the most frequently occurring label in our input sequence as the current action label and in the ground truth future as our target label for this experiment. We report several metrics for this experiment in the Table 5.4. In some test scenarios, we observe that incorrect action predictions may arise because of insufficient stimulus in the input to feature a change in the future transformation. We define a new metric, mean *time-to-correct-prediction* (MTCP), as a measure of the number of new frames to be subsequently processed to correct a mistaken crossing intention. As another experiment, we slide our input window of 16 frames by every frame once we determine there is change in a pedestrian's crossing intent by looking at the ground truth. We do this until the classifier predicts correctly. Therefore, the maximum MTCP is 16 beyond which sliding would effectively mean seeing the future, and we consider this as a misprediction. We report the MTCP for different models in the Table 5.5 along with other binary classification metrics to compare with those from the Table 5.4. The MTCP score is a means of comparing an

empirical latency in predicting a different behaviour, triggered by an input feature such as a stepping forward cue by a crossing pedestrian. However, the score should not be taken to imply that by waiting to process new frames all initial mispredictions can be corrected. The Conv3D model, with the most visually obscure predicted images, has the highest precision at 92.86% when sliding is employed, but suffers with poor recall with and without sliding. The model also shows the lowest MTCP at 1.56, meaning that it responds to input changes faster than the other models experimented with. We suspect this could be because of fully convolutional nature of the model. All other models experimented with feature a recurrent variant of the decoder and sport a similar MTCP of around 2.20. The Res-EnDec model performs about 10% better than the Conv3D in terms of prediction accuracy and reports around 10% better average precision for our MTCP experiments. Also notable is the high recall score at 89.61% for this model with sliding, giving it the best F_1 score.

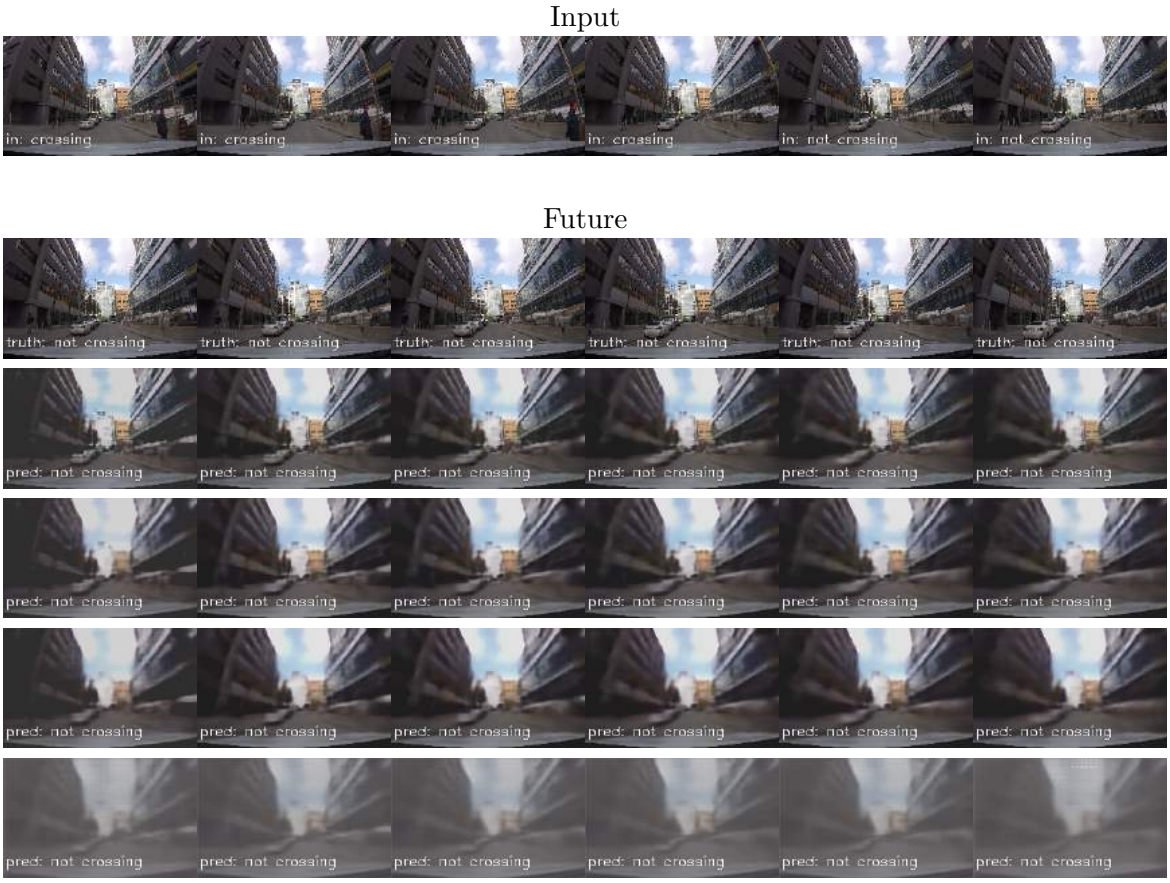


Figure 5.20: Example predictions by various models. Every third frame shown for brevity. Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: Segment model; Row 6: Conv3D model. The pedestrian exits the frame towards the right going from crossing to not-crossing.

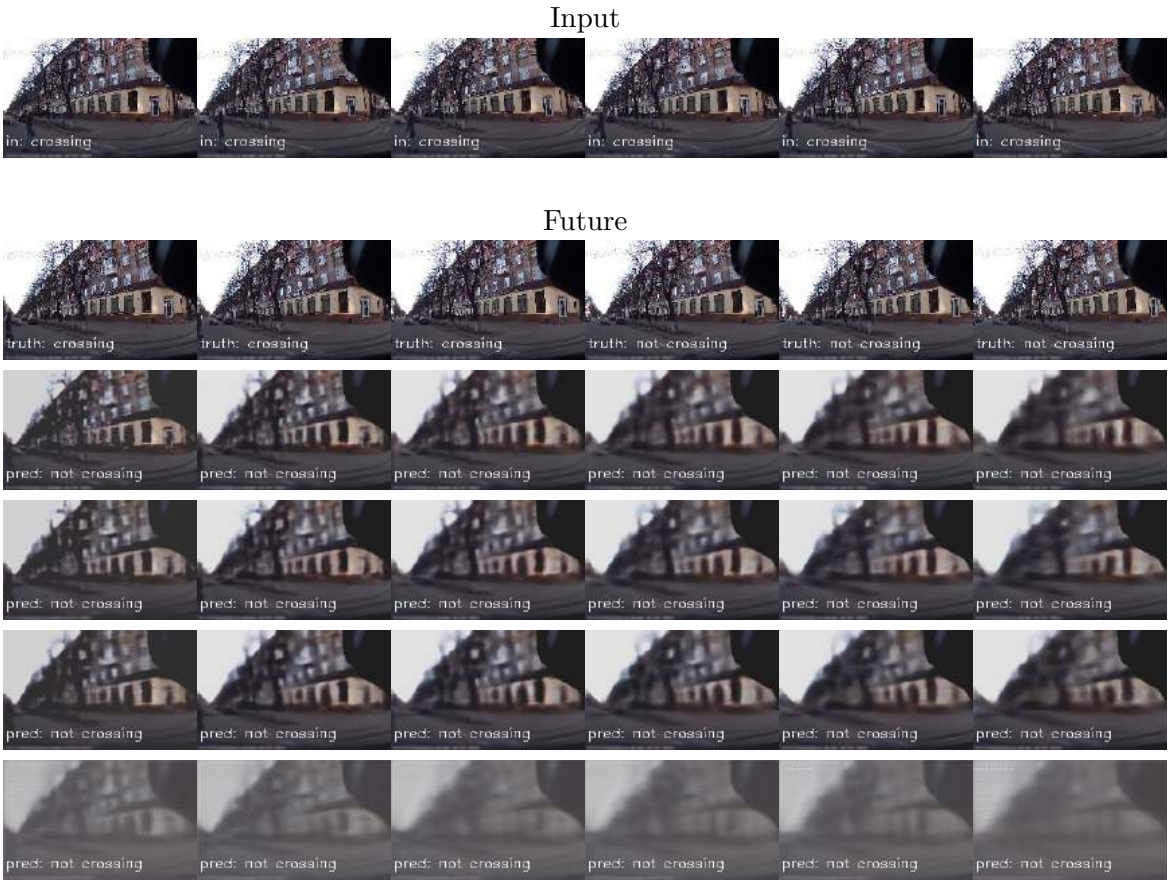


Figure 5.21: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: Segment model; Row 6: Conv3D model. The pedestrian is seen to be exiting the view frame towards the left with some partial presence in the first few frames of the future. Action transition is from crossing to not-crossing.

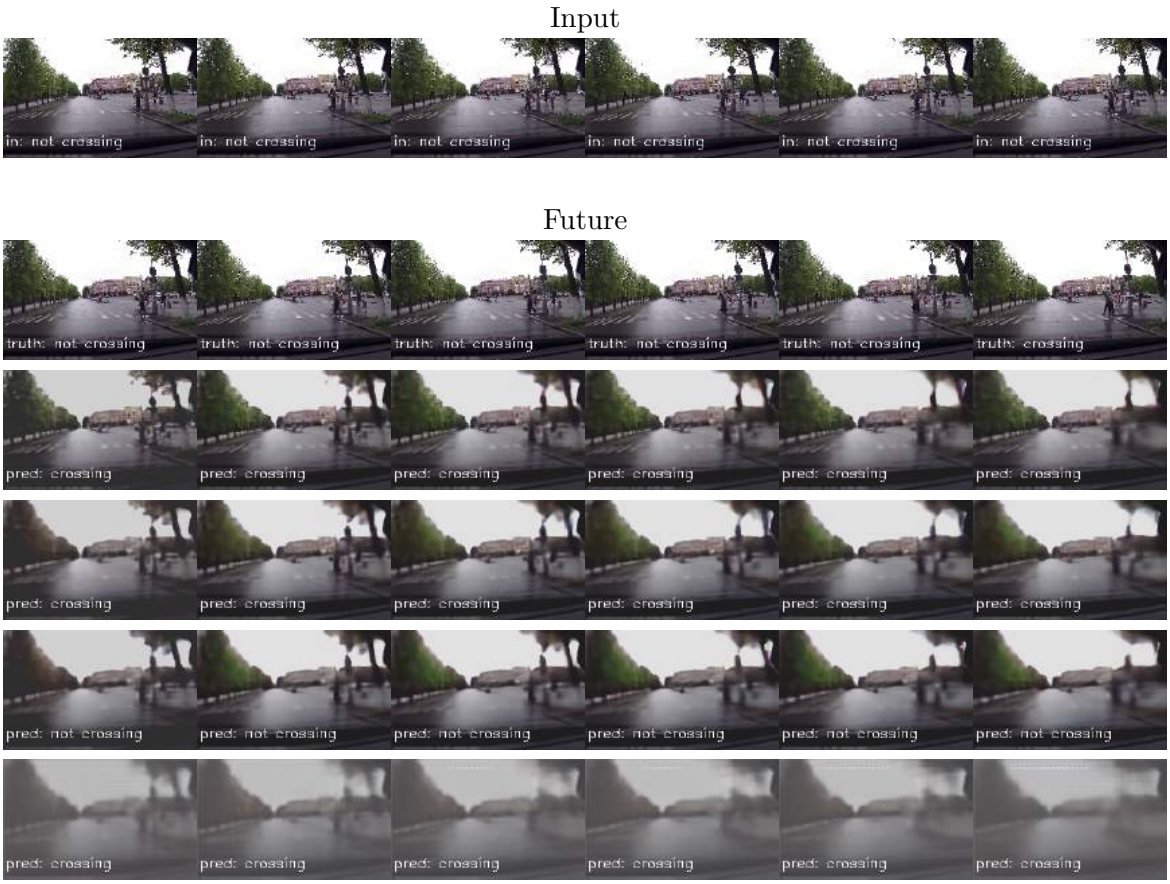


Figure 5.22: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: Segment model; Row 6: Conv3D model. A pedestrian is attempting to cross the street with action transition from not crossing to crossing. Although, from the ground truth, a crossing label is seen only in the last frame, Res-EnDec, Res and Conv3D models are able to correctly identify the action for this example.

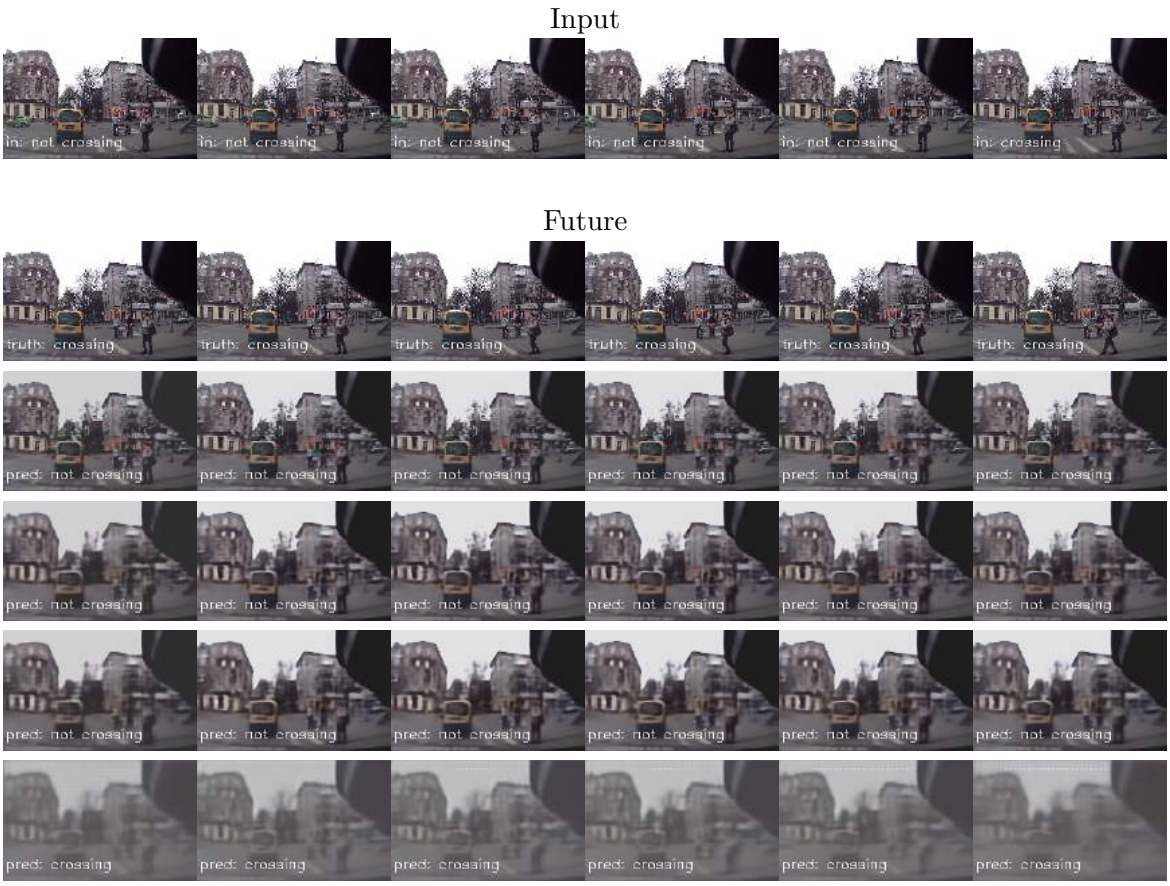


Figure 5.23: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: Segment model; Row 6: Conv3D model. A pedestrian can be seen to be attempting to cross the street towards the left going from not-crossing to crossing. All models except Conv3D mis-predict this change. We believe this is because of seemingly inconspicuous movement from the pedestrian indicating an intent to cross until the very last frame of the input sequence. Reading a few more frames could potentially help the models to correct their prediction.

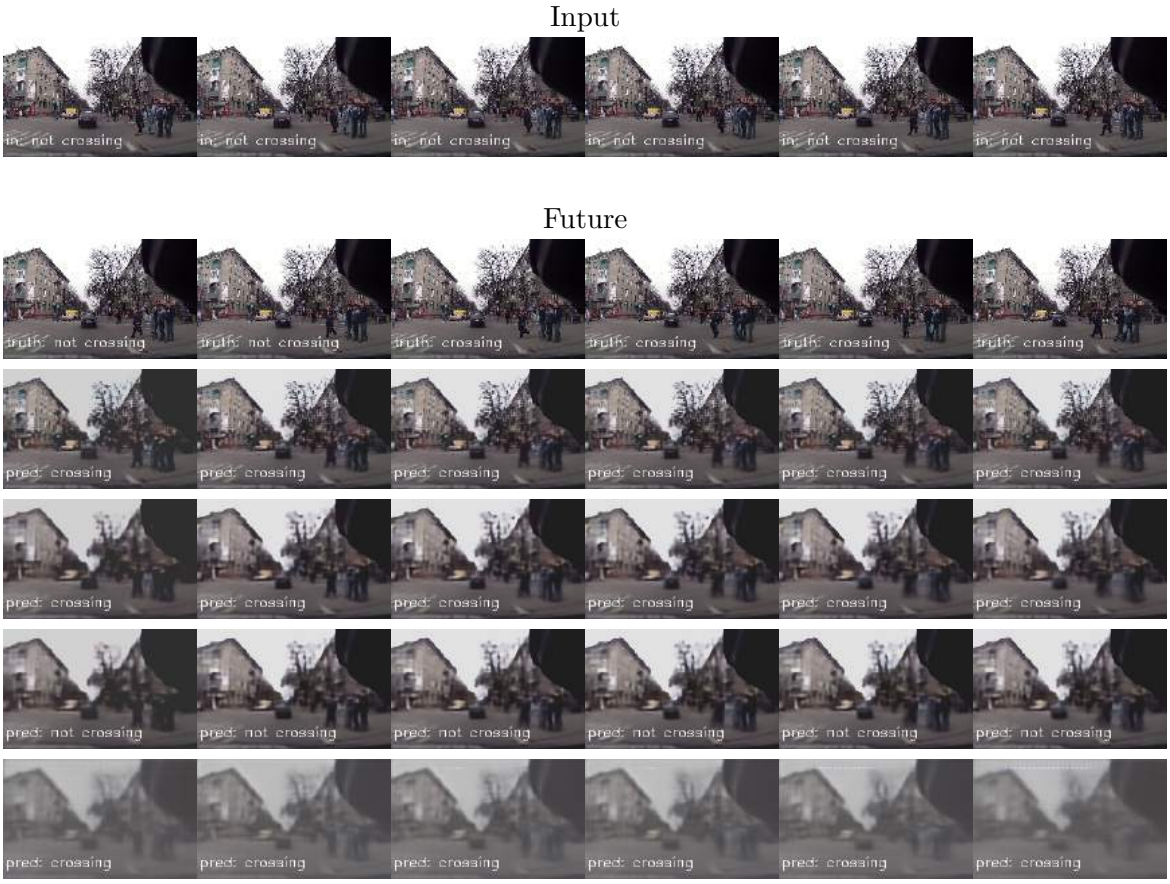


Figure 5.24: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: Segment model; Row 6: Conv3D model. A group of pedestrians can be seen attempting to cross the street towards the left with the foremost pedestrian stepping forward first. All models except the Segment model predict the change correctly.

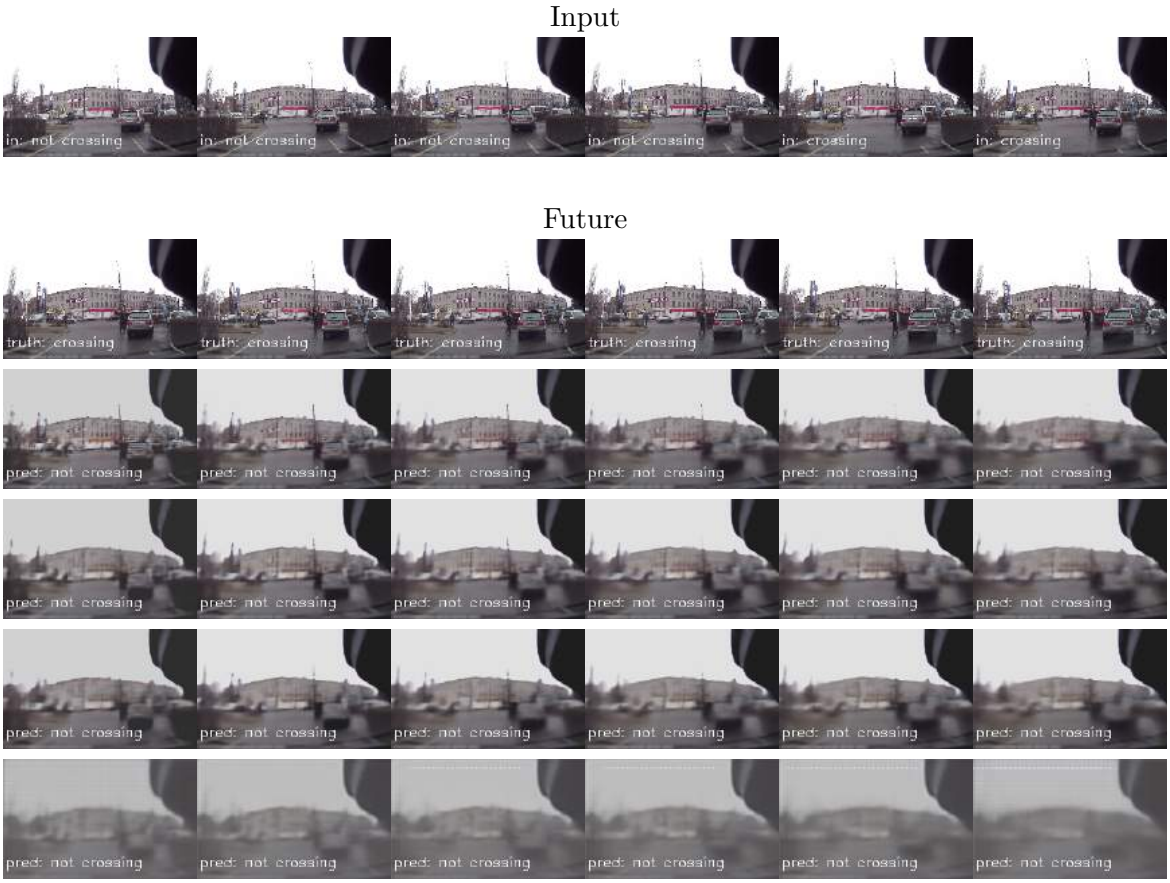


Figure 5.25: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: Segment model; Row 6: Conv3D model. A pedestrian can be seen to cross the intersection from behind the car ahead from the ground truth sequence. All models are unable to capture this movement as can be seen by the stationary generations for the pedestrian in the future.

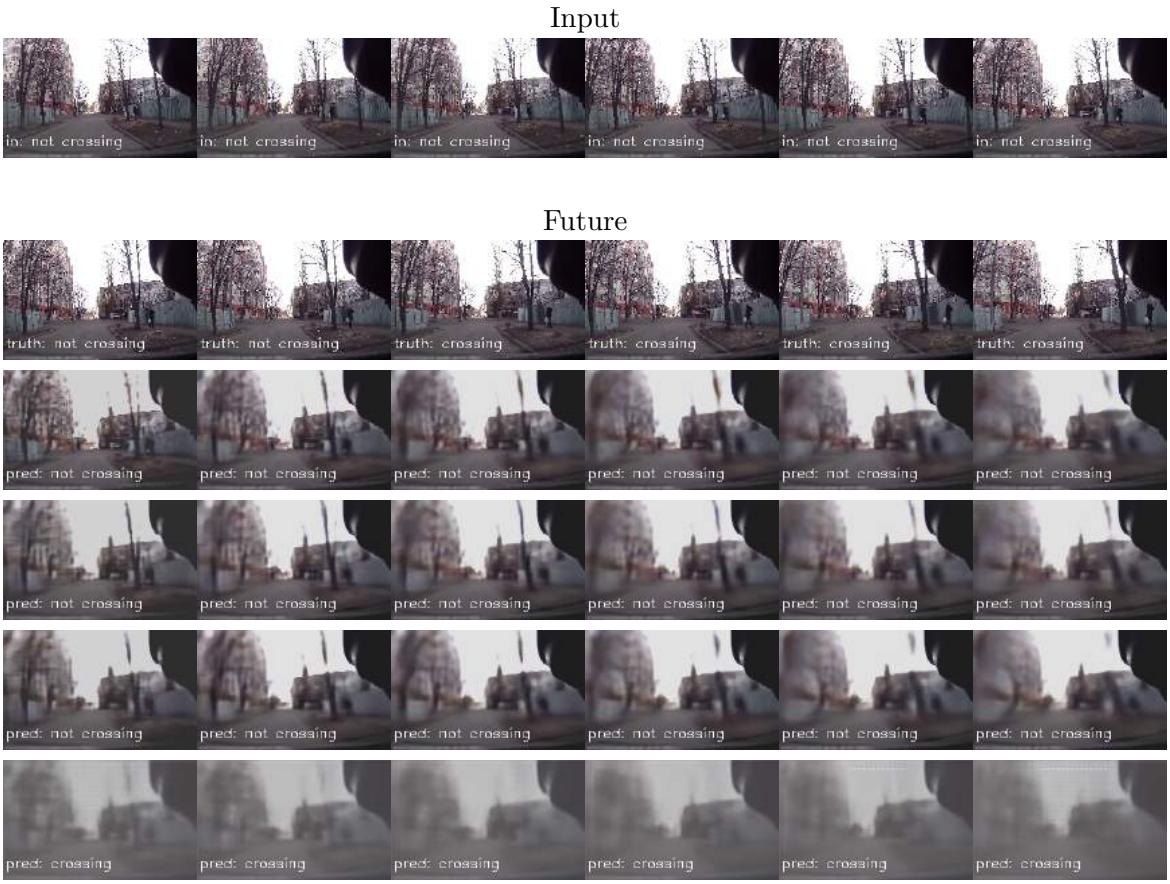


Figure 5.26: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: Segment model; Row 6: Conv3D model. A pedestrian is seen walking longitudinally along the road not appearing to cross. Ambiguous labelling in such scenarios is a challenge.

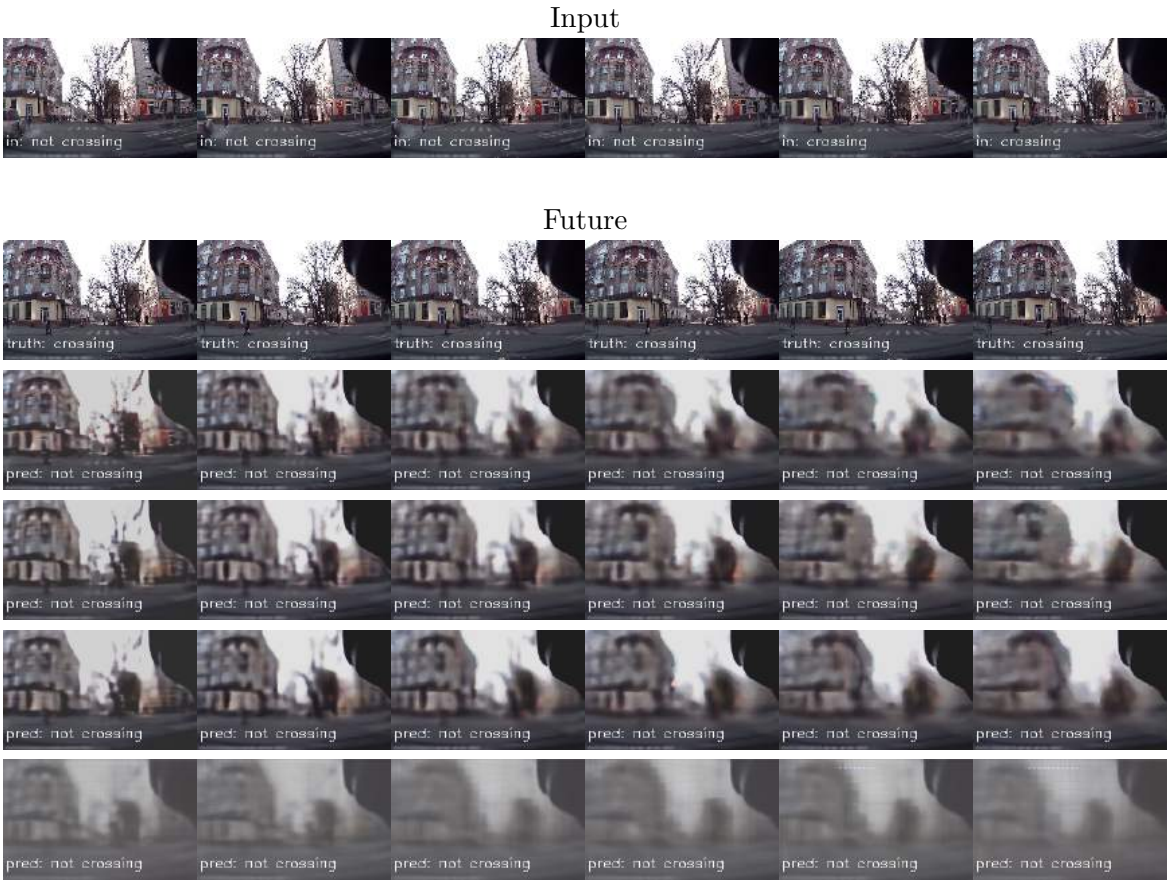


Figure 5.27: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: Segment model; Row 6: Conv3D model. No model succeeds in correctly predicting the action for the pedestrian crossing longitudinally in the left region of the frame. We believe this is because this pedestrian, as can be seen from the input and the ground truth, features only in the future. The models cannot predict motion for unseen participants.

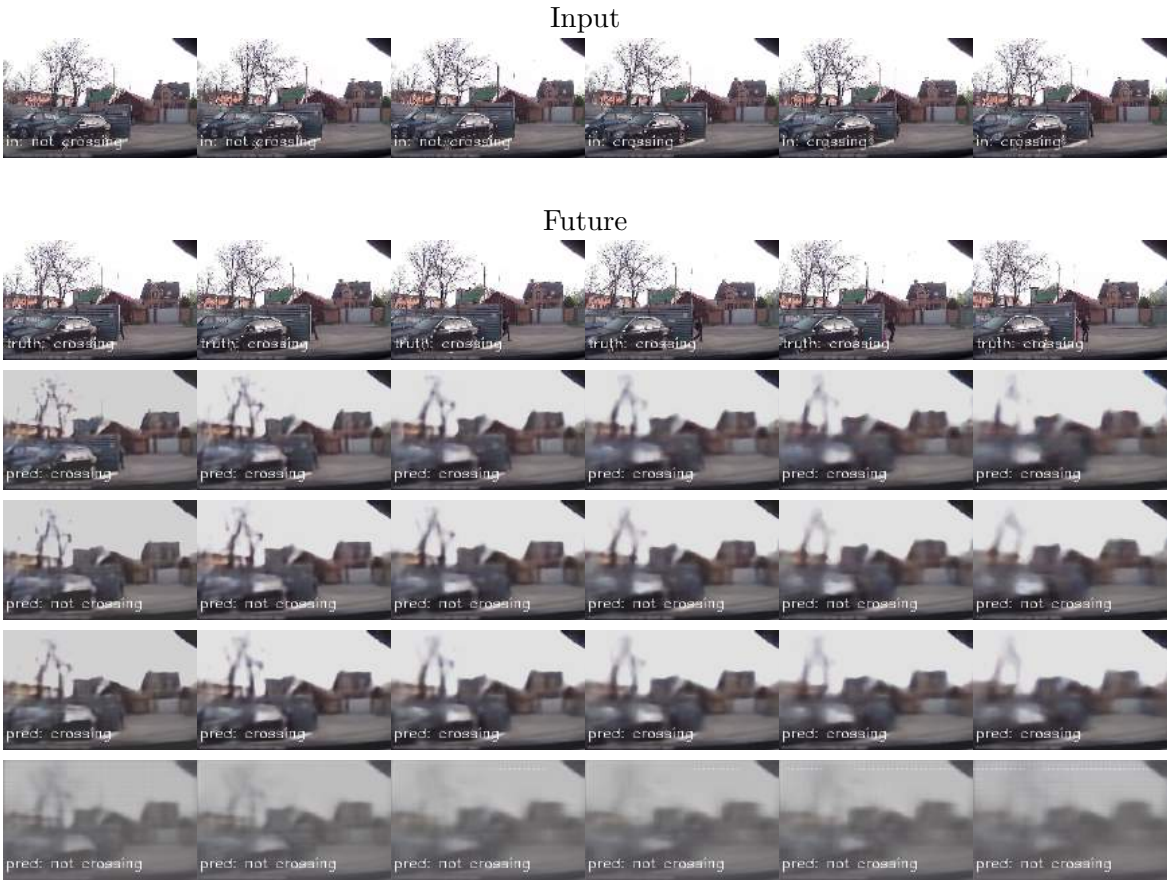


Figure 5.28: Row 1: Input frames; Row 2: Ground Truth. Futures predicted by, Row 3: Res-EnDec model; Row 4: Res model; Row 5: Segment model; Row 6: Conv3D model. An example where the car's own egomotion is in a direction non-orthogonal to that of the crossing pedestrian. Res-EnDec and Segment models correctly predict the crossing action.

5.2.2 Discussion

Prediction of crossing intent in traffic scenarios has the potential to save lives. Conversely, braking every time a pedestrian is on the curbside could make the drive intermittent and uncomfortable. Reasoning from a probable future affords a stochastic estimate of a pedestrian's crossing intent, reducing the chances of unnecessary braking. Such a system could tolerate false positives, but false negatives need to be penalized. Or, in other words, a good precision is desired but a good recall is necessary. From the Table 5.6, we can see that the Res-EnDec model has comparable precision and recall scores on our test set. The Conv3D model has a precision ∼10% higher than the Res-EnDec, but has the lowest recall. The F_1 scores better summarize this trend.

Model	Precision	Recall	F_1 score
Conv3D	**83.75**	46.74	60.00
Segment	81.80	56.83	67.07
Res	79.23	62.84	70.09
Res-EnDec	74.77	**71.90**	**73.31**

Table 5.6: Crossing intent prediction accuracy across various models

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 Table 5.7 lists the run time to load a 16-length image, predict the next 16 frames and classify the video 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 463ms. Res-EnDec has a runtime of $\sim$ 117 ms, with a similar time for the Res model and an effective look-ahead of 416ms.

Model	Time (ms)
Conv3D	68.84 ± 26.34
Segment	96.10 ± 28.89
Res	116.11 ± 42.22
Res-EnDec	116.39 ± 38.78

Table 5.7: Time taken to recognize crossing intent. The time estimated is from the instant the input frames are fed to the future generator to the instant the classifier makes a prediction.

Chapter 6

Conclusion

6.1 Video Prediction

In this thesis, we demonstrated three broad categories of neural network algorithms tasked with generating future videos. All models followed an encoder-decoder strategy. The three categories are: a) fully convolutional, b) recurrent decoder c) residual network. We intuited that time dilated causal convolutions can help an encoder look deeper in time for the same number of parameters for the purpose of tackling potentially negligible temporal variation in subsequent frames. Ablation studies were also performed by unreversing the input frame sequence, undilating convolutional kernels, removing residual connections between the encoder and decoder and the ones within the encoder and decoder for the Res-EnDec model, producing the lowest l_1 error between predictions and ground truth future. We then introduced a *Temporal Variation Graph* for all models, to measure their contributions in per-frame visual reproducibility and temporal coherence. Our results suggest that the residual connections encourage learnt intermediate representations to be mutually different and along with multi-stage recurrent decoding, iterative inference can be seen.

Our insight was learning a sequence of representations from an encoder rather than a comprehensive tensor as done in many sequence generation approaches. We do this with the understanding that motion is nuanced and distributed throughout direct and indirect participants in a traffic interaction. A vehicle's own egomotion is hard but essential to model in conjunction with the scene. From the t-sne plots in Section 5.1.4 we see that an autoregressive consumption of learnt representations makes the latent space organize same frame vectors into a cloud almost distinct from one another. Recurrent decoding with residual learning further structures this latent space suggesting a transformation that iterates the representations to the future one at a time, validating our hypothesis.

6.2 Reasoning from a future

We also proposed and demonstrated a classifier algorithm based on the C3D sports action classifier model presented by Tran et al. [6]. The network was tasked with recognizing a crossing action by looking at a video of a generated future, thereby being able to predict a pedestrian's crossing intent. We showed that our best model with an average precision score of 81.14% is 18% higher than the model introduced by Rasouli et al. [1] for videos from the JAAD Dataset [9]. Their model also studies the contribution of context in the overall gain in performance by explicitly assigning the network the additional job of detecting scene elements such as traffic signs against our implicit modelling of motion as a scene phenomenon. For the specific instances of a changing action from the input to the future, our experiments with the Res-EnDec model result in the best accuracy in prediction at 88.74% with an F1 score of 90.81%. We then introduced a new metric called *mean time to correct prediction* as a measure of the average number of new frames that lead to a correct prediction in the case of a changing intention in crossing. It also serves as a measure of the future generator's responsiveness to variations in the input that trigger a change in the action generated. The best MTCP was that of 1.56 frames with a standard deviation of 2.12 frames for the Conv3D model, but the model suffered in all other metrics. All other models report a similar MTCP of 2.12 frames. In the discussion we described the necessity for such a classifier to have a good recall because false negatives invoke severe penalty. A good precision is desired to reduce intermittent braking caused by false positives. This is likely to result in a smoother and more comfortable drive for the passengers.

6.3 Unification

The idea of looking into the future to predict a potential hazardous crossing action is only validated as long the computational processing time is fast enough to take preventative action. We reported that the fastest model Conv3D can look-ahead by about $463ms$ and our best performing model, Res-EnDec has an effective look-ahead of $416ms$. Data driven methods, by extracting features from videos as a learning optimization of weights, help in establishing a prior on motion from challenging traffic scenes. However some cases, as discussed in Section 5.1.5, showcase a few shortcomings. We also present potential improvements by studying motion in 3-dimensions. Another avenue to consider could be fine-tuning the future generator while freezing the weights of a satisfactorily trained classifier so that only the features responsible for action recognition from the scenes can be refined.

Our hypothesis is that an intelligence predicting, preparing, anticipating, expecting or prospecting the future, will augment motor control and decision making in humans. This feature is critical to combat an artificial indifference with which we inherently build our autonomous cars. For they will soon start to live amongst us.

Bibliography

[1] 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)*. IEEE, Oct 2017, pp. 206–213.

[2] *See-Think-Do, Learn to drive smart: Your guide to driving safely*, Insurance Corporation of British Columbia, 2015. [Online]. Available: http://www.icbc.com/driver-licensing/documents/driver-full.pdf

[3] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Agreeing to Cross: How Drivers and Pedestrians Communicate," in *arXiv preprint (CoRR)*, vol. abs/1702.03555, 2017. [Online]. Available: http://arxiv.org/abs/1702.03555

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *arXiv preprint (CoRR)*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[5] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in *arXiv preprint (CoRR)*, vol. abs/1506.04214, 2015. [Online]. Available: http://arxiv.org/abs/1506.04214

[6] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: Generic Features for Video Analysis," in *arXiv preprint (CoRR)*, vol. abs/1412.0767, 2014. [Online]. Available: http://arxiv.org/abs/1412.0767

[7] The State of Queensland 1995-2018, *Stopping distances on wet and dry roads*, 14 Nov 2016 (accessed June 19, 2018). [Online]. Available: https://www.qld.gov.au/transport/safety/road-safety/driving-safely/stopping-distances/graph

[8] *Light Vehicle Brake Systems (Standard 135)*, Transport Canada, Feb 2015, rev 3. [Online]. Available: https://www.tc.gc.ca/media/documents/roadsafety/135_TSD_rev_3.pdf

[9] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Joint Attention in Autonomous Driving (JAAD)," in *arXiv preprint (CoRR)*, vol. abs/1609.04741, 2016. [Online]. Available: http://arxiv.org/abs/1609.04741

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision Meets Robotics: The KITTI Dataset," in *International Journal of Robotics Research*, vol. 32, no. 11. Thousand Oaks, CA, USA: Sage Publications, Inc., Sept. 2013, pp. 1231–1237. [Online]. Available: http://dx.doi.org/10.1177/0278364913491297

[11] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2009, pp. 304–311.

[12] K. Fragkiadaki, W. Zhang, G. Zhang, and J. Shi, "Two-granularity Tracking: Mediating Trajectory and Detection Graphs for Tracking Under Occlusions," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, ser. ECCV'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 552–565. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33715-4_40

[13] 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.

[14] S. Schmidt and B. Färber, "Pedestrians at the kerb – Recognising the action intentions of humans," in *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 12, no. 4. Elsevier BV, jul 2009, pp. 300–310. [Online]. Available: https://doi.org/10.1016%2Fj.trf.2009.02.003

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

[16] E. Ohn-Bar and M. M. Trivedi, "Looking at Humans in the Age of Self-Driving and Highly Automated Vehicles," in *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1. IEEE, March 2016, pp. 90–104.

[17] 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)*. IEEE, June 2015, pp. 35–43.

[18] 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*. IEEE, Aug 2014, pp. 660–665.

[19] 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*. IEEE, June 2014, pp. 1–6.

[20] 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)*. IEEE, June 2016, pp. 1224–1231.

[21] 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)*. IEEE Press, Oct 2014, pp. 988–994.

[22] C. Tran, A. Doshi, and M. M. Trivedi, "Modeling and Prediction of Driver Behavior by Foot Gesture Analysis," in *Computer Vision and Image Understanding*, vol. 116, no. 3. New York, NY, USA: Elsevier Science Inc., Mar. 2012, pp. 435–445.

[23] S. Koehler, M. Goldhammer, S. Bauer, S. Zecha, K. Doll, U. Brunsmann, and K. Dietmayer, "Stationary Detection of the Pedestrian?s Intention at Intersections," in *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 4. IEEE, winter 2013, pp. 87–99.

[24] C. G. Keller and D. M. Gavrila, "Will the Pedestrian Cross? A Study on Pedestrian Path Prediction," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2. IEEE, April 2014, pp. 494–506.

[25] R. Quintero, J. Almeida, D. F. Llorca, and M. A. Sotelo, "Pedestrian path prediction using body language traits," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, June 2014, pp. 317–323.

[26] A. Robicquet, A. Alahi, A. Sadeghian, B. Anenberg, J. Doherty, E. Wu, and S. Savarese, "Forecasting social navigation in crowded complex scenes," in *arXiv preprint (CoRR)*, vol. abs/1601.00998, 2016. [Online]. Available: http://arxiv.org/abs/1601.00998

[27] W. Choi, K. Shahid, and S. Savarese, "What are they doing? : Collective activity classification using spatio-temporal relationship among people," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, Sept 2009, pp. 1282–1289.

[28] R. Bellman, "A Markovian Decision Process," in *Journal of Mathematics and Mechanics*, vol. 6, no. 5. Indiana University Mathematics Department, 1957, pp. 679–684. [Online]. Available: http://www.jstor.org/stable/24900506

[29] 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)*. IEEE, March 2008, pp. 367–374.

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

[31] 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

[32] 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)*. IEEE, May 2016, pp. 2543–2549.

[33] C. E. Rasmussen, *Gaussian Processes in Machine Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 63–71. [Online]. Available: https://doi.org/10.1007/978-3-540-28650-9_4

[34] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: the case for cooperation," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, May 2013, pp. 2153–2160.

[35] 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*. IEEE, Sept 2009, pp. 1229–1234.

[36] S. Ferguson, B. Luders, R. C. Grande, and J. P. How, "Real-Time Predictive Modeling and Robust Avoidance of Pedestrians with Uncertain, Changing Intentions," in *arXiv preprint (CoRR)*, vol. abs/1405.5581, 2014. [Online]. Available: http://arxiv.org/abs/1405.5581

[37] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, "Context-based pedestrian path prediction," in *Lecture Notes in Computer Science*, vol. 8694 LNCS, no. PART 6. Springer, Cham, 2014, pp. 618–633.

[38] I. Batkovic, M. Zanon, N. Lubbe, and P. Falcone, "A Computationally Efficient Model for Pedestrian Motion Prediction," in *arXiv preprint (CoRR)*, vol. abs/1803.04702, 2018. [Online]. Available: http://arxiv.org/abs/1803.04702

[39] B. VÃűlz, 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)*. IEEE, Nov 2016, pp. 2607–2612.

[40] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *arXiv preprint (CoRR)*, vol. abs/1612.08242, 2016. [Online]. Available: http://arxiv.org/abs/1612.08242

[41] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent Models of Visual Attention," in *arXiv preprint (CoRR)*, vol. abs/1406.6247, 2014. [Online]. Available: http://arxiv.org/abs/1406.6247

[42] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *arXiv preprint (CoRR)*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[43] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in *Neural Comput.*, vol. 9, no. 8. Cambridge, MA, USA: MIT Press, Nov. 1997, pp. 1735–1780. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[44] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning Temporal Regularity in Video Sequences," in *arXiv preprint (CoRR)*, vol. abs/1604.04574, 2016. [Online]. Available: http://arxiv.org/abs/1604.04574

[45] C. Lu, J. Shi, and J. Jia, "Abnormal Event Detection at 150 FPS in MATLAB," in *2013 IEEE International Conference on Computer Vision*. IEEE, Dec 2013, pp. 2720–2727.

[46] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," vol. 30, no. 5. IEEE Computer Society, May 2008, pp. 909–926.

[47] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating Videos with Scene Dynamics," in *arXiv preprint (CoRR)*, vol. abs/1609.02612, 2016. [Online]. Available: http://arxiv.org/abs/1609.02612

[48] W. Lotter, G. Kreiman, and D. D. Cox, "Unsupervised Learning of Visual Structure using Predictive Generative Networks," in *arXiv preprint (CoRR)*, vol. abs/1511.06380, 2015. [Online]. Available: http://arxiv.org/abs/1511.06380

[49] 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

[50] W. Lotter, G. Kreiman, and D. D. Cox, "Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning," in *arXiv preprint (CoRR)*, vol. abs/1605.08104, 2016. [Online]. Available: http://arxiv.org/abs/1605.08104

[51] S. Jastrzebski, D. Arpit, N. Ballas, V. Verma, T. Che, and Y. Bengio, "Residual Connections Encourage Iterative Inference," in *arXiv preprint (CoRR)*, vol. abs/1710.04773, 2017. [Online]. Available: http://arxiv.org/abs/1710.04773

[52] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *arXiv preprint (CoRR)*, vol. abs/1409.3215, 2014. [Online]. Available: http://arxiv.org/abs/1409.3215

[53] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *arXiv preprint (CoRR)*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[54] M. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," in *arXiv preprint (CoRR)*, vol. abs/1508.04025, 2015. [Online]. Available: http://arxiv.org/abs/1508.04025

[55] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2013, pp. 1700–1709. [Online]. Available: http://www.aclweb.org/anthology/D13-1176

[56] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *arXiv preprint (CoRR)*, vol. abs/1310.4546, 2013. [Online]. Available: http://arxiv.org/abs/1310.4546

[57] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," in *arXiv preprint (CoRR)*, vol. abs/1511.07122, 2015. [Online]. Available: http://arxiv.org/abs/1511.07122

[58] 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

[59] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *arXiv preprint (CoRR)*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[60] 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*. IEEE, June 2010, pp. 2528–2535.

[61] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," in *Journal of machine learning research*, vol. 9, no. Nov. JMLR.org, 2008, pp. 2579–2605.