A Review on Deep Learning Techniques for Video Prediction

S. Oprea, P. Martinez-Gonzalez, A. Garcia-Garcia, J.A. Castro-Vargas, S. Orts-Escolano, J. Garcia-Rodriguez, and A. Argyros

Abstract—The ability to predict, anticipate and reason about future outcomes is a key component of intelligent decision-making systems. In light of the success of deep learning in computer vision, deep-learning-based video prediction emerged as a promising research direction. Defined as a self-supervised learning task, video prediction represents a suitable framework for representation learning, as it demonstrated potential capabilities for extracting meaningful representations of the underlying patterns in natural videos. Motivated by the increasing interest in this task, we provide a review on the deep learning methods for prediction in video sequences. We firstly define the video prediction fundamentals, as well as mandatory background concepts and the most used datasets. Next, we carefully analyze existing video prediction models organized according to a proposed taxonomy, highlighting their contributions and their significance in the field. The summary of the datasets and methods is accompanied with experimental results that facilitate the assessment of the state of the art on a quantitative basis. The paper is summarized by drawing some general conclusions, identifying open research challenges and by pointing out future research directions.

Index Terms—Video prediction, future frame prediction, deep learning, representation learning, self-supervised learning

1 Introduction

Will the car hit the pedestrian? That might be one of the questions that comes to our minds when we observe Figure 1. Answering this question might be in principle a hard task; however, if we take a careful look into the image sequence we may notice subtle clues that can help us predicting into the future, e.g., the person's body indicates that he is running fast enough so he will be able to escape the car's trajectory. This example is just one situation among many others in which predicting future frames in video is useful.

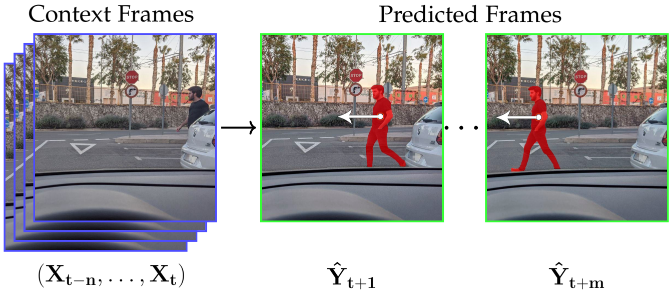

Context Frames
Predicted Frames
$(\mathbf{X_{t-n}}, \ldots, \mathbf{X_t})$
$\hat{\mathbf{Y}}_{\mathbf{t+1}}$
$\hat{\mathbf{Y}}_{\mathbf{t+m}}$

Fig. 1: A pedestrian appeared from behind the white car with the intention of crossing the street. The autonomous car must make a call: hit the emergency braking routine or not. This all comes down to predict the next frames $(\hat{Y}_{t+1}, \ldots, \hat{Y}_{t+m})$ given a sequence of context frames $(X_{t-n}, \ldots, X_t)$, where n and m denote the number of context and predicted frames, respectively. From these predictions at a representation level (RGB, high-level semantics, etc.) a decision-making system would make the car avoid the collision.

In general terms, the prediction and anticipation of future events is a key component of intelligent decision-making systems. Despite the fact that we, humans, solve this problem quite easily and effortlessly, it is extremely challenging from a machine's point of view. Some of the factors that contribute to such complexity are occlusions, camera movement, lighting conditions, clutter, or object deformations. Nevertheless, despite such challenging conditions, many predictive methods have been applied with a certain degree of success in a broad range of application domains such as autonomous driving, robot navigation and human-machine interaction. Some of the tasks in which future prediction has been applied successfully are: anticipating activities and events [1]–[4], long-term planning [5], future prediction of object locations [6], video interpolation [7], predicting instance/semantic segmentation maps [8]–[10], prediction of pedestrian trajectories in traffic [11], anomaly detection [12], precipitation nowcasting [13], [14], and autonomous driving [15].

The great strides made by deep learning algorithms in a variety of research fields such as semantic segmen-

- *S. Oprea, P. Martinez-Gonzalez A. Garcia-Garcia, J. A. Castro-Vargas, and J. Garcia-Rodriguez are with the 3D Perception Lab (3DPL), Department of Computer Technology, University of Alicante, Carrer de San Vicente del Raspeig s/n, E-03690 San Vicent del Raspeig Spain, Spain. E-mail: {soprea, pmartinez, jacastro, jgarcia}@dtic.ua.es*
- *A. Garcia-Garcia is with the Institute of Space Sciences (ICE-CSIC), Campus UAB, Carrer de Can Magrans s/n, E-08193 Barcelona, Spain. E-mail: garciagarcia@ice.csic.es.*
- *S. Orts-Escolano is with the Department of Computer Science and Artificial Intelligence (DCCIA), University of Alicante, Carrer de San Vicente del Raspeig s/n, E-03690 San Vicent del Raspeig Spain, Spain. E-mail: sorts@dccia.ua.es.*
- *A. Argyros is with the Institute of Computer Science, FORTH, Heraklion GR-700 13, Greece and with the Computer Science Department, University of Crete, Heraklion, Rethimno 741 00, Greece. E-mail: argyros@ics.forth.gr.*

tation [16], human action recognition and prediction [17], object pose estimation [18] and registration [19] to name a few, motivated authors to explore deep representation-learning models for future video frame prediction. What made the deep architectures take a leap over the traditional approaches is their ability to learn adequate representations from high-dimensional data in an end-to-end fashion without hand-engineered features [20]. Deep learning-based models fit perfectly into the learning by prediction paradigm, enabling the extraction of meaningful spatio-temporal correlations from video data in a self-supervised fashion.

In this review, we put our focus on deep learning techniques and how they have been extended or applied to future video prediction. We limit this review to the future video prediction given the context of a sequence of previous frames, leaving aside methods that predict future from a static image. In this context, the terms video prediction, future frame prediction, next video frame prediction, future frame forecasting, and future frame generation are used interchangeably. To the best of our knowledge, this is the first review in the literature that focuses on video prediction using deep learning techniques.

This review is organized as follows. First, Sections 2 and 3 lay down the terminology and explain important background concepts that will be necessary throughout the rest of the paper. Next, Section 4 surveys the datasets used by the video prediction methods that are carefully reviewed in Section 5, providing a comprehensive description as well as an analysis of their strengths and weaknesses. Section 6 analyzes typical metrics and evaluation protocols for the aforementioned methods and provides quantitative results for them in the reviewed datasets. Section 7 presents a brief discussion on the presented proposals and enumerates potential future research directions. Finally, Section 8 summarizes the paper and draws conclusions about this work.

2 Video Prediction

The ability to predict, anticipate and reason about future events is the essence of intelligence [21] and one of the main goals of decision-making systems. This idea has biological roots, and also draws inspiration from the predictive coding paradigm [22] borrowed from the cognitive neuroscience field [23]. From a neuroscience perspective, the human brain builds complex mental representations of the physical and causal rules that govern the world. This is primarily through observation and interaction [24]–[26]. The common sense we have about the world arises from the conceptual acquisition and the accumulation of background knowledge from early ages, e.g. biological motion and intuitive physics to name a few. But how can the brain check and refine the learned mental representations from its raw sensory input? The brain is continuously learning through prediction, and refines the already understood world models from the mismatch between its predictions and what actually happened [27]. This is the essence of the predictive coding paradigm that early works tried to implement as computational models [22], [28]–[30].

Video prediction task closely captures the fundamentals of the predictive coding paradigm and it is considered the intermediate step between raw video data and decision making. Its potential to extract meaningful representations of the underlying patterns in video data makes the video prediction task a promising avenue for self-supervised representation learning.

2.1 Problem Definition

We formally define the task of predicting future frames in videos, i.e. video prediction, as follows. Let $\mathbf{X}_t \in \mathrm{R}^{w \times h \times c}$ be the t-th frame in the video sequence $\mathbf{X} = (X_{t-n}, \ldots, X_{t-1}, X_t)$ with n frames, where w, h, and c denote width, height, and number of channels, respectively. The target is to predict the next frames $\mathbf{Y} = (\hat{Y}_{t+1}, \hat{Y}_{t+2}, \ldots, \hat{Y}_{t+m})$ from the input $\mathbf{X}$.

Under the assumption that good predictions can only be the result of accurate representations, learning by prediction is a feasible approach to verify how accurately the system has learned the underlying patterns in the input data. In other words, it represents a suitable framework for representation learning [31], [32]. The essence of predictive learning paradigm is the prediction of plausible future outcomes from a set of historical inputs. On this basis, the task of video prediction is defined as: given a sequence of video frames as context, predict the subsequent frames –generation of continuing video given a sequence of previous frames. Different from video generation that is mostly unconditioned, video prediction is conditioned on a previously learned representation from a sequence of input frames. At a first glance, and in the context of learning paradigms, we can think about the future video frame prediction task as a supervised learning approach because the target frame acts as a label. However, as this information is already available in the input video sequence, no extra labels or human supervision is needed. Therefore, learning by prediction is a self-supervised task, filling the gap between supervised and unsupervised learning.

2.2 Exploiting the Time Dimension of Videos

Unlike static images, videos provide complex transformations and motion patterns in the time dimension. At a fine granularity, if we focus on a small patch at the same spatial location across consecutive time steps, we could identify a wide range of local visually similar deformations due to the temporal coherence. In contrast, by looking at the big picture, consecutive frames would be visually different but semantically coherent. This variability in the visual appearance of a video at different scales is mainly due to, occlusions, changes in the lighting conditions, and camera motion, among other factors. From this source of temporally ordered visual cues, predictive models are able to extract representative spatio-temporal correlations depicting the dynamics in a video sequence. For instance, Agrawal *et al.* [33] established a direct link between vision and motion, attempting to reduce supervision efforts when training deep predictive models.

Recent works study how important is the time dimension for video understanding models [34]. The implicit temporal ordering in videos, also known as the arrow of time, indicates whether a video sequence is playing forward or backward. This temporal direction is also used in the

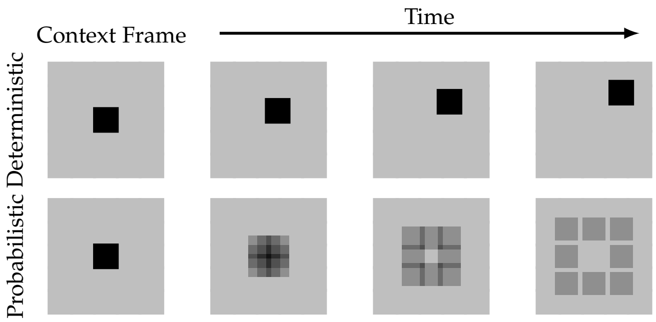


Fig. 2: At top, a deterministic environment where a geometric object, e.g. a black square, starts moving following a random direction. At bottom, probabilistic outcome. Darker areas correspond to higher probability outcomes. As uncertainty is introduced, probabilities get blurry and averaged. Figure inspired by [38].

literature as a supervisory signal [35]–[37]. This further encouraged predictive models to implicitly or explicitly model the spatio-temporal correlations of a video sequence to understand the dynamics of a scene. The time dimension of a video reduces the supervision effort and makes the prediction task self-supervised.

2.3 Dealing with Stochasticity

Predicting how a square is moving, could be extremely challenging even in a deterministic environment such as the one represented in Figure 2. The lack of contextual information and the multiple equally probable outcomes hinder the prediction task. But, what if we use two consecutive frames as context? Under this configuration and assuming a physically perfect environment, the square will be indefinitely moving in the same direction. This represents a deterministic outcome, an assumption that many authors made in order to deal with future uncertainty. Assuming a deterministic outcome would narrow the prediction space to a unique solution. However, this assumption is not suitable for natural videos. The future is by nature multimodal, since the probability distribution defining all the possible future outcomes in a context has multiple modes, i.e. there are multiple equally probable and valid outcomes. Furthermore, on the basis of a deterministic universe, we indirectly assume that all possible outcomes are reflected in the input data. These assumptions make the prediction under uncertainty an extremely challenging task.

Most of the existing deep learning-based models in the literature are deterministic. Although the future is uncertain, a deterministic prediction would suffice some easily predictable situations. For instance, most of the movement of a car is largely deterministic, while only a small part is uncertain. However, when multiple predictions are equally probable, a deterministic model will learn to average between all the possible outcomes. This unpredictability is visually represented in the predictions as blurriness, especially on long time horizons. As deterministic models are unable to handle real-world settings characterized by chaotic dynamics, authors considered that incorporating uncertainty to the model is a crucial aspect. Probabilistic approaches dealing with these issues are discussed in Section 5.6.

2.4 The Devil is in the Loss Function

The design and selection of the loss function for the video prediction task is of utmost importance. Pixel-wise losses, e.g. Cross Entropy (CE), ℓ_2, ℓ_1 and Mean-Squared Error (MSE), are widely used in both unstructured and structured predictions. Although leading to plausible predictions in deterministic scenarios, such as synthetic datasets and video games, they struggle with the inherent uncertainty of natural videos. In a probabilistic environment, with different equally probable outcomes, pixel-wise losses aim to accommodate uncertainty by blurring the prediction, as we can observe in Figure 2. In other words, the deterministic loss functions average out multiple equally plausible outcomes in a single, blurred prediction. In the pixel space, these losses are unstable to slight deformations and fail to capture discriminative representations to efficiently regress the broad range of possible outcomes. This makes difficult to draw predictions maintaining the consistency with our visual similarity notion. Besides video prediction, several studies analyzed the impact of different loss functions in image restoration [39], classification [40], camera pose regression [41] and structured prediction [42], among others. This fosters reasoning about the importance of the loss function, particularly when making long-term predictions in high-dimensional and multimodal natural videos.

Most of distance-based loss functions, such as based on ℓ_p norm, come from the assumption that data is drawn from a Gaussian distribution. But, how these loss functions address multimodal distributions? Assuming that a pixel is drawn from a bimodal distribution with two equally likely modes Mo_1 and Mo_2, the mean value $\overline{Mo} = (Mo_1 + Mo_2)/2$ would minimize the ℓ_p-based losses over the data, even if $\overline{Mo}$ has very low probability [43]. This suggests that the average of two equally probable outcomes would minimize distance-based losses such as, the MSE loss. However, this applies to a lesser extent when using ℓ_1 norm as the pixel values would be the median of the two equally likely modes in the distribution. In contrast to the ℓ_2 norm that emphasizes outliers with the squaring term, the ℓ_1 promotes sparsity thus making it more suitable for prediction in high-dimensional data [43]. Based on the ℓ_2 norm, the MSE is also commonly used in the training of video prediction models. However, it produces low reconstruction errors by merely averaging all the possible outcomes in a blurry prediction as uncertainty is introduced. In other words, the mean image would minimize the MSE error as it is the global optimum, thus avoiding finer details such as facial features and subtle movements as they are noise for the model. Most of the video prediction approaches rely on pixel-wise loss functions, obtaining roughly accurate predictions in easily predictable datasets.

One of the ultimate goals of many video prediction approaches is to palliate the blurry predictions when it comes to uncertainty. For this purpose, authors broadly focused on: directly improving the loss functions; exploring adversarial training; alleviating the training process by reformulating the problem in a higher-level space; or exploring probabilistic alternatives. Some promising results were reported by combining the loss functions with sophisticated regularization terms, e.g. the Gradient Difference Loss (GDL) to enhance prediction sharpness [43] and the Total Variation (TV) regularization to reduce visual artifacts and enforce coherence [7]. Perceptual losses were also used to further improve the visual quality of the predictions [44]–[48]. However, in light of the success of the Generative Adversarial Networks (GANs), adversarial training emerged as a promising alternative to disambiguate between multiple equally probable modes. It was widely used in conjunction with different distance-based losses such as: MSE [49], ℓ_2 [50]–[52], or a combination of them [43], [53]–[57]. To alleviate the training process, many authors reformulated the optimization process in a higher-level space (see Section 5.5). While great strides have been made to mitigate blurriness, most of the existing approaches still rely on distance-based loss functions. As a consequence, the regress-to-the-mean problem remains an open issue. This has further encouraged authors to reformulate existing deterministic models in a probabilistic fashion.

3 Backbone Deep Learning Architectures

In this section, we will briefly review the most common deep networks that are used as building blocks for the video prediction models discussed in this review: convolutional neural networks, recurrent networks, and generative models.

3.1 Convolutional Models

Convolutional layers are the basic building blocks of deep learning architectures designed for visual reasoning since the Convolutional Neural Networks (CNNs) efficiently model the spatial structure of images [58]. As we focus on the visual prediction, CNNs represent the foundation of predictive learning literature. However, their performance is limited by the intra-frame and inter-frame dependencies.

Convolutional operations account for short-range intra-frame dependencies due to their limited receptive fields, determined by the kernel size. This is a well-addressed issue, that many authors circumvented by (1) stacking more convolutional layers [59], (2) increasing the kernel size (although it becomes prohibitively expensive), (3) by linearly combining multiple scales [43] as in the reconstruction process of a Laplacian pyramid [60], (4) using dilated convolutions to capture long-range spatial dependencies [61], (5) enlarging the receptive fields [62], [63], or subsampling, i.e. using pooling operations in exchange for losing resolution. The latter could be mitigated by using residual connections [64], [65] to preserve resolution while increasing the number of stacking convolutions. But even addressing these limitations, would CNNs be able to predict in a longer time horizon?

Vanilla CNNs lack of explicit inter-frame modeling capabilities. To properly model inter-frame variability in a video sequence, 3D convolutions come into play as a promising alternative to recurrent modeling. Several video prediction approaches leveraged 3D convolutions to capture temporal consistency [66]–[70]. Also modeling time dimension, Amersfoort *et al.* [71] replicated a purely convolutional approach in time to address multi-scale predictions in the transformation space. In this case, the learned affine transforms at each time step play the role of a recurrent state.

3.2 Recurrent Models

Recurrent models were specifically designed to model a spatio-temporal representation of sequential data such as video sequences. Among other sequence learning tasks, such as machine translation, speech recognition and video captioning, to name a few, Recurrent Neural Networks (RNNs) [72] demonstrated great success in the video prediction scenario [10], [13], [49], [50], [52], [53], [53], [70], [73]–[85]. Vanilla RNNs have some important limitations when dealing with long-term representations due to the vanishing and exploding gradient issues, making the Backpropagation through time (BPTT) cumbersome. By extending classical RNNs to more sophisticated recurrent models, such as Long Short-Term Memory (LSTM) [86] and Gated Recurrent Unit (GRU) [87], these problems were mitigated. Shi *et al.* extended the use of LSTM-based models to the image space [13]. While some authors explored multidimensional LSTM (MD-LSTM) [88], others stacked recurrent layers to capture abstract spatio-temporal correlations [49], [89]. Zhang *et al.* addressed the duplicated representations along the same recurrent paths [90].

3.3 Generative Models

Whilst discriminative models learn the decision boundaries between classes, generative models learn the underlying distribution of individual classes. More formally, discriminative models capture the conditional probability $p(y|x)$, while generative models capture the joint probability $p(x, y)$, or $p(x)$ in the absence of labels y. The goal of generative models is the following: given some training data, generate new samples from the same distribution. Let input data $\sim p_{data}(x)$ and generated samples $\sim p_{model}(x)$ where, p_{data} and p_{model} are the underlying input data and model's probability distribution respectively. The training process consists in learning a $p_{model}(x)$ similar to $p_{data}(x)$. This is done by explicitly, e.g VAEs, or implicitly, e.g. GANs, estimating a density function from the input data. In the context of video prediction, generative models are mainly used to cope with future uncertainty by generating a wide spectrum of feasible predictions rather than a single eventual outcome.

3.3.1 *Explicit Density Modeling*

These models explicitly define and solve for $p_{model}(x)$.

PixelRNNs and PixelCNNs [91]: These are a type of Fully Visible Belief Networks (FVBNs) [92], [93] that explicitly define a tractable density and estimate the joint distribution $p(x)$ as a product of conditional distributions over the pixels. Informally, they turn pixel generation into a sequential modeling problem, where next pixel values are determined by previously generated ones. In PixelRNNs, this conditional dependency on previous pixels is modeled using two-dimensional (2d) LSTMs. On the other hand, dependencies are modeled using convolutional operations over a context region, thus making training faster. In a nutshell, these methods are outputting a distribution over pixel values at each location in the image, aiming to maximize the likelihood of the training data being generated. Further improvements of the original architectures have been carried out to address different issues. The Gated PixelCNN [94] is computationally more efficient and improves the receptive fields of the original architecture. In the same work, authors also explored conditional modeling of natural images, where the joint probability distribution is conditioned on a latent vector —it represents a high-level image description. This further enabled the extension to video prediction [95].

Variational Autoencoders (VAEs): These models are an extension of Autoencoders (AEs) that encode and reconstruct its own input data x in order to capture a low-dimensional representation z containing the most meaningful factors of variation in x. Extending this architecture to generation, VAEs aim to sample new images from a prior over the underlying latent representation z. VAEs represent a probabilistic spin over the deterministic latent space in AEs. Instead of directly optimizing the density function, which is intractable, they derive and optimize a lower bound on the likelihood. Data is generated from the learned distribution by perturbing the latent variables. In the video prediction context, VAEs are the foundation of many probabilistic models dealing with future uncertainty [9], [38], [55], [81], [85], [96], [97]. Although these variational approaches are able to generate various plausible outcomes, the predictions are blurrier and of lower quality compared to state-of-the-art GAN-based models. Many approaches were taken to leverage the advantages of variational inference: combined adversarial training with VAEs [55], and others incorporated latent probabilistic variables into deterministic models, such as Variational Recurrent Neural Networks (VRNNs) [97], [98] and Variational Encoder-Decoders (VEDs) [99].

3.3.2 *Implicit Density Modeling*

These models learn to sample from p_{model} without explicitly defining it.

GANs [100]: are the backbone of many video prediction approaches [43], [49]–[55], [57], [65], [67], [68], [78], [101]–[106]. Inspired on game theory, these networks consist of two models that are jointly trained as a minimax game to generate new fake samples that resemble the real data. On one hand, we have the discriminator model featuring a probability distribution function describing the real data. On the other hand, we have the generator which tries to generate new samples that fool the discriminator. In their original formulation, GANs are unconditioned –the generator samples new data from a random noise, e.g. Gaussian noise. Nevertheless, Mirza *et al.* [107] proposed the conditional Generative Adversarial Network (cGAN), a conditional version where the generator and discriminator are conditioned on some extra information, e.g. class labels, previous predictions, and multimodal data, among others. CGANs are suitable for video prediction, since the spatio-temporal coherence between the generated frames and the input sequence is guaranteed. The use of adversarial training for the video prediction task, represented a leap over the previous state-of-the-art methods in terms of prediction quality and sharpness. However, adversarial training is unstable. Without an explicit latent variable interpretation, GANs are prone to mode collapse —generator fails to cover the space of possible predictions by getting stuck into a single mode [99], [108]. Moreover, GANs often struggle to balance between the adversarial and reconstruction loss, thus getting blurry predictions. Among the dense literature on adversarial networks, we find some other interesting works addressing GANs limitations [109], [110].

4 Datasets

As video prediction models are mostly self-supervised, they need video sequences as input data. However, some video prediction methods rely on extra supervisory signals, e.g. segmentation maps, and human poses. This makes out-of-domain video datasets perfectly suitable for video prediction. This section describes the most relevant datasets, discussing their pros and cons. Datasets were organized according to their main purpose and summarized in Table 1.

4.1 Action and Human Pose Recognition Datasets

KTH [111]: is an action recognition dataset which includes 2391 video sequences of 4 seconds mean duration, each of them containing an actor performing an action taken with a static camera, over homogeneous backgrounds, at 25 frames per second (fps) and with its resolution downsampled to 160 × 120 pixels. Just 6 different actions are performed, but it was the biggest dataset of this kind at its moment.

Weizmann [112]: is also an action recognition dataset, created for modelling actions as space-time shapes. For this reason, it was recorded at a higher frame rate (50 fps). It just includes 90 video sequences, but performing 10 different actions. It uses a static-camera, homogeneous backgrounds and low resolution (180 × 144 pixels). KTH and Weizmann are usually used together due to their similarities in order to augment the amount of available data.

HMDB-51 [113]: is a large-scale database for human motion recognition. It claims to represent the richness of human motion taking profit from the huge amount of video available online. It is composed by 6766 normalized videos (with mean duration of 3.15 seconds) where humans appear performing one of the 51 considered action categories. Moreover, a stabilized dataset version is provided, in which camera movement is disabled by detecting static backgrounds and displacing the action as a window. It also provides interesting data for each sequence such as body parts visible, point of view respect the human, and if there is camera movement or not. It also exists a joint-annotated version called J-HMBD [114] in which the key points of joints were mannually added for 21 of the HMDB actions.

UCF101 [115]: is an action recognition dataset of realistic action videos, collected from YouTube. It has 101 different action categories, and it is an extension of UCF50, which has 50 action categories. All videos have a frame rate of 25 fps and a resolution of 320 × 240 pixels. Despite being the most used dataset among predictive models, a problem it has is that only a few sequences really represent movement, i.e. they often show an action over a fixed background.

Penn Action Dataset [116]: is an action and human pose recognition dataset from the University of Pennsylvania. It contains 2326 video sequences of 15 different actions, and it also provides human joint and viewpoint (position of the camera respect the human) annotations for each sequence. Each action is balanced in terms of different viewpoints representation.

Human3.6M [117]: is a human pose dataset in which 11 actors with marker-based suits were recorded performing 15 different types of actions. It features RGB images, depth maps (time-of-flight range data), poses and scanned 3D surface meshes of all actors. Silhouette masks and 2D bounding boxes are also provided. Moreover, the dataset was extended by inserting high-quality 3D rigged human models (animated with the previously recorded actions) in real videos, to create a realistic and complex background.

THUMOS-15 [118]: is an action recognition challenge that was celebrated in 2015. It didn't just focus on recognizing an action in a video, but also on determining the time span in which that action occurs. With that purpose, the challenge provided a dataset that extends UCF101 [115] (trimmed videos with one action) with 2100 untrimmed videos where one or more actions take place (with the correspondent temporal annotations) and almost 3000 relevant videos without any of the 101 proposed actions.

4.2 Driving and Urban Scene Understanding Datasets

CamVid [136]: the Cambridge-driving Labeled Video Database is a driving/urban scene understanding dataset which consists of 5 video sequences recorded with a 960 × 720 pixels resolution camera mounted on the dashboard of a car. Four of those sequences were sampled at 1 fps, and one at 15 fps, resulting in 701 frames which were manually per-pixel annotated for semantic segmentation (under 32 classes). It was the first video sequence dataset of this kind to incorporate semantic annotations.

CalTech Pedestrian Dataset [119]: is a driving dataset focused on detecting pedestrians, since its unique annotations are pedestrian bounding boxes. It is conformed of approximately 10 hours of 640 × 480 30fps video taken from a vehicle driving through regular traffic in an urban environment, making a total of 250 000 annotated frames distributed in 137 approximately minute-long segments. The total pedestrian bounding boxes is 350 000, identifying 2300 unique pedestrians. Temporal correspondence between bounding boxes and detailed occlusion labels are also provided.

Kitti [120]: is one of the most popular datasets for mobile robotics and autonomous driving, as well as a benchmark for computer vision algorithms. It is composed by hours of traffic scenarios recorded with a variety of sensor modalities, including high-resolution RGB, gray-scale stereo cameras, and a 3D laser scanner. Despite its popularity, the original dataset did not contain ground truth for semantic segmentation. However, after various researchers manually annotated parts of the dataset to fit their necessities, in 2015 Kitti dataset was updated with 200 annotated frames at pixel level for both semantic and instance segmentation, following the format proposed by the Cityscapes [121] dataset.

Cityscapes [121]: is a large-scale database which focuses on semantic understanding of urban street scenes. It provides semantic, instance-wise, and dense pixel annotations for 30 classes grouped into 8 categories. The dataset consist of around 5000 fine annotated images (1 frame in 30) and 20 000 coarse annotated ones (one frame every 20 seconds or 20 meters run by the car). Data was captured in 50 cities during several months, daytimes, and good weather conditions. All frames are provided as stereo pairs, and the dataset also includes vehicle odometry obtained from in-vehicle sensors, outside temperature, and GPS tracks.

Comma.ai steering angle [137]: is a driving dataset composed by 7.25 hours of largely highway routes. It was recorded as 360 × 180 camera images at 20 fps (divided in 11 different clips), and steering angles, among other vehicle data (speed, GPS, etc.).

Apolloscape [122]: is a driving/urban scene understanding dataset that focuses on 3D semantic reconstruction of the environment. It provides highly precise information about location and 6D camera pose, as well as a much bigger amount of dense per-pixel annotations than other datasets. Along with that, depth information is retireved from a LIDAR sensor, that allows to semantically reconstruct the scene in 3D as a point cloud. It also provides RGB stereo pairs as video sequences recorded under various weather conditions and daytimes. This video sequences and their per-pixel instance annotations make this dataset very interesting for a wide variety of predictive models.

4.3 Object and Video Classification Datasets

Sports1M [123]: is a video classification dataset that also consists of annotated YouTube videos. In this case, it is fully focused on sports: its 487 classes correspond to the sport label retrieved from the YouTube Topics API. Video resolution, duration and frame rate differ across all available videos, but they can be normalized when accessed from YouTube. It is much bigger than UCF101 (over 1 million videos), and movement is also much more frequent.

Youtube-8M [124]: Sports1M [123] dataset is, since 2016, part of a bigger one called YouTube8M, which follows the same philosophy, but with all kind of videos, not just sports. Moreover, it has been updated in order to improve the quality and precision of their annotations. In 2019 YouTube-8M Segments was released with segment-level human-verified labels on about 237 000 video segments on 1000 different classes, which are collected from the validation set of the YouTube-8M dataset. Since YouTube is the biggest video source on the planet, having annotations for some of their videos at segment level is great for predictive models.

YFCC100M [125]: *Yahoo Flickr Creative Commons 100 Million Dataset* is a collection of 100 million images and videos uploaded to Flickr between 2004 and 2014. All those media files were published in Flickr under Creative Commons license, overcoming one of the biggest issues affecting existing multimedia datasets, licensing and volume. Although only 0.8% of the elements of the dataset are videos, it is

TABLE 1: Summary of the most widely used datasets for video prediction (**S/R**: **S**ynthetic/**R**eal, **st**: **st**ereo, **de**: **de**pth, **ss**: **s**emantic **s**egmentation, **is**: **i**nstance **s**egmentation, **sem**: **sem**antic, **I/O**: **I**ndoor/**O**utdoor environment, **bb**: **b**ounding **b**ox, **Act**: **Act**ion label, **ann**: **ann**otated, **env**: **env**ironment, **ToF**: **T**ime **o**f **F**light, **vp**: camera **v**iew**p**oints respect human).

								provided data and ground-truth						
name[1]	**year**	**S/R**	**#videos**	**#frames**	**#ann. frames**	**resolution**	**#classes**	**RGB**	**st**	**de**	**ss**	**is**	**other annotations**	**env.**
Action and human pose recognition datasets														
KTH [111]	2004	R	2391	250 000[2]	0	160 × 120	6 (action)	✓	✗	✗	✗	✗	Act.	O
Weizmann [112]	2007	R	90	9000[2]	0	180 × 144	10 (action)	✓	✗	✗	✗	✗	Act.	O
HMDB-51 [113]	2011	R	6766	639 300	0	*var* × 240	51 (action)	✓	✗	✗	✗	✗	Act., vp	I/O
UCF101 [115]	2012	R	13 320	2 000 000[2]	0	320 × 240	101 (action)	✓	✗	✗	✗	✗	Act.	I/O
Penn Action D. [116]	2013	R	2326	163 841	0	480 × 270	15 (action)	✓	✗	✗	✗	✗	Act., Human poses, vp	I/O
Human3.6M [117]	2014	SR	4000[2]	3 600 000	0	1000x1000	15 (action)	✓	✗	ToF	✗	✗	Act., Human poses & meshes	I/O
THUMOS-15 [118]	2017	R	18 404	3 000 000[2]	0	320 × 240	101 (action)	✓	✗	✗	✗	✗	Act., Time span	I/O
Driving and urban scene understanding datasets														
Camvid [77]	2008	R	5	18 202	701 (ss)	960 × 720	32 (sem)	✓	✗	✗	✓	✗	✗	O
CalTech Pedest. [119]	2009	R	137	1 000 000[2]	250 000 (bb)	640 × 480	-	✓	✗	✗	✗	✗	Pedestrian bb & occlusions	O
Kitti [120]	2013	R	151	48 791	200 (ss)	1392 × 512	30 (sem)	✓	✓	LiDAR	✓	✓	Odometry	O
Cityscapes [121]	2016	R	50	7 000 000[2]	25 000 (ss)	2048 × 1024	30 (sem)	✓	✓	stereo	✓	✓	Odometry, temp, GPS	O
Comma.ai [75]	2016	R	11	522 000[2]	0	160 × 320	-	✓	✗	✗	✗	✗	Steering angles & speed	O
Apolloscape [122]	2018	R	4	200 000	146 997 (ss)	3384 × 2710	25 (sem)	✓	✓	LiDAR	✓	✓	Odometry, GPS	O
Object and video classification datasets														
Sports1m [123]	2014	R	1 133 158	n/a	0	640 × 360 (*var.*)	487 (sport)	✓	✗	✗	✗	✗	Sport label	I/O
YouTube8M [124]	2016	R	8 200 000	n/a	0	*variable*	1000 (topic)	✓	✗	✗	✗	✗	Topic label, Segment info	I/O
YFCC100M [125]	2016	SR	8000	n/a	0	*variable*	-	✓	✗	✗	✗	✗	User tags, Localization	I/O
Video prediction datasets														
Bouncing balls [126]	2008	S	4000	20 000	0	150 × 150	-	✓	✗	✗	✗	✗	✗	-
Van Hateren [127]	2012	R	56	3584	0	128 × 128	-	✓	✗	✗	✗	✗	✗	I/O
NORBvideos [128]	2013	R	110 560	552 800	All (is)	640 × 480	5 (object)	✓	✗	✗	✗	✓	✗	I
Moving MNIST [74]	2015	SR	*custom*[3]	*custom*[3]	0	64 × 64	-	✓	✗	✗	✗	✗	✗	-
Robotic Pushing [89]	2016	R	57 000	1 500 000[2]	0	640 × 512	-	✓	✗	✗	✗	✗	Arm pose	I
BAIR Robot [129]	2017	R	45 000	n/a	0	n/a	-	✓	✗	✗	✗	✗	Arm pose	I
RoboNet [130]	2019	R	161 000	15 000 000	0	*variable*	-	✓	✗	✗	✗	✗	Arm pose	I
Other-purpose and multi-purpose datasets														
ViSOR [131]	2010	R	1529	1 360 000[2]	0	*variable*	-	✓	✗	✗	✗	✗	User tags, human bb	I/O
PROST [132]	2010	R	4 (10)	4936 (9296)	All (bb)	*variable*	-	✓	✗	✗	✗	✗	Object bb	I
Arcade Learning [133]	2013	S	*custom*[3]	*custom*[3]	0	210 × 160	-	✓	✗	✗	✗	✗	✗	-
Inria 3DMovie v2 [134]	2016	R	27	2476	235 (is)	960 × 540	-	✓	✓	✗	✗	✓	Human poses, bb	I/O
Robotrix [16]	2018	S	67	3 039 252	All (ss)	1920 × 1080	39 (sem)	✓	✗	✓	✓	✓	Normal maps, 6D poses	I
UASOL [135]	2019	R	33	165 365	0	2280 × 1282	-	✓	✓	stereo	✗	✗	✗	O

[1] some dataset names have been abbreviated to enhance table's readability.
[2] values estimated based on the framerate and the total number of frames or videos, as the original values are not provided by the authors.
[3] *custom* indicates that as many frames as needed can be generated. This is related to datasets generated from a game, algorithm or simulation, involving interaction or randomness.

still useful for predictive models due to the great variety of these, and therefore the challenge that it represents.

4.4 Video Prediction Datasets

Standard bouncing balls dataset [126]: is a common test set for models that generate high dimensional sequences. It consists of simulations of three balls bouncing in a box. Its clips can be generated randomly with custom resolution but the common structure is composed by 4000 training videos, 200 testing videos and 200 more for validation. This kind of datasets are purely focused on video prediction.

Van Hateren Dataset of natural videos (version [127]): is a very small dataset of 56 videos, each 64 frames long, that has been widely used in unsupervised learning. Original images were taken and given for scientific use by the photographer Hans van Hateren, and they feature moving animals in grasslands along rivers and streams. Its frame size is 128 × 128 pixels. The version we are reviewing is the one provided along with the work of Cadieu and Olshausen [127].

NORBvideos [128]: NORB (NYU Object Recognition Benchmark) dataset [138] is a compilation of static stereo pairs of 50 homogeneously colored objects from various points of view and 6 lightning conditions. Those images were processed to obtain their object masks and even their casted shadows, allowing them to augment the dataset introducing random backgrounds. Viewpoints are determined by rotating the camera through 9 elevations and 18 azimuths (every 20 degrees) around the object. *NORBvideos* dataset was built by sequencing all these frames for each object.

Moving MNIST [74] (M-MNIST): is a video prediction dataset built from the composition of 20-frame video sequences where two handwritten digits from the MNIST database are combined inside a 64 × 64 patch, and moved with some velocity and direction along frames, potentially overlapping between them. This dataset is almost infinite (as new sequences can be generated on the fly), and it also has interesting behaviours due to occlusions and the dynamics of digits bouncing off the walls of the patch. For these reasons, this dataset is widely used by many predictive models. A stochastic variant of this dataset is also available. In the original M-MNIST the digits move with constant velocity and bounce off the walls in a deterministic manner. In contrast, in SM-MNIST digits move with a constant velocity along a trajectory until they hit at wall at which point they bounce off with a random speed and direction. In this way, moments of uncertainty (each time a digit hits a wall) are interspersed with deterministic motion.

Robotic Pushing Dataset [89]: is a dataset created for learning about physical object motion. It consist on 640×512 pixels image sequences of 10 different 7-degree-of-freedom robotic arms interacting with real-world physical objects. No additional labeling is given, the dataset was designed to model motion at pixel level through deep learning algorithms based on convolutional LSTM (ConvLSTM).

BAIR Robot Pushing Dataset (used in [129]): BAIR (Berkeley Artificial Intelligence Research) group has been working on robots that can learn through unsupervised training (also known in this case as self-supervised), this is, learning the consequences that its actions (movement of the arm and grip) have over the data it can measure (images from two cameras). In this way, the robot assimilates physics of the objects and can predict the effects that its actions will generate on the environment, allowing it to plan strategies to achieve more general goals. This was improved by showing the robot how it can grab tools to interact with other objects. The dataset is composed by hours of this self-supervised learning with the robotic arm *Sawyer*.

RoboNet [130]: is a dataset composed by the aggregation of various self-supervised training sequences of seven robotic arms from four different research laboratories. The previously described BAIR group is one of them, along with *Stanford AI Laboratory, Grasp Lab of the University of Pennsylvania and Google Brain Robotics*. It was created with the goal of being a standard, in the same way as ImageNet is for images, but for robotic self-supervised learning. Several experiments have been performed studying how the transfer among robotic arms can be achieved.

4.5 Other-purpose and Multi-purpose Datasets

ViSOR [131]: ViSOR (Video Surveillance Online Repository) is a repository designed with the aim of establishing an open platform for collecting, annotating, retrieving, and sharing surveillance videos, as well as evaluating the performance of automatic surveillance systems. Its raw data could be very useful for video prediction due to its implicit static camera.

PROST [132]: is a method for online tracking that used ten manually annotated videos to test its performance. Four of them were created by PROST authors, and they conform the dataset with the same name. The remaining six sequences were borrowed from other authors, who released their annotated clips to test their tracking methods. We will consider both 4-sequences PROST dataset and 10-sequences aggregated dataset when providing statistics. In each video, different challenges are presented for tracking methods: occlusions, 3D motion, varying illumination, heavy appearance/scale changes, moving camera, motion blur, among others. Provided annotations include bounding boxes for the object/element being tracked.

Arcade Learning Environment [133]: is a platform that enables machine learning algorithms to interact with the Atari 2600 open-source emulator Stella to play over 500 Atari games. The interface provides a single 2D frame of 210×160 pixels resolution at 60 fps in real-time, and up to 6000 fps when it is running at full speed. It also offers the possibility of saving and restoring the state of a game. Although its obvious main application is reinforcement learning, it could also be profitable as source of almost-infinite interactive video sequences from which prediction models can learn.

Inria 3DMovie Dataset v2 [134]: is a video dataset which extracted its data from the *StreetDance 3D* stereo movies. The dataset includes stereo pairs, and manually generated ground-truth for human segmentation, poses and bounding boxes. The second version of this dataset, used in [134], is composed by 27 clips, which represent 2476 frames, of which just a sparse subset of 235 were annotated.

RobotriX [16]: is a synthetic dataset designed for assistance robotics, that consist of sequences where a humanoid robot is moving through various indoor scenes and interacting with objects, recorded from multiple points of view, including robot-mounted cameras. It provides a huge variety of ground-truth data generated synthetically from highly-realistic environments deployed on the cutting-edge game engine UnrealEngine, through the also available tool UnrealROX [139]. RGB frames are provided at 1920×1080 pixels resolution and at 60 fps, along with pixel-precise instance masks, depth and normal maps, and 6D poses of objects, skeletons and cameras. Moreover, UnrealROX is an open source tool for retrieving ground-truth data from any simulation running in UnrealEngine.

UASOL [135]: is a large-scale dataset consisting of high-resolution sequences of stereo pairs recorded outdoors at pedestrian (egocentric) point of view. Along with them, precise depth maps are provided, computed offline from stereo pairs by the same camera. This dataset is intended to be useful for depth estimation, both from single and stereo images, research fields where outdoor and pedestrian-point-of-view data is not abundant. Frames were taken at a resolution of 2280×1282 pixels at 15 fps.

5 Video Prediction Methods

In the video prediction literature we find a broad range of different methods and approaches. Early models focused on directly predicting raw pixel intensities, by implicitly modeling scene dynamics and low-level details (Section 5.1). However, extracting a meaningful and robust representation from raw videos is challenging, since the pixel space is highly dimensional and extremely variable. From this point, reducing the supervision effort and the representation dimensionality emerged as a natural evolution. On the one hand, the authors aimed to disentangle the factors of variation from the visual content, i.e. factorizing the prediction space. For this purpose, they: (1) formulated the prediction problem into an intermediate transformation space by explicitly modeling the source of variability as transformations between frames (Section 5.2); (2) separated motion from the visual content with a two-stream computation (Section 5.3). On the other hand, some models narrowed the output space by conditioning the predictions on extra variables (Section 5.4), or reformulating the problem in a higher-level space (Section 5.5). High-level representation spaces are increasingly more attractive, since intelligent systems rarely

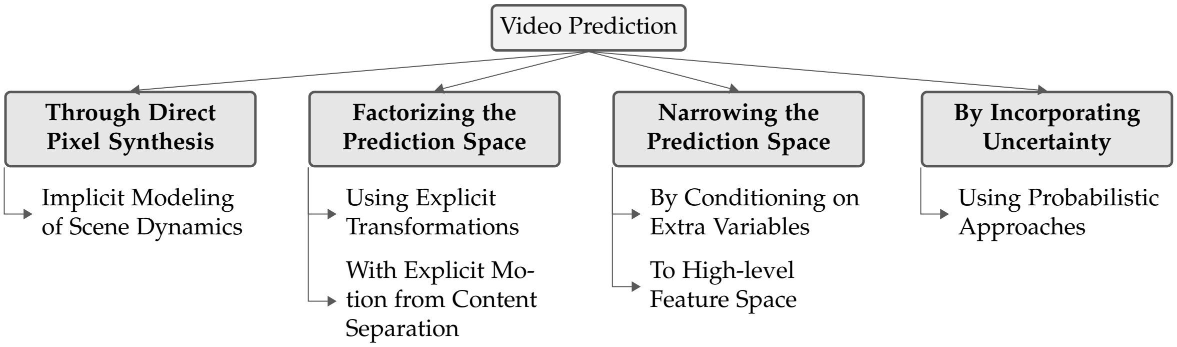

Video Prediction
Through Direct Pixel Synthesis
Factorizing the Prediction Space
Narrowing the Prediction Space
By Incorporating Uncertainty
Implicit Modeling of Scene Dynamics
Using Explicit Transformations
With Explicit Motion from Content Separation
By Conditioning on Extra Variables
To High-level Feature Space
Using Probabilistic Approaches

Fig. 3: Classification of video prediction models.

rely on raw pixel information for decision making. Besides simplifying the prediction task, some other works addressed the future uncertainty in predictions. As the vast majority of video prediction models are deterministic, they are unable to manage probabilistic environments. To address this issue, several authors proposed modeling future uncertainty with probabilistic models (Section 5.6).

So far in the literature, there is no specific taxonomy that classifies video prediction models. In this review, we have classified the existing methods according to the video prediction problem they addressed and following the classification illustrated in Figure 3. For simplicity, each subsection extends directly the last level in the taxonomy. Moreover, some methods in this review can be classified in more than one category since they addressed multiple problems. For instance, [9], [54], [85] are probabilistic models making predictions in a high-level space as they addressed both the future uncertainty and high dimensionality in videos. The category of these models were specified according to their main contribution. The most relevant methods, ordered in a chronological order, are summarized in Table 2 containing low-level details. Prediction is a widely discussed topic in different fields and at different levels of abstraction. For instance, the future prediction from a static image [3], [106], [140]–[143], vehicle behavior prediction [144] and human action prediction [17] are a different but inspiring research fields. Although related, the aforementioned topics are outside the scope of this particular review, as it focuses purely on the video prediction methods using a sequence of previous frames as context and is limited to 2D RGB data.

5.1 Direct Pixel Synthesis

Initial video prediction models attempted to directly predict future pixel intensities without any explicit modeling of the scene dynamics. Ranzato *et al.* [73] discretized video frames in patch clusters using k-means. They assumed that non-overlapping patches are equally different in a k-means discretized space, yet similarities can be found between patches. The method is a convolutional extension of a RNN-based model [145] making short-term predictions at the patch-level. As the full-resolution frame is a composition of the predicted patches, some tilling effect can be noticed. Predictions of large and fast-moving objects are accurate, however, when it comes to small and slow-moving objects there is still room for improvement. These are common issues for most methods making predictions at the patch-level. Addressing longer-term predictions, Srivastava *et al.* [74] proposed different AE-based approaches incorporating LSTM units to model the temporal coherence. Using convolutional [146] and flow [147] percepts alongside RGB image patches, authors tested the models on multi-domain tasks and considered both unconditioned and conditioned decoder versions. The latter only marginally improved the prediction accuracy. Replacing the fully connected LSTMs with convolutional LSTMs, Shi *et al.* proposed an end-to-end model efficiently exploiting spatial correlations [13]. This enhanced prediction accuracy and reduced the number of parameters.

Inspired on adversarial training: Building on the recent success of the Laplacian Generative Adversarial Network (LAPGAN), Mathieu *et al.* proposed the first multi-scale architecture for video prediction that was trained in an adversarial fashion [43]. Their novel GDL regularization combined with ℓ_1-based reconstruction and adversarial training represented a leap over the previous state-of-the-art models [73], [74] in terms of prediction sharpness. However, it was outperformed by the Predictive Coding Network (PredNet) [75] which stacked several ConvLSTMs vertically connected by a bottom-up propagation of the local ℓ_1 error computed at each level. Previously to PredNet, the same authors proposed the Predictive Generative Network (PGN) [49], an end-to-end model trained with a weighted combination of adversarial loss and MSE on synthetic data. However, no tests on natural videos and comparison with state-of-the-art predictive models were carried out. Using a similar training strategy as [43], Zhou *et al.* used a convolutional AE to learn long-term dependencies from time-lapse videos [103]. Build on Progressively Growing GANs (PGGANs) [148], Aigner *et al.* proposed the FutureGAN [69], a three-dimensional (3d) convolutional Encoder-decoder (ED)-based model. They used the Wasserstein GAN with gradient penalty (WGAN-GP) loss [149] and conducted experiments on increasingly complex datasets. Extending [13], Zhang *et al.* proposed a novel LSTM-based architecture where hidden states are updated along a z-order curve [70]. Dealing with distortion and temporal inconsistency in predictions and inspired by the Human Visual System (HVS), Jin *et al.* [150] first incorporated multi-frequency analysis into the video prediction task to decompose images into low and high frequency bands. This allowed high-fidelity and temporally consistent predictions with the ground truth, as the model better leverages the spatial and temporal details. The proposed method outperformed previous state-of-the-art in all metrics except in the Learned Perceptual Image Patch Similarity (LPIPS), where probabilistic models achieved a better performance since their predictions are clearer and realistic but less consistent with the ground truth. Distortion and blurriness are further accentuated when it comes to predict under fast camera motions. To this end, Shouno [151] implemented a hierarchical residual network with top-down connections. Leveraging parallel prediction at multiple scales, authors reported finer details and textures under fast and large camera motion.

Bidirectional flow: Under the assumption that video sequences are symmetric in time, Kwon *et al.* [101] explored a retrospective prediction scheme training a generator for both, forward and backward prediction (reversing the input sequence to predict the past). Their cycle GAN-based approach ensure the consistency of bidirectional prediction through retrospective cycle constraints. Similarly, Hu *et al.* [57] proposed a novel cycle-consistency loss used to train a GAN-based approach (VPGAN). Future frames are generated from a sequence of context frames and their variation in time, denoted as Z. Under the assumption that Z is symmetric in the encoding space, it is manipulated by the model manipulates to generate desirable moving directions. In the same spirit, other works focused on both, forward and backward predictions [37], [152]. Enabling state sharing between the encoder and decoder, Oliu *et al.* proposed the folded Recurrent Neural Network (fRNN) [153], a recurrent AE architecture featuring GRUs that implement a bidirectional flow of the information. The model demonstrated a stratified representation, which makes the topology more explainable, as well as efficient compared to regular AEs in terms or memory consumption and computational requirements.

Exploiting 3D convolutions: for modeling short-term features, Wang *et al.* [66] integrated them into a recurrent network demonstrating promising results in both video prediction and early activity recognition. While 3D convolutions efficiently preserves local dynamics, RNNs enables long-range video reasoning. The *eidetic* 3d LSTM (E3d-LSTM) network, represented in Figure 4, features a gated-controlled self-attention module, i.e. *eidetic* 3D memory, that effectively manages historical memory records across multiple time steps. Outperforming previous works, Yu *et al.* proposed the Conditionally Reversible Network (CrevNet) [154] consisting of two modules, an invertible AE and a Reversible Predictive Model (RPM). While the bijective two-way AE ensures no information loss and reduces the memory consumption, the RPM extends the reversibility from spatial to temporal domain. Some other works used 3D convolutional operations to model the time dimension [69].

Analyzing the previous works, Byeon *et al.* [76] identified a lack of spatial-temporal context in the representations, leading to blurry results when it comes to the future uncertainty. Although authors addressed this contextual limitation with dilated convolutions and multi-scale architectures, the context representation progressively vanishes in long-term predictions. To address this issue, they proposed a context-aware model that efficiently aggregates per-pixel contextual information at each layer and in multiple directions. The core of their proposal is a context-aware layer consisting of two blocks, one aggregating the information from multiple directions and the other blending them into a unified context.

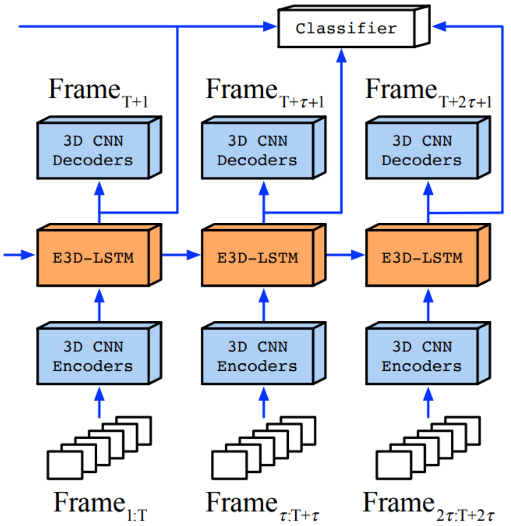
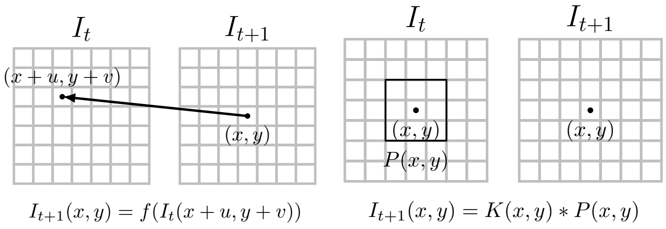


Fig. 4: Representation of the 3D encoder-decoder architecture of E3d-LSTM [66]. After reducing T consecutive input frames to high-dimensional feature maps, these are directly fed into a novel *eidetic* module for modeling long-term spatiotemporal dependencies. Finally, stacked 3D CNN decoder outputs the predicted video frames. For classification tasks the hidden states can be directly used as the learned video representation. Figure extracted from [66].



(a) Vector-based.

(b) Kernel-based.

Fig. 5: Representation of transformation-based approaches. (a) Vector-based with a bilinear interpolation. (b) Applying transformations as a convolutional operation. Figure inspired by [155].

Extracting a robust representation from raw pixel values is an overly complicated task due to the high-dimensionality of the pixel space. The per-pixel variability between consecutive frames, causes an exponential growth in the prediction error on the long-term horizon.

5.2 Using Explicit Transformations

Let $\mathbf{X} = (X_{t-n}, \dots, X_{t-1}, X_t)$ be a video sequence of n frames, where t denotes time. Instead of learning the visual appearance, transformation-based approaches assume that visual information is already available in the input sequence. To deal with the strong similarity and pixel redundancy between successive frames, these methods explicitly model the transformations that takes a frame at time t to the frame at $t+1$. These models are formally defined as follows:

$$\mathbf{Y}_{t+1} = \mathcal{T}\left(\mathcal{G}\left(\mathbf{X}_{t-n:t}\right), \mathbf{X}_{t-n:t}\right), \tag{1}$$

where $\mathcal{G}$ is a learned function that outputs future transformation parameters, which applied to the last observed frame $\mathbf{X}_t$ using the function $\mathcal{T}$, generates the future frame prediction $\mathbf{Y}_{t+1}$. According to the classification of Reda *et al.* [155], $\mathcal{T}$ function can be defined as a vector-based resampling such as bilinear sampling, or adaptive kernel-based resampling, e.g. using convolutional operations. For instance, a bilinear sampling operation is defined as:

$$\mathbf{Y}_{t+1}(x, y) = f\left(\mathbf{X}_t(x + u, y + v)\right), \tag{2}$$

where f is a bilinear interpolator such as [7], [156], [157], (u, v) is a motion vector predicted by $\mathcal{G}$, and $X_t(x, y)$ is a pixel value at (x,y) in the last observed frame X_t. Approaches following this formulation are categorized as vector-based resampling operations and are depicted in Figure 5a.

On the other side, in the kernel-based resampling, the $\mathcal{G}$ function predicts the kernel $\mathrm{K}(x, y)$ which is applied as a convolution operation using $\mathcal{T}$, as depicted in Figure 5b and is mathematically represented as follows:

$$\mathbf{Y}_{t+1}(x, y) = \mathrm{K}(x, y) * \mathbf{P}_t(x, y), \tag{3}$$

where $\mathrm{K}(x, y) \in \mathbb{R}^{NxN}$ is the 2D kernel predicted by the function $\mathcal{G}$ and $P_t(x, y)$ is an $N \times N$ patch centered at (x, y).

Combining kernel and vector-based resampling into a hybrid solution, Reda *et al.* [155] proposed the Spatially Displaced Convolution (SDC) module that synthesizes high-resolution images applying a learned per-pixel motion vector and kernel at a displaced location in the source image. Their 3D CNN model trained on synthetic data and featuring the SDC modules, reported promising predictions of a high-fidelity.

5.2.1 Vector-based Resampling

Bilinear models use multiplicative interactions to extract transformations from pairs of observations in order to relate images, such as Gated Autoencoders (GAEs) [158]. Inspired by these models, Michalski *et al.* proposed the Predictive Gating Pyramid (PGP) [159] consisting on a recurrent pyramid of stacked GAEs. To the best of our knowledge, this was the first attempt to predict future frames in the affine transform space. Multiple GAEs are stacked to represent a hierarchy of transformations and capture higher-order dependencies. From the experiments on predicting frequency modulated sin-waves, authors stated that standard RNNs were outperformed in terms of accuracy. However, no performance comparison was conducted on videos.

Based on the Spatial Transformer (ST) module [160]: To provide spatial transformation capabilities to existing CNNs, Jaderberg *et al.* [160] proposed the ST module represented in Figure 6. It regresses different affine transformation parameters for each input, to be applied as a single transformation to the whole feature map(s) or image(s).

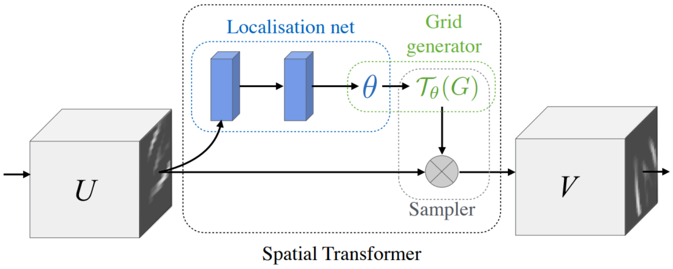


Fig. 6: A representation of the spatial transformer module proposed by [160]. First, the localization network regresses the transformation parameters, denoted as θ, from the input feature map U. Then, the grid generator creates a sampling grid from the predicted transformation parameters. Finally, the sampler produces the output map by sampling the input at the points defined in the sampling grid. Figure extracted from [160].

Moreover, it can be incorporated at any part of the CNNs and it is fully differentiable. The ST module is the essence of vector-based resampling approaches for video prediction. As an extension, Patraucean *et al.* [77] modified the grid generator to consider per-pixel transformations instead of a single dense transformation map for the entire image. They nested a LSTM-based temporal encoder into a spatial AE, proposing the AE-convLSTM-flow architecture. The prediction is generated by resampling the current frame with the flow-based predicted transformation. Using the components of the AE-convLSTM-flow architecture, Lu *et al.* [78] assembled an extrapolation module which is unfolded in time for multi-step prediction. Their Flexible Spatio-semporal Network (FSTN) features a novel loss function using the DeePSiM perceptual loss [44] in order to mitigate blurriness. An exhaustive experimentation and ablation study was carried out, testing multiple combinations of loss functions. Also inspired on the ST module for the volume sampling layer, Liu *et al.* proposed the Deep Voxel Flow (DVF) architecture [7]. It consists of a multi-scale flow-based ED model originally designed for the video frame interpolation task, but also evaluated on a predictive basis reporting sharp results. Liang *et al.* [55] use a flow-warping layer based on a bilinear interpolation. Finn *et al.* proposed the Spatial Transformer Predictor (STP) motion-based model [89] producing 2D affine transformations for bilinear sampling. Pursuing efficiency, Amersfoort *et al.* [71] proposed a CNN designed to predict local affine transformations of overlapping image patches. Unlike the ST module, authors estimated transformations of input frames off-line and at patch level. As the model is parameter-efficient, it was unfolded in time for multi-step prediction. This resembles RNNs as the parameters are shared over time and the local affine transforms play the role of recurrent states.

5.2.2 Kernel-based Resampling

As a promising alternative to the vector-based resampling, recent approaches synthesize pixels by convolving input patches with a predicted kernel. However, convolutional operations are limited in learning spatial invariant representations of complex transformations. Moreover, due to their local receptive fields, global spatial information is not fully preserved. Using larger kernels would help to preserve global features, but in exchange for a higher memory consumption. Pooling layers are another alternative, but loosing spatial resolution. Preserving spatial resolution at a low computational cost is still an open challenge for future video frame prediction task. Transformation layers used in vector-based resampling [7], [77], [160] enabled CNNs to be spatially invariant and also inspired kernel-based architectures.

Inspired on the Convolutional Dynamic Neural Advection (CDNA) module [89]: In addition to the STP vector-based model, Finn *et al.* [89] proposed two different kernel-based motion prediction modules outperforming previous approaches [43], [80], (1) the Dynamic Neural Advection (DNA) module predicting different distributions for each pixel and (2) the CDNA module that instead of predicting different distributions for each pixel, it predicts multiple discrete distributions applied convolutionally to the input. While, CDNA and STP mask out objects that are moving in consistent directions, the DNA module produces per-pixel motion. These modules inspired several kernel-based approaches. Similar to the CDNA module, Klein *et al.* proposed the Dynamic Convolutional Layer (DCL) [161] for short-range weather prediction. Likewise, Brabandere *et al.* [162] proposed the Dynamic Filter Networks (DFN) generating sample (for each image) and position-specific (for each pixel) kernels. This enabled sophisticated and local filtering operations in comparison with the ST module, that is limited to global spatial transformations. Different to the CDNA model, the DFN uses a softmax layer to filter values of greater magnitude, thus obtaining sharper predictions. Moreover, temporal correlations are exploited using a parameter-efficient recurrent layer, much simpler than [13], [74]. Exploiting adversarial training, Vondrick *et al.* proposed a cGAN-based model [102] consisting of a discriminator similar to [67] and a CNN generator featuring a transformer module inspired on the CDNA model. Different from the CDNA model, transformations are not applied recurrently on a per-frame basis. To deal with in-the-wild videos and make predictions invariant to camera motion, authors stabilized the input videos. However, no performance comparison with previous works has been conducted.

Relying on kernel-based transformations and improving [163], Luc *et al.* [164] proposed the Transformation-based & TrIple Video Discriminator GAN (TrIVD-GAN-FP) featuring a novel recurrent unit that computes the parameters of a transformation used to warp previous hidden states without any supervision. These Transformation-based Spatial Recurrent Units (TSRUs) are generic modules and can replace any traditional recurrent unit in currently existent video prediction approaches.

Object-centric representation: Instead of focusing on the whole input, Chen *et al.* [50] modeled individual motion of local objects, i.e. object-centered representations. Based on the ST module and a pyramid-like sampling [165], authors implemented an attention mechanism for object selection. Moreover, transformation kernels were generated dynamically as in the DFN, to then apply them to the last patch containing an object. Although object-centered predictions is novel, performance drops when dealing with multiple objects and occlusions as the attention module fails to distinguish them correctly.

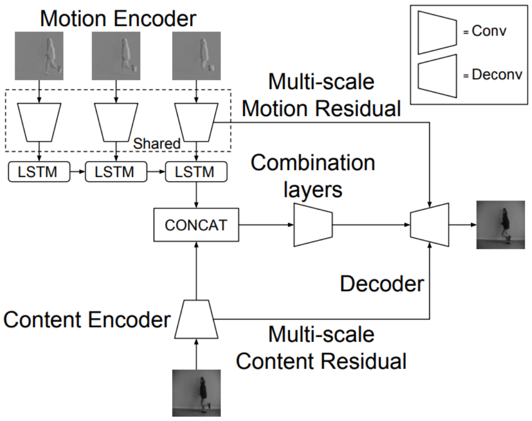


Fig. 7: MCnet with Multi-scale Motion-Content Residuals. While the motion encoder captures the temporal dynamics in a sequence of image differences, the content encoder extracts meaningful spatial features from the last observed RGB frame. After that, the network computes motion-content features that are fed into the decoder to predict the next frame. Figure extracted from [65].

5.3 Explicit Motion from Content Separation

Drawing inspiration from two-stream architectures for action recognition [166], video generation from a static image [67], and unconditioned video generation [68], authors decided to factorize the video into content and motion to process each on a separate pathway. By decomposing the high-dimensional videos, the prediction is performed on a lower-dimensional temporal dynamics separately from the spatial layout. Although this makes end-to-end training difficult, factorizing the prediction task into more tractable problems demonstrated good results.

The Motion-content Network (MCnet) [65], represented in Figure 7 was the first end-to-end model that disentangled scene dynamics from the visual appearance. Authors performed an in-depth performance analysis ensuring the motion and content separation through generalization capabilities and stable long-term predictions compared to models that lack of explicit motion-content factorization [43], [74]. In a similar fashion, yet working in a higher-level pose space, Denton *et al.* proposed Disentangled-representation Net (DRNET) [79] using a novel adversarial loss —it isolates the scene dynamics from the visual content, considered as the discriminative component— to completely disentangle motion dynamics from content. Outperforming [43], [65], the DRNET demonstrated a clean motion from content separation by reporting plausible long-term predictions on both synthetic and natural videos. To improve prediction variability, Liang *et al.* [55] fused the future-frame and future-flow prediction into a unified architecture with a shared probabilistic motion encoder. Aiming to mitigate the ghosting effect in disoccluded regions, Gae *et al.* [167]

proposed a two-staged approach consisting of a separate computation of flow and pixel predictions. As they focused on inpainting occluded regions of the image using flow information, they improved results on disoccluded areas avoiding undesirable artifacts and enhancing sharpness. Separating the moving objects and the static background, Wu *et al.* [168] proposed a two-staged architecture that firstly predicts the static background to then, using this information, predict the moving objects in the foreground. Final results are generated through composition and by means of a video inpainting module. Reported predictions are quite accurate, yet performance was not contrasted with the latest video prediction models.

Although previous approaches disentangled motion from content, they have not performed an explicit decomposition into low-dimensional components. Addressing this issue, Hsieh *et al.* proposed the Decompositional Disentangled Predictive Autoencoder (DDPAE) [169] that decomposes the high-dimensional video into components represented with low-dimensional temporal dynamics. On the Moving MNIST dataset, DDPAE first decomposes images into individual digits (components) to then factorize each digit into its visual appearance and spatial location, being the latter easier to predict. Although experiments were performed on synthetic data, this approach represents a promising baseline to disentangle and decompose natural videos. Moreover, it is applicable to other existing models to improve their predictions.

5.4 Conditioned on Extra Variables

Conditioning the prediction on extra variables such as vehicle odometry or robot state, among others, would narrow the prediction space. These variables have a direct influence on the dynamics of the scene, providing valuable information that facilitates the prediction task. For instance, the motion captured by a camera placed on the dashboard of an autonomous vehicle is directly influenced by the wheel-steering and acceleration. Without explicitly exploiting this information, we rely blindly on the model's capabilities to correlate the wheel-steering and acceleration with the perceived motion. However, the explicit use of these variables would guide the prediction.

Following this paradigm, Oh *et al.* first made long-term video predictions conditioned by control inputs from Atari games [80]. Although the proposed ED-based models reported very long-term predictions (+100), performance drops when dealing with small objects (e.g. bullets in Space Invaders) and while handling stochasticity due to the squared error. However, by simply minimizing ℓ_2 error can lead to accurate and long-term predictions for deterministic synthetic videos, such as those extracted from Atari video games. Building on [80], Chiappa *et al.* [170] proposed alternative architectures and training schemes alongside an in-depth performance analysis for both short and long-term prediction. Similar model-based control from visual inputs performed well in restricted scenarios [171], but was inadequate for unconstrained environments. These deterministic approaches are unable to deal with natural videos in the absence of control variables.

To address this limitation, the models proposed by Finn et al. [89] successfully made predictions on natural images, conditioned on the robot state and robot-object interactions performed in a controlled scenario. These models predict per-pixel transformations conditioned by the previous frame, to finally combine them using a composition mask. They outperformed [43], [80] on both conditioned and unconditioned predictions, however the quality of long-term predictions degrades over time because of the blurriness caused by the MSE loss function. Also, using high-dimensional sensory such as images, Dosovitskiy *et al.* [172] proposed a sensorimotor control model which enables interaction in complex and dynamic 3d environments. The approach is a reinforcement learning (RL)-based techniques, with the difference that instead of building upon a monolithic state and a scalar reward, the authors consider high-dimensional input streams, such as raw visual input, alongside a stream of measurements or player statistics. Although the outputs are future measurements instead of visual predictions, it was proven that using multivariate data benefits decision-making over conventional scalar reward approaches.

5.5 In the High-level Feature Space

Despite the vast work on video prediction models, there is still room for improvement in natural video prediction. To deal with the curse of dimensionality, authors reduced the prediction space to high-level representations, such as semantic and instance segmentation, and human pose. Since the pixels are categorical, the semantic space greatly simplifies the prediction task, yet unexpected deformations in semantic maps and disocclusions, i.e. initially occluded scene entities become visible, induce uncertainty. However, high-level prediction spaces are more tractable and constitute good intermediate representations. By bypassing the prediction in the pixel space, models become able to report longer-term and more accurate predictions.

5.5.1 Semantic Segmentation

In recent years, semantic and instance representations have gained increasing attention, emerging as a promising avenue for complete scene understanding. By decomposing the visual scene into semantic entities, such as pedestrians, vehicles and obstacles, the output space is narrowed to high-level scene properties. This intermediate representation represents a more tractable space as pixel values of a semantic map are categorical. In other words, scene dynamics are modeled at the semantic entity level instead of being modeled at the pixel level. This has encouraged authors to (1) leverage future prediction to improve parsing results [51] and (2) directly predict segmentation maps into the future [8], [56], [173].

Exploring the scene parsing in future frames, Jin *et al.* proposed the Parsing with prEdictive feAtuRe Learning (PEARL) framework [51] which was the first to explore the potential of a GAN-based frame prediction model to improve per-pixel segmentation. Specifically, this framework conducts two complementary predictive learning tasks. Firstly, it captures the temporal context from input data by using a single-frame prediction network. Then, these temporal features are embedded into a frame parsing network through a transform layer for generating per-pixel

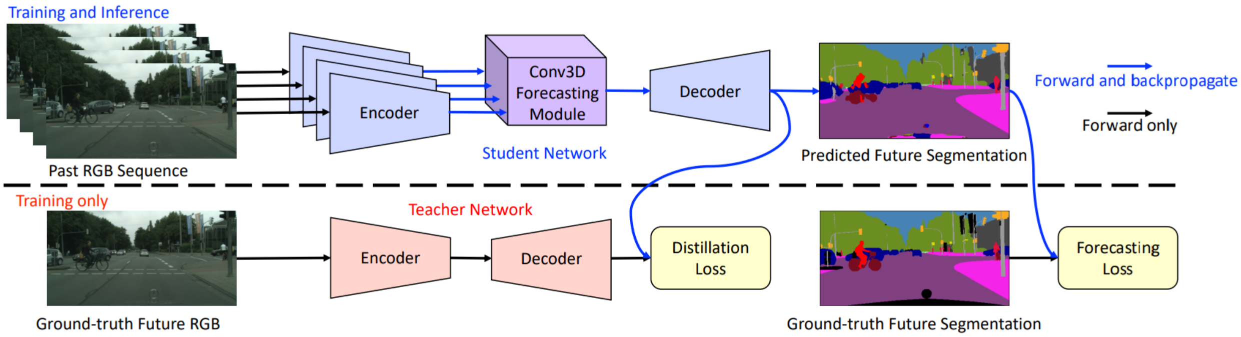


Fig. 8: Two-staged method proposed by Chiu *et al.* [174]. In the upper half, the student network consists on an ED-based architecture featuring a 3D convolutional forecasting module. It performs the forecasting task guided by an additional loss generated by the teacher network (represented in the lower half). Figure extracted from [174].

future segmentations. Although the predictive net was not compared with existing approaches, PEARL outperforms the traditional parsing methods by generating temporally consistent segmentations. In a similar fashion, Luc *et al.* [56] extended the msCNN model of [43] to the novel task of predicting semantic segmentations of future frames, using softmax pre-activations instead of raw pixels as input. The use of intermediate features or higher-level data as input is a common practice in the video prediction performed in the high-level feature space. Some authors refer to this type or input data as percepts. Luc *et al.* explored different combinations of loss functions, inputs (using RGB information alongside percepts), and outputs (autoregressive and batch models). Results on short, medium and long-term predictions are sound, however, the models are not end-to-end and they do not capture explicitly the temporal continuity across frames. To address this limitation and extending [51], Jin *et al.* first proposed a model for jointly predicting motion flow and scene parsing [175]. Flow-based representations implicitly draw temporal correlations from the input data, thus producing temporally coherent per-pixel segmentations. As in [56], the authors tested different network configurations, as using Res101-FCN percepts for the prediction of semantic maps, and also performed multi-step prediction up to 10 time-steps into the future. Per-pixel accuracy improved when segmenting small objects, e.g. pedestrians and traffic signs, which are more likely to vanish in long-term predictions. Similarly, except that time dimension is modeled with LSTMs instead of motion flow estimation, Nabavi *et al.* proposed a simple bidirectional ED-LSTM [82] using segmentation masks as input. Although the literature on knowledge distillation [176], [177] stated that softmax pre-activations carry more information than class labels, this model outperforms [56], [175] on short-term predictions.

Another relevant idea is to use both motion flow estimation alongside LSTM-based temporal modeling. In this direction, Terwilliger *et al.* [10] proposed a novel method performing a LSTM-based feature-flow aggregation. Authors also tried to further simplify the semantic space by disentangling motion from semantic entities [65], achieving low overhead and efficiency. The prediction problem was decomposed into two subtasks, that is, current frame segmentation and future optical flow prediction, which are finally combined with a novel end-to-end warp layer. An improvement on short-term predictions were reported over previous works [56], [175], yet performing worse on mid-term predictions.

A different approach was proposed by Vora *et al.* [83] which first incorporated structure information to predict future 3D segmented point clouds. Their geometry-based model consists of several derivable sub-modules: (1) the pixel-wise segmentation and depth estimation modules which are jointly used to generate the 3d segmented point cloud of the current RGB frame; and (2) an LSTM-based module trained to predict future camera ego-motion trajectories. The future 3d segmented point clouds are obtained by transforming the previous point clouds with the predicted ego-motion. Their short-term predictions improved the results of [56], however, the use of structure information for longer-term predictions is not clear.

The main disadvantage of two-staged, i.e. not end-to-end, approaches [10], [56], [82], [83], [175] is that their performance is constrained by external supervisory signals, e.g. optical flow [178], segmentation [179] and intermediate features or percepts [61]. Breaking this trend, Chiu *et al.* [174] first solved jointly the semantic segmentation and forecasting problems in a single end-to-end trainable model by using raw pixels as input. This ED architecture is based on two networks, with one performing the forecasting task (student) and the other (teacher) guiding the student by means of a novel knowledge distillation loss. An in-depth ablation study was performed, validating the performance of the ED architectures as well as the 3D convolution used for capturing temporal scale instead of a LSTM or ConvLSTM, as in previous works.

Avoiding the flood of deterministic models, Bhattacharyya *et al.* proposed a Bayesian formulation of the ResNet model in a novel architecture to capture model and observation uncertainty [9]. As main contribution, their dropout-based Bayesian approach leverages synthetic likelihoods [180] to encourage prediction diversity and deal with multi-modal outcomes. Since Cityscapes sequences have been recorded in the frame of reference of a moving vehicle, authors conditioned the predictions on vehicle odometry.

5.5.2 Instance Segmentation

While great strides have been made in predicting future segmentation maps, the authors attempted to make predictions at a semantically richer level, i.e. future prediction of semantic instances. Predicting future instance-level segmentations is a challenging and weakly unexplored task. This is because instance labels are inconsistent and variable in number across the frames in a video sequence. Since the representation of semantic segmentation prediction models is of fixed-size, they cannot directly address semantics at the instance level.

To overcome this limitation and introducing the novel task of predicting instance segmentations, Luc *et al.* [8] predict fixed-sized feature pyramids, i.e. features at multiple scales, used by the Mask R-CNN [181] network. The combination of dilated convolutions and multi-scale, efficiently preserve high-resolution details improving the results over previous methods [56]. To further improve predictions, Sun *et al.* [84] focused on modeling not only the spatio-temporal correlations between the pyramids, but also the intrinsic relations among the feature layers inside them. By enriching the contextual information using the proposed Context Pyramid ConvLSTMs (CP-ConvLSTM), an improvement in the prediction was noticed. Although the authors have not shown any long-term predictions nor compared with semantic segmentation models, their approach is currently the state of the art in the task of predicting instance segmentations, outperforming [8].

5.5.3 *Other High-level Spaces*

Although semantic and instance segmentation spaces were the most used in video prediction, other high-level spaces such as human pose and keypoints represent a promising avenue.

Human Pose: As the human pose is a low-dimensional and interpretable structure, it represents a cheap supervisory signal for predictive models. This fostered pose-guided prediction methods, where future frame regression in the pixel space is conditioned by intermediate prediction of human poses. However, these methods are limited to videos with human presence. As this review focuses on video prediction, we will briefly review some of the most relevant methods predicting human poses as an intermediate representation.

From a supervised prediction of human poses, Villegas *et al.* [53] regress future frames through analogy making [182]. Although background is not considered in the prediction, authors compared the model against [13], [43] reporting long-term results. To make the model unsupervised on the human pose, Wichers *et al.* [52] adopted different training strategies: end-to-end prediction minimizing the ℓ_2 loss, and through analogy making, constraining the predicted features to be close to the outputs of the future encoder. Different from [53], in this work the predictions are made in the feature space. As a probabilistic alternative, Walker *et al.* [54] fused a conditioned Variational Autoencoder (cVAE)-based probabilistic pose predictor with a GAN. While the probabilistic predictor enhances the diversity in the predicted poses, the adversarial network ensures prediction realism. As this model struggles with long-term predictions, Fushishita *et al.* [183] addressed long-term video prediction of multiple outcomes avoiding the error accumulation and vanishing gradients by using a unidimensional CNN trained in an adversarial fashion. To enable multiple predictions, they have used additional inputs ensuring trajectory and behavior variability at a human pose level. To better preserve the visual appearance in the predictions than [53], [65], [108], Tang *et al.* [184] firstly predict human poses using a LSTM-based model to then synthesize pose-conditioned future frames using a combination of different networks: a global GAN modeling the time-invariant background and a coarse human pose, a local GAN refining the coarse-predicted human pose, and a 3D-AE to ensure temporal consistency across frames.

Keypoints-based representations: The keypoint coordinate space is a meaningful, tractable and structured representation for prediction, ensuring stable learning. It enforces model's internal representation to contain object-level information. This leads to better results on tasks requiring object-level understanding such as, trajectory prediction, action recognition and reward prediction. As keypoints are a natural representation of dynamic objects, Minderer *et al.* [85] reformulated the prediction task in the keypoint coordinate space. They proposed an AE architecture with a keypoint-based representational bottleneck, consisting of a VRNN that predicts dynamics in the keypoint space. Although this model qualitatively outperforms the Stochastic Video Generation (SVG) [81], Stochastic Adversarial Video Prediction (SAVP) [108] and EPVA [52] models, the quantitative evaluation reported similar results.

5.6 Incorporating Uncertainty

Although high-level representations significantly reduce the prediction space, the underlying distribution still has multiple modes. In other words, different plausible outcomes would be equally probable for the same input sequence. Addressing multimodal distributions is not straightforward for regression and classification approaches, as they regress to the mean and aim to discretize a continuous high-dimensional space, respectively. To deal with the inherent unpredictability of natural videos, some works introduced latent variables into existing deterministic models or directly relied on generative models such as GANs and VAEs.

Inspired by the DVF, Xue *et al.* [202] proposed a cVAE-based [222], [223] multi-scale model featuring a novel cross convolutional layer trained to regress the difference image or Eulerian motion [224]. Background on natural videos is not uniform, however the model implicitly assumes that the difference image would accurately capture the movement in foreground objects. Introducing latent variables into a convolutional AE, Goroshin *et al.* [211] proposed a probabilistic model for learning linearized feature representations to linearly extrapolate the predicted frame in a feature space. Uncertainty is introduced to the loss by using a cosine distance as an explicit curvature penalty. Authors focused on evaluating the linearization properties, yet the model was not contrasted to previous works. Extending [141], [202], Fragkiadaki *et al.* [96] proposed several architectural changes and training schemes to handle marginalization over stochastic variables, such as sampling from the prior and variational inference. They proposed a stochastic ED architecture that predicts future optical flow, i.e., dense pixel motion field, used to spatially transform the current frame into the next frame prediction. To introduce uncertainty

TABLE 2: Summary of video prediction models (**c**: **c**onvolutional; **r**: **r**ecurrent; **v**: **v**ariational; **ms**: **m**ulti-**s**cale; **st**: **st**acked; **bi**: **bi**directional; **P**: **P**ercepts; **M**: **M**otion; **PL**: **P**erceptual **L**oss; **AL**: **A**dversarial **L**oss; **S/R**: using **S**ynthetic/**R**eal datasets; **SS**: **S**emantic **S**egmentation; **D**: **D**epth; **S**: **S**tate; **Po**: **Po**se; **O**: **O**dometry; **IS**: **I**nstance **S**egmentation; **ms**: **m**ulti-**s**tep prediction; **pred-fr**: number of **pred**icted **fr**ames, $\star$ 1-5 frames, $\star\star$ 5-10 frames, $\star\star\star$ 10-100 frames, $\star\star\star\star$ over 100 frames; **ood**: indicates if model was tested on **o**ut-**o**f-**d**omain tasks).

					details				**evaluation**			
method	**year**	**based on**	**architecture**	**datasets (train, valid, test)**	**input**	**output**	**MS**	**loss function**	**S/R**	**pred-fr**	**ood**	**code**
Direct Pixel Synthesis												
Ranzato *et al.* [73]	2014	[145], [185]	rCNN	[115], [127]	RGB	RGB	✗	CE	R	$\star$	✗	✗
Srivastava *et al.* [74]	2015	[186]	LSTM-AE	[74], [113], [115], [123]	RGB,P	RGB	✓	CE, ℓ_2	SR	$\star\star\star$	✓	✓
PGN [49]	2015	-	LSTM-cED	[126]	RGB	RGB	✗	MSE, AL	S	$\star$	✗	✗
Shi *et al.* [13]	2015	[74]	cLSTM	[74]	RGB	RGB	✗	CE	S	$\star\star\star$	✓	✗
BeyondMSE [43]	2016	[60], [187]	msCNN	[115], [123]	RGB	RGB	✓	ℓ_1, GDL, AL	R	$\star\star$	✗	✓
PredNet [75]	2017	[22], [188]	stLSTMs	[117], [119], [120], [137]	RGB	RGB	✓	ℓ_1,ℓ_2	SR	$\star\star$	✓	✓
ContextVP [76]	2018	[88], [189]	MD-LSTM	[115], [117], [119], [120]	RGB	RGB	✓	ℓ_1, GDL	R	$\star\star$	✗	✗
fRNN [153]	2018	-	cGRU-AE	[74], [111], [115]	RGB	RGB	✓	ℓ_1	SR	$\star\star\star$	✗	✓
E3d-LSTM [66]	2019	[13]	r3D-CNN	[74], [111], [190], [191]	RGB	RGB	✓	ℓ_1, ℓ_2, CE	SR	$\star\star\star$	✓	✓
Kwon *et al.* [101]	2019	[45], [192], [193]	cycleGAN	[115], [119], [120], [194], [195]	RGB	RGB	✓	ℓ_1, LoG, AL	R	$\star\star\star$	✗	✗
Znet [70]	2019	[13]	cLSTM	[74], [111]	RGB	RGB	✓	ℓ_2, BCE, AL	SR	$\star\star\star$	✗	✗
VPGAN [57]	2019	[79], [193]	GAN	[111], [129]	RGB,Z	RGB	✓	ℓ_1, L_{cycle}, AL	R	$\star\star\star$	✗	✗
Jin *et al.* [150]	2020	-	cED-GAN	[111], [119], [120], [129]	RGB	RGB	✓	ℓ_2, GDL, AL	R	$\star\star\star$	✗	✗
Shouno *et al.* [151]	2020	[75]	GAN	[119], [120]	RGB	RGB	✓	L_p, AL, PL	R	$\star\star\star$	✗	✗
CrevNet [154]	2020	[13], [196], [197]	3d-cED	[74], [119], [120], [198]	RGB	RGB	✓	MSE	SR	$\star\star\star$	✓	✓
Using Explicit Transformations												
PGP [159]	2014	[157]	st-rGAEs	[126], [128]	RGB	RGB	✓	ℓ_2	SR	$\star$	✗	✗
Patraucean *et al.* [77]	2015	[186]	LSTM-cAE	[74], [113], [131], [132]	RGB	RGB	✗	ℓ_2, ℓ_δ	SR	$\star$	✓	✓
DFN [162]	2016	[89], [161]	r-cED	[74], [115]	RGB	RGB	✓	BCE	SR	$\star\star\star$	✓	✓
Amersfoort *et al.* [71]	2017	[77]	CNN	[74], [115]	RGB	RGB	✓	MSE	SR	$\star\star$	✗	✗
FSTN [78]	2017	[44], [77]	LSTM-cED	[74], [115], [123], [131], [132]	RGB	RGB	✓	ℓ_2, ℓ_δ, PL	SR	$\star\star\star$	✗	✗
Vondrick *et al.* [102]	2017	[67], [89]	cGAN	[125]	RGB	RGB	✓	CE, AL	R	$\star\star\star$	✓	✗
Chen *et al.* [50]	2017	[71], [160], [162]	rCNN-ED	[74], [115]	RGB	RGB	✓	CE, ℓ_2, GDL, AL	SR	$\star\star$	✗	✗
DVF [7]	2017	[160]	ms-cED	[115], [118]	RGB	RGB	✓	ℓ_1, TV	R	$\star$	✓	✓
SDC-Net [155]	2018	[199], [200]	CNN	[119], [124]	RGB,M	RGB	✓	ℓ_1, PL	SR	$\star\star$	✓	✗
TrIVD-GAN-FP [164]	2020	[142], [163], [167]	DVD-GAN	[115], [129], [201]	RGB	RGB	✓	L_{hinge} [55]	R	$\star\star\star$	✗	✗
Explicit Motion from Content Separation												
MCnet [65]	2017	[13], [166], [202]	LSTM-cED	[111], [112], [115], [123]	RGB	RGB	✓	ℓ_p, GDL, AL	R	$\star\star\star$	✗	✓
Dual-GAN [55]	2017	[100]	VAE-GAN	[115], [118]–[120]	RGB	RGB	✓	ℓ_1, KL, AL	R	$\star\star$	✗	✗
DRNET [79]	2017	[65]	LSTM-ED	[74], [111], [138], [203]	RGB	RGB	✓	ℓ_2, CE, AL	SR	$\star\star\star\star$	✓	✓
DPG [167]	2019	[89], [142]	cED	[119], [204], [205]	RGB	RGB	✓	ℓ_p, TV, PL, CE	SR	$\star\star$	✗	✗
Conditioned on Extra Variables												
Oh *et al.* [80]	2015	[13]	rED	[133]	RGB,A	RGB	✓	ℓ_2	S	$\star\star\star\star$	✓	✓
Finn *et al.* [89]	2016	[13], [80]	st-cLSTMs	[89], [117]	RGB,A,S	RGB	✓	ℓ_2	R	$\star\star\star$	✗	✓
In the High-level Feature Space												
Villegas *et al.* [53]	2017	[182], [206], [207]	LSTM-cED	[116], [117]	RGB,Po	RGB,Po	✓	ℓ_2, PL, AL [44]	R	$\star\star\star\star$	✓	✗
PEARL [51]	2017	-	cED	[121], [136]	RGB	SS	✗	ℓ_2, AL	R	$\star$	✓	✗
S2S [56]	2017	[43]	msCNN	[121], [136]	P	SS	✓	ℓ_1, GDL, AL	R	$\star\star\star$	✗	✓
Walker *et al.* [54]	2017	[208]	vED	[115], [116]	RGB,Po	RGB	✓	ℓ_2, CE, KL, AL	R	$\star\star\star$	✓	✗
Jin et al. [175]	2017	[51], [56], [77]	cED	[121], [137]	RGB,P	SS,M	✓	ℓ_1, GDL, CE	R	$\star\star\star$	✓	✗
EPVA (EPVA) [52]	2018	[53]	LSTM-ED	[117]	RGB	RGB	✓	ℓ_2, AL	SR	$\star\star\star\star$	✓	✓
Nabavi *et al.* [82]	2018	[56], [175]	biLSTM-cED	[121]	P	SS	✓	CE	R	$\star\star$	✗	✗
F2F *et al.* [8]	2018	[56], [181]	st-msCNN	[121]	P	P,SS,IS	✓	ℓ_2	R	$\star\star\star$	✓	✓
Vora *et al.* [83]	2018	-	LSTM	[121]	ego-M	ego-M	✗	ℓ_1	R	$\star$	✓	✗
Chiu *et al.* [174]	2019	-	3D-cED	[121], [122]	RGB	SS	✗	CE, MSE	R	$\star\star$	✗	✗
Bayes-WD-SL [9]	2019	[56], [175]	bayesResNet	[121]	SS,O	SS	✓	KL	SR	$\star\star\star$	✓	✓
Sun *et al.* [84]	2019	[8]	st-ms-cLSTM	[121], [134]	P	P,IS	✓	ℓ_2, [181]	R	$\star\star$	✗	✗
Terwilliger *et al.* [10]	2019	[65], [175]	M-cLSTM	[121]	RGB,P	SS	✓	CE, ℓ_1	R	$\star\star\star$	✗	✓
Struct-VRNN [85]	2019	[209], [210]	cVRNN	[90], [117]	RGB	RGB	✓	ℓ_2, KL	SR	$\star\star$	✓	✓
Incorporating Uncertainty												
Goroshin *et al.* [211]	2015	[212]	cAE	[138], [213]	RGB	RGB	✗	$\ell_2, penalty$	SR	$\star$	✗	✗
Fragkiadaki *et al.* [96]	2017	[141], [202]	vED	[117], [214]	RGB	RGB	✗	$KL, MCbest$	R	$\star$	✓	✗
EEN [99]	2017	[22], [75], [215]	vED	[216]–[218]	RGB	RGB	✓	ℓ_1, ℓ_2	SR	$\star\star$	✗	✓
SV2P [38]	2018	[89]	CDNA	[89], [117], [129]	RGB	RGB	✓	ℓ_p, KL	SR	$\star\star\star$	✗	✓
SVG [81]	2018	[38]	LSTM-cED	[74], [111], [129]	RGB	RGB	✓	ℓ_2, KL	SR	$\star\star\star\star$	✗	✓
Castrejon *et al.* [97]	2019	[81], [98]	vRNN	[74], [121], [129]	RGB	RGB	✓	KL	SR	$\star\star\star$	✗	✗
Hu *et al.* [15]	2020	[56], [163], [219]	cED	[121], [122], [220], [221]	RGB	SS,D,M	✓	$CE, \ell_\delta, L_d, L_c, L_p$	R	$\star\star\star$	✓	✗

in predictions, the authors proposed the k-best-sample-loss (MCbest) that draws K outcomes penalizing those similar to the ground-truth.

Incorporating latent variables into the deterministic CDNA architecture for the first time, Babaeizadeh *et al.* proposed the Stochastic Variational Video Prediction (SV2P) [38] model handling natural videos. Their time-invariant posterior distribution is approximated from the entire input video sequence. Authors demonstrated that, by explicitly modeling uncertainty with latent variables, the deterministic CDNA model is outperformed. By combining a standard deterministic architecture (LSTM-ED) with stochastic latent variables, Denton *et al.* proposed the SVG network [81]. Different from SV2P, the prior is sampled from a time-varying posterior distribution, i.e. it is a learned-prior instead of fixed-prior sampled from the same distribution. Most of the VAEs use a fixed Gaussian as a prior, sampling randomly at each time step. Exploiting the temporal dependencies, a learned-prior predicts high variance in uncertain situations, and a low variance when a deterministic prediction suffices. The SVG model is easier to train and reported sharper predictions in contrast to [38]. Built upon SVG, Villegas *et al.* [225] implemented a baseline to perform an in-depth empirical study on the importance of the inductive bias, stochasticity, and model's capacity in the video prediction task. Different from previous approaches, Henaff *et al.* proposed the Error Encoding Network (EEN) [99] that incorporates uncertainty by feeding back the residual error —the difference between the ground truth and the deterministic prediction— encoded as a low-dimensional latent variable. In this way, the model implicitly separates the input video into deterministic and stochastic components.

On the one hand, latent variable-based approaches cover the space of possible outcomes, yet predictions lack of realism. On the other hand, GANs struggle with uncertainty, but predictions are more realistic. Searching for a trade-off between VAEs and GANs, Lee *et al.* [108] proposed the SAVP model, being the first to combine latent variable models with GANs to improve variability in video predictions, while maintaining realism. Under the assumption that blurry predictions of VAEs are a sign of underfitting, Castrejon *et al.* extended the VRNNs to leverage a hierarchy of latent variables and better approximate data likelihood [97]. Although the backpropagation through a hierarchy of conditioned latents is not straightforward, several techniques alleviated this issue such as, KL beta warm-up, dense connectivity pattern between inputs and latents, Ladder Variational Autoencoders (LVAEs) [226]. As most of the probabilistic approaches fail in approximating the true distribution of future frames, Pottorff *et al.* [227] reformulated the video prediction task without making any assumption about the data distribution. They proposed the Invertible Linear Embedding (ILE) enabling exact maximum likelihood learning of video sequences, by combining an invertible neural network [228], also known as reversible flows, and a linear time-invariant dynamic system. The ILE handles nonlinear motion in the pixel space and scales better to longer-term predictions compared to adversarial models [43].

While previous variational approaches [81], [108] focused on predicting a single frame of low resolution in restricted, predictable or simulated datasets, Hu *et al.* [15] jointly predict full-frame ego-motion, static scene, and object dynamics on complex real-world urban driving. Featuring a novel spatio-temporal module, their five-component architecture learns rich representations that incorporate both local and global spatio-temporal context. Authors validated the model on predicting semantic segmentation, depth and optical flow, two seconds in the future outperforming existing spatio-temporal architectures. However, no performance comparison with [81], [108] has been carried out.

6 Performance Evaluation

This section presents the results of the previously analyzed video prediction models on the most popular datasets on the basis of the metrics described below.

6.1 Metrics and Evaluation Protocols

For a fair evaluation of video prediction systems, multiple aspects in the prediction have to be addressed such as whether the predicted sequences look realistic, are plausible and cover all possible outcomes. To the best of our knowledge, there are no evaluation protocols and metrics that evaluate the predictions by fulfilling simultaneously all these aspects.

The most widely used evaluation protocols for video prediction rely on image similarity-based metrics such as, Mean-Squared Error (MSE), Structural Similarity Index Measure (SSIM) [229], and Peak Signal to Noise Ratio (PSNR). However, evaluating a prediction according to the mismatch between its visual appearance and the ground truth is not always reliable. In practice, these metrics penalize all predictions that deviate from the ground truth. In other words, they prefer blurry predictions nearly accommodating the exact ground truth than sharper and plausible but imperfect generations [97], [108], [230]. Pixel-wise metrics do not always reflect how accurate a model captured video scene dynamics and their temporal variability. In addition, the success of a metric is influenced by the loss function used to train the model. For instance, the models trained with MSE loss function would obviously perform well on MSE metric, but also on PSNR metric as it is based on MSE. Suffering from similar problems, SSIM measures the similarity between two images, from -1 (very dissimilar) to $+1$ (the same image). As a difference, it measures similarities on image patches instead of performing pixel-wise comparison. These metrics are easily fooled by learning to match the background in predictions. To address this issue, Mathieu *et al.* [43] evaluated the predictions only on the dynamic parts of the sequence, avoiding background influence.

As the pixel space is multimodal and highly-dimensional, it is challenging to evaluate how accurately a prediction sequence covers the full distribution of possible outcomes. Addressing this issue, some probabilistic approaches [81], [97], [108] adopted a different evaluation protocol to assess prediction coverage. Basically, they sample multiple random predictions and then they search for the best match with the ground truth sequence. Finally, they report the best match using common metrics. This

TABLE 3: Results on M-MNIST (Moving MNIST). Predicting the next y frames from x context frames ($x \rightarrow y$). † results reported by Oliu *et al.* [153], ‡ results reported by Wang *et al.* [66], ∗ results reported by Wang *et al.* [197], ◁ results reported by Wang *et al.* [235]. **MSE** represents per-pixel average MSE (10^{-3}). **MSE**◇ represents per-frame error.

	M-MNIST (10 → 10)					M-MNIST (10 → 30)	
method	**MSE**	**MSE◇**	**SSIM**	**PSNR**	**CE**	**MSE◇**	**SSIM**
BeyondMSE [43]	27.48†	122.6∗	0.713∗	15.969†	-	-	-
Srivastava *et al.* [74]	17.37†	118.3∗	0.690∗	18.183†	341.2	180.1◁	0.583◁
Shi *et al.* [13]	-	96.5‡	0.713‡	-	367.2∗	156.2◁	0.597◁
DFN [162]	-	89.0‡	0.726‡	-	285.2	149.5◁	0.601◁
CDNA [89]	-	84.2‡	0.728‡	-	346.6∗	142.3◁	0.609◁
VLN [236]	-	-	-	-	187.7		
Patraucean *et al.* [77]	43.9	-	-	-	179.8	-	-
MCnet [65]†	42.54	-	-	13.857	-	-	-
RLN [237]†	42.54	-	-	13.857	-	-	-
PredNet [75]†	41.61	-	-	13.968	-	-	-
fRNN [153]	**9.47**	68.4‡	0.819‡	**21.386**	-	-	-
PredRNN [197]	-	56.8	0.867	-	97.0	-	-
VPN [95]	-	64.1‡	0.870‡	-	**87.6**	129.6◁	0.620◁
Znet [70]	-	50.5	0.877	-	-	-	-
PredRNN++ [235]	-	46.5	0.898	-	-	**91.1**	**0.733**
E3d-LSTM [66]	-	41.3	0.910	-	-	-	-
CrevNet [154]	-	**22.3**	**0.949**	-	-	-	-

TABLE 4: Results on KTH dataset. Predicting the next y frames from x context frames ($x \rightarrow y$). † results reported by Oliu *et al.* [153], ‡ results reported by Wang *et al.* [66], ∗ results reported by Zhang *et al.* [70], ◁ results reported by Jin *et al.* [150]. Per-pixel average MSE (10^{-3}). Best results are represented in bold.

	KTH (10 → 10)		KTH (10 → 20)		KTH (10 → 40)	
method	**MSE**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**
Srivastava *et al.* [74]†	9.95	21.22	-	-	-	-
PredNet [75]†	3.09	28.42	-	-	-	-
BeyondMSE [43]†	1.80	29.34	-	-	-	-
fRNN [153]	1.75	29.299	0.771◁	26.12◁	0.678◁	23.77◁
MCnet [65]	1.65†	30.95†	0.804‡	25.95‡	0.73◁	23.89◁
RLN [237]†	**1.39**	**31.27**	-	-	-	-
Shi *et al.* [13]‡	-	-	0.712	23.58	0.639	22.85
SAVP [108]◁	-	-	0.746	25.38	0.701	23.97
VPN [95]∗	-	-	0.746	23.76	-	-
DFN [162]‡	-	-	0.794	27.26	0.652	23.01
fRNN [153]‡	-	-	0.771	26.12	0.678	23.77
Znet [70]	-	-	0.817	27.58	-	-
SV2P invariant [38]◁	-	-	0.826	27.56	0.778	25.92
SV2P variant [38]◁	-	-	0.838	27.79	0.789	26.12
PredRNN [197]	-	-	0.839	27.55	0.703‡	24.16‡
VarNet [238]◁	-	-	0.843	28.48	0.739	25.37
SAVP-VAE [108]◁	-	-	0.852	27.77	0.811	26.18
PredRNN++ [235]	-	-	0.865	28.47	0.741‡	25.21‡
MSNET [239]	-	-	0.876	27.08	-	-
E3d-LSTM [66]	-	-	0.879	29.31	0.810	27.24
Jin *et al.* [150]	-	-	**0.893**	**29.85**	**0.851**	**27.56**

represents the most common evaluation protocol for probabilistic video prediction. Other methods [97], [150], [151] also reported results using: LPIPS [230] as a perceptual metric comparing CNN features, or Frchet Video Distance (FVD) [231] to measure sample realism by comparing underlying distributions of predictions and ground truth. Moreover, Lee *et al.* [108] used the VGG Cosine Similarity metric that performs cosine similarity to the features extracted with the VGGnet [146] from the predictions.

Some other alternative metrics include the inception score [232] introduced to deal with GANs mode collapse problem by measuring the diversity of generated samples; perceptual similarity metrics, such as DeePSiM [44]; measuring sharpness based on difference of gradients [43]; Parzen window [233], yet deficient for high-dimensional images; and the Laplacian of Gaussians (LoG) [60], [234] used in [101]. In the semantic segmentation space, authors used the popular Intersection over Union (IoU) metric. Inception score was also widely used to report results on different methods [54], [65], [67], [79]. Differently, on the basis of the EPVA model [52] a quantitative evaluation was performed, based on the confidence of an external method trained to identify whether the generated video contains a recognizable person. While some authors [10], [43], [56] evaluated the performance only on the dynamic parts of the image, other directly opted for visual human evaluation through Amazon Mechanical Turk (AMT) workers, without a direct quantitative evaluation.

6.2 Results

In this section we report the quantitative results of the most relevant methods reviewed in the previous sections. To achieve a wide comparison, we limited the quantitative results to the most common metrics and datasets. We have distributed the results in different tables, given the large variation in the evaluation protocols of the video prediction models.

Many authors evaluated their methods on the Moving MNIST synthetic environment. Although it represents a restricted and quasi-deterministic scenario, long-term predictions are still challenging. The black and homogeneous background induce methods to accurately extrapolate black frames and vanish the predicted digits in the long-term horizon. Under this configuration, the CrevNet model demonstrated a leap over the previous state of the art. As the second best, the E3d-LSTM network reported stable errors in both short-term and longer-term predictions showing the advantages of their memory attention mechanism. It also reported the second best results on the KTH dataset, after [150] which achieved the best overall performance and demonstrated quality predictions on natural videos.

Performing short-term predictions in the KTH dataset, the Recurrent Ladder Network (RLN) outperformed MCnet and fRNN by a slight margin. The RLN architecture draws similarities with fRNN, except that the former uses bridge connections and the latter, state sharing that improves memory consumption. On the Moving MNIST and UCF101 datasets, fRNN outperformed RLN. Other interesting methods to highlight are PredRNN and PredRNN++, both providing close results to E3d-LSTM. State-of-the-art results using different metrics were reported on Caltech Pedestrian by Kwon *et al.* [101], CrevNet [154], and Jin *et al.* [150]. The former, by taking advantage of its retrospective prediction scheme, was also the overall winner on the UCF-101 dataset meanwhile the latter outperformed previous

TABLE 5: Results on Caltech Pedestrian. Predicting the next y frames from x context frames ($x \rightarrow y$). † reported by Kwon *et al.* [101], ‡ reported by Reda *et al.* [155], $*$ reported by Gao *et al.* [167], $\triangleleft$ reported by Jin *et al.* [150]. Per-pixel average MSE (10^{-3}). Best results are represented in bold.

	Caltech Pedestrian			
	(10 → 1)			
method	**MSE**	**SSIM**	**PSNR**	**LPIPS**
BeyondMSE [43]‡	3.42	0.847	-	-
MCnet [65]‡	2.50	0.879	-	-
DVF [7]$*$	-	0.897	26.2	5.57$\triangleleft$
Dual-GAN [55]	2.41	0.899	-	-
CtrlGen [142]$*$	-	0.900	26.5	6.38$\triangleleft$
PredNet [75]†	2.42	0.905	27.6	7.47$\triangleleft$
ContextVP [76]	1.94	0.921	28.7	6.03$\triangleleft$
GAN-VGG [151]	-	0.916	-	3.61
G-VGG [151]	-	0.917	-	**3.52**
SDC-Net [155]	1.62	0.918	-	-
Kwon et al. [101]	**1.61**	0.919	29.2	-
DPG [167]	–	0.923	28.2	5.04$\triangleleft$
G-MAE [151]	-	0.923	-	4.30
GAN-MAE [151]	-	0.923	-	4.09
CrevNet [154]	-	0.925	**29.3**	-
Jin *et al.* [150]	-	**0.927**	29.1	5.89

TABLE 6: Results on UCF-101 dataset. Predicting the next x frames from y context frames ($x \rightarrow y$). † results reported by Oliu *et al.* [153]. Per-pixel average MSE (10^{-3}). Best results are represented in bold.

	UCF-101		**UCF-101**		
	(10 → 10)		**(4 → 1)**		
method	**MSE**	**PSNR**	**MSE**	**SSIM**	**PSNR**
Srivastava *et al.* [74]†	148.66	10.02	-	-	-
PredNet [75]†	15.50	19.87	-	-	-
BeyondMSE [43]†	9.26	22.78	-	-	-
MCnet [65]	9.40†	23.46†	-	0.91	31.0
RLN [237]†	9.18	23.56	-	-	-
fRNN [153]	**9.08**	**23.87**	-	-	-
BeyondMSE [43]	-	-	-	0.92	32
Dual-GAN [55]	-	-	-	**0.94**	30.5
DVF [7]	-	-	-	**0.94**	33.4
ContextVP [76]	-	-	-	0.92	34.9
Kwon et al. [101]	-	-	**1.37**	**0.94**	**35.0**

methods on the BAIR Push dataset.

On the one hand, some approaches have been evaluated on other datasets: SDC-Net [155] outperformed [43], [65] on YouTube8M, TrIVD-GAN-FP outperformed [163], [240] on Kinetics-600 test set [201], E3d-LSTM compared their method with [95], [153], [197], [235] on the TaxiBJ dataset [190], and CrevNet [154] on Traffic4cast [198]. On the other hand, some explored out-of-domain tasks [13], [66], [102], [154], [162] (see ood column in Table 2).

6.2.1 Results on Probabilistic Approaches

Video prediction probabilistic methods have been mainly evaluated on the Stochastic Moving MNIST, Bair Push and Cityscapes datasets. Different from the original Moving MNIST dataset, the stochastic version includes uncertain digit trajectories, i.e. the digits bounce off the border with a random new direction. On this dataset, both versions of Castrejon *et al.* models (1L, without a hierarchy of latents, and 3L with a 3-level hierarchy of latents) outperform SVG by a large margin. On the Bair Push dataset, SAVP reported sharper and more realistic-looking predictions than SVG which suffer of blurriness. However, both models were outperformed by [97] as well on the Cityscapes dataset. The model based on a 3-level hierarchy of latents [97] outperform previous works on all three datasets, showing the advantages of the extra expressiveness of this model.

TABLE 7: Results on SM-MNIST (Stochastic Moving MNIST), BAIR Push and Cityscapes datasets. † results reported by Castrejon *et al.* [97]. ‡ results reported by Jin *et al.* [150].

	SM-MNIST		**BAIR Push**			**Cityscapes**	
	(5 → 10)		**(2 → 28)**			**(2 → 28)**	
method	**FVD**	**SSIM**	**FVD**	**SSIM**	**PSNR**	**FVD**	**SSIM**
SVG [81]	90.81†	0.688†	256.62†	0.816†	17.72‡	1300.26†	0.574†
SAVP [108]	-	-	143.43†	0.795†	18.42‡	-	-
SAVP-VAE [108]	-	-	-	0.815‡	19.09‡	-	-
SV2P inv. [38]‡	-	-	-	0.817	20.36	-	-
vRNN 1L [97]	63.81	**0.763**	149.22	0.829	-	682.08	0.609
vRNN 3L [97]	**57.17**	0.760	**143.40**	0.822	-	**567.51**	**0.628**
Jin *et al.* [150]	-	-	-	**0.844**	**21.02**	-	-

TABLE 8: Results on Cityscapes dataset. Predicting the next y time-steps of semantic segmented frames from 4 context frames ($4 \rightarrow y$). ‡ IoU results on eight moving objects classes. † results reported by Chiu *et al.* [174]

	Cityscapes			
	(4 → 1)	(4 → 3)	(4 → 9)	(4 → 10)
method	**IoU**	**IoU**	**IoU**	**IoU**
S2S [56]‡	-	55.3	40.8	-
S2S-maskRCNN [8]‡	-	55.4	42.4	-
S2S [56]	62.60‡	59.4	47.8	-
Nabavi *et al.* [82]	71.37	60.06	-	-
F2F [8]	-	61.2	41.2	-
Vora *et al.* [83]	-	61.47	45.4	-
S2S-Res101-FCN [175]	-	62.6	-	50.8
Terwilliger *et al.* [10]‡	-	65.1	46.3	-
Chiu *et al.* [174]	72.43	65.53	50.52	
Jin *et al.* [175]	-	66.1	-	**53.9**
Bayes-WD-SL [9]	**75.3**	66.7	**52.5**	-
Terwilliger *et al.* [10]	73.2	**67.1**	51.5	52.5

6.2.2 Results on the High-level Prediction Space

Most of the methods have chosen the semantic segmentation space to make predictions. Although they relied on different datasets for training, performance results were mostly reported on the Cityscapes dataset using the IoU metric. Authors explored short-term (next-frame prediction), mid-term (+3 time steps in the future) and long-term (up to +10 time step in the future) predictions. On the semantic segmentation prediction space, Bayes-WD-SL [9], the model proposed by Terwilliger *et al.* [10], and Jin *et al.* [51] reported the best results. Among these methods, it is noteworthy that Bayes-WD-SL was the only one to explore prediction diversity on the basis of a Bayesian formulation.

In the instance segmentation space, the F2F pioneering method [8] was outperformed by Sun *et al.* [84] on short and mid-term predictions using the AP50 and AP evaluation metrics. On the other hand, in the keypoint coordinate space, the seminal model of Minderer *et al.* [85] qualitatively outperforms SVG [81], SAVP [108] and EPVA [52], yet pixel-wise metrics reported similar results. In the human pose space, Tang *et al.* [184] by regressing future frames from human pose predictions outperformed SAVP [108], MCnet [65] and [53] on the basis of the PSNR and SSIM metrics on the Penn Action and J-HMDB [114] datasets.

7 Discussion

The video prediction literature ranges from a direct synthesis of future pixel intensities, to complex probabilistic models addressing prediction uncertainty. The range between these approaches consists of methods that try to factorize or narrow the prediction space. Simplifying the prediction task has been a natural evolution of video prediction models, influenced by several open research challenges discussed below. Due to the curse of dimensionality and the inherent pixel variability, developing a robust prediction based on raw pixel intensities is overly-complicated. This often leads to the regression-to-the-mean problem, visually represented as blurriness. Making parametric models larger would improve the quality of predictions, yet this is currently incompatible with high-resolution predictions due to memory constraints. Transformation-based approaches propagate pixels from previous frames based on estimated flow maps. In this case, prediction quality is directly influenced by the accuracy of the estimated flow. Similarly, the prediction in a high-level space is mostly conditioned by the quality of some extra supervisory signals such as semantic maps and human poses, to name a few. Erroneous supervision signals would harm prediction quality.

Analyzing the impact of the inductive bias on the performance of a video prediction model, Villegas *et al.* [225] demonstrated the maximization of the SVG model [81] performance with minimal inductive bias (e.g. segmentation or instance maps, optical flow, adversarial losses, etc.) by increasing progressively the scale of computation. A common assumption when addressing the prediction task in a high-level feature space, is the direct improvement of long-term predictions as a result of simplifying the prediction space. Even if the complexity of the prediction space is reduced, it is still multimodal when dealing with natural videos. For instance, when it comes to long-term predictions in the semantic segmentation space, most of the models reported predictions only up to ten time steps into the future. This directly suggests that the choice of the prediction space is still an unsolved problem. Finding a trade-off between the complexity of the prediction space and the output quality is challenging. An overly-simplified representation could limit the prediction on complex data such as natural videos. Although abstract predictions suffice for many of the decision-making systems based on visual reasoning, prediction in pixel space is still being addressed.

From the analysis performed in this review and in line with the conclusions extracted from [225] we state that: (1) including recurrent connections and stochasticity in a video prediction model generally lead to improved performance; (2) increasing model capacity while maintaining a low inductive bias also improves prediction performance; (3) multi-step predictions conditioned by previously generated outputs are prone to accumulate errors, diverging from the ground truth when addressing long-term horizons; (4) authors predicted further in the future without relying on high-level feature spaces; (5) combining pixel-wise losses with adversarial training somewhat mitigates the regression-to-the-mean issue.

7.1 Research Challenges

Despite the wealth of currently existing video prediction approaches and the significant progress made in this field, there is still room to improve state-of-the-art algorithms. To foster progress, open research challenges must be clearly identified and disentangled. So far in this review, we have already discussed about: (1) the importance of spatio-temporal correlations as a self-supervisory signal for predictive models; (2) how to deal with future uncertainty and model the underlying multimodal distributions of natural videos; (3) the over-complicated task of learning meaningful representations and deal with the curse of dimensionality; (4) pixel-wise loss functions and blurry results when dealing with equally probable outcomes, i.e. probabilistic environments. These issues define the open research challenges in video prediction.

Currently existing methods are limited to short-term horizons. While frames in the immediate future are extrapolated with high accuracy, in the long term horizon the prediction problem becomes multimodal by nature. Initial solutions consisted on conditioning the prediction on previously predicted frames. However, these autoregressive models tend to accumulate prediction errors that progressively diverge the generated prediction from the expected outcome. On the other hand, due to memory issues, there is a lack of resolution in predictions. Authors tried to address this issue by composing the full-resolution image from small predicted patches. However, as the results are not convincing because of the annoying tilling effect, most of the available models are still limited to low-resolution predictions. In addition to the lack of resolution and long-term predictions, models are still prone to the regress-to-the-mean problem that consists on averaging the output frame to accommodate multiple equally probable outcomes. This is directly related to the pixel-wise loss functions, that focus the learning process on the visual appearance. The choice of the loss function is an open research problem with a direct influence on the prediction quality. Finally, the lack of reliable and fair evaluation models makes the qualitative evaluation of video prediction challenging and represents another potential open problem.

7.2 Future Directions

Based on the reviewed research identifying the state-of-the-art video prediction methods, we present some future promising research directions.

Consider alternative loss functions: Pixel-wise loss functions are widely used in the video prediction task, causing blurry predictions when dealing with uncontrolled environments or long-term horizon. In this regard, great efforts have been made in the literature for identifying more suitable loss functions for the prediction task. However, despite the existing wide spectrum of loss functions, most models still blindly rely on deterministic loss functions.

Alternatives to RNNs: Currently, RNNs are still widely used in this field to model temporal dependencies, and achieved state-of-the-art results on different benchmarks [66], [153], [197], [235]. Nevertheless, some methods also relied on 3D convolutions to further enhance video prediction [66], [174] representing a promising avenue.

Use synthetically generated videos: Simplifying the prediction is a current trend in the video prediction literature. A vast amount of video prediction models explored higher-level features spaces to reformulate the prediction task into a more tractable problem. However, this mostly conditions the prediction to the accuracy of an external source of supervision such as optical flow, human pose, pre-activations (percepts) extracted from supervised networks, and more. However, this issue could be alleviated by taking advantage of existing fully-annotated and photorealistic synthetic datasets or by using data generation tools. Video prediction in photorealistic synthetic scenarios has not been explored in the literature.

Evaluation metrics: Since the most widely used evaluation protocols for video prediction rely on image similarity-based metrics, the need for fairer evaluation metrics is imminent. A fair metric should not penalize predictions that deviate from the ground truth at the pixel level, if their content represents a plausible future prediction in a higher level, i.e., the dynamics of the scene correspond to the reality of the labels. In this regard, some methods evaluate the similarity between distributions or at a higher-level. However, there is still room for improvement in the evaluation protocols for video prediction and generation [241].

8 Conclusion

In this review, after reformulating the predictive learning paradigm in the context of video prediction, we have closely reviewed the fundamentals on which it is based: exploiting the time dimension of videos, dealing with stochasticity, and the importance of the loss functions in the learning process. Moreover, an analysis of the backbone deep learning-based architectures for this task was performed in order to provide the reader the necessary background knowledge. The core of this study encompasses the analysis and classification of more than 50 methods and the datasets they have used. Methods were analyzed from three perspectives: method description, contribution over the previous works and performance results. They have also been classified according to a proposed taxonomy based on their main contribution. In addition, we have presented a comparative summary of the datasets and methods in tabular form so as the reader, at a glance, could identify low-level details. In the end, we have discussed the performance results on the most popular datasets and metrics to finally provide useful insight in shape of future research directions and open problems. In conclusion, video prediction is a promising avenue for the self-supervised learning of rich spatio-temporal correlations to provide prediction capabilities to existing intelligent decision-making systems. While great strides have been made, there is still room for improvement in video prediction using deep learning techniques.

Acknowledgments

This work has been funded by the Spanish Government PID2019-104818RB-I00 grant for the MoDeaAS project. This work has also been supported by two Spanish national grants for PhD studies, FPU17/00166, and ACIF/2018/197 respectively. Experiments were made possible by a generous hardware donation from NVIDIA.

References

[1] M. H. Nguyen and F. D. la Torre, “Max-margin early event detectors,” in *CVPR*, 2012.

[2] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, “Activity Forecasting,” in *ECCV*, 2012.

[3] C. Vondrick, H. Pirsiavash, and A. Torralba, “Anticipating Visual Representations from Unlabeled Video,” in *CVPR*, 2016.

[4] K. Zeng, W. B. Shen, D. Huang, M. Sun, and J. C. Niebles, “Visual Forecasting by Imitating Dynamics in Natural Sequences,” in *ICCV*, 2017.

[5] S. Shalev-Shwartz, N. Ben-Zrihem, A. Cohen, and A. Shashua, “Long-term planning by short-term prediction,” *arXiv:1602.01580*, 2016.

[6] O. Makansi, E. Ilg, O. Cicek, and T. Brox, “Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction,” in *CVPR*, 2019.

[7] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, “Video frame synthesis using deep voxel flow,” in *ICCV*, 2017.

[8] P. Luc, C. Couprie, Y. LeCun, and J. Verbeek, “Predicting Future Instance Segmentation by Forecasting Convolutional Features,” in *ECCV*, 2018, pp. 593–608.

[9] A. Bhattacharyya, M. Fritz, and B. Schiele, “Bayesian prediction of future street scenes using synthetic likelihoods,” in *ICLR*, 2019.

[10] A. Terwilliger, G. Brazil, and X. Liu, “Recurrent flow-guided semantic forecasting,” in *WACV*, 2019.

[11] A. Bhattacharyya, M. Fritz, and B. Schiele, “Long-Term On-Board Prediction of People in Traffic Scenes Under Uncertainty,” in *CVPR*, 2018.

[12] W. Liu, W. Luo, D. Lian, and S. Gao, “Future frame prediction for anomaly detection - A new baseline,” in *CVPR*. IEEE, 2018.

[13] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *NeurIPS*, 2015.

[14] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO, “Deep learning for precipitation nowcasting: A benchmark and a new model,” in *NeurIPS*, 2017.

[15] A. Hu, F. Cotter, N. Mohan, C. Gurau, and A. Kendall, “Probabilistic future prediction for video scene understanding,” *arXiv:2003.06409*, 2020.

[16] A. Garcia-Garcia, P. Martinez-Gonzalez, S. Oprea, J. A. Castro-Vargas, S. Orts-Escolano, J. Garcia-Rodriguez, and A. Jover-Alvarez, “The robotrix: An extremely photorealistic and very-large-scale indoor dataset of sequences with robot trajectories and interactions,” in *IROS*, 2018, pp. 6790–6797.

[17] Y. Kong and Y. Fu, “Human action recognition and prediction: A survey,” *arXiv:1806.11230*, 2018.

[18] C. Sahin, G. Garcia-Hernando, J. Sock, and T. Kim, “A review on object pose recovery: from 3d bounding box detectors to full 6d pose estimators,” *arXiv:2001.10609*, 2020.

[19] V. Villena-Martinez, S. Oprea, M. Saval-Calvo, J. A. López, A. F. Guilló, and R. B. Fisher, “When deep learning meets data alignment: A review on deep registration networks (DRNs),” *arXiv:2003.03167*, 2020.

[20] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015.

[21] J. Hawkins and S. Blakeslee, *On Intelligence*. Times Books, 2004.

[22] R. P. N. Rao and D. H. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature Neuroscience*, vol. 2, no. 1, 1999.

[23] D. Mumford, "On the computational architecture of the neocortex," *Biological Cybernetics*, vol. 66, no. 3, 1992.

[24] A. Cleeremans and J. L. McClelland, "Learning the structure of event sequences." *Journal of Experimental Psychology: General*, vol. 120, no. 3, 1991.

[25] A. Cleeremans and J. Elman, *Mechanisms of implicit learning: Connectionist models of sequence processing*. MIT press, 1993.

[26] R. Baker, M. Dexter, T. E. Hardwicke, A. Goldstone, and Z. Kourtzi, "Learning to predict: Exposure to temporal sequences facilitates prediction of future events," *Vision Research*, vol. 99, 2014.

[27] H. E. M. den Ouden, P. Kok, and F. P. de Lange, "How prediction errors shape perception, attention, and motivation," in *Front. Psychology*, 2012.

[28] W. R. Softky, "Unsupervised pixel-prediction," in *NeurIPS*, 1995.

[29] G. Deco and B. Schürmann, "Predictive coding in the visual cortex by a recurrent network with gabor receptive fields," *Neural Processing Letters*, vol. 14, no. 2, 2001.

[30] A. Hollingworth, "Constructing visual representations of natural scenes: the roles of short- and long-term visual memory." *Journal of experimental psychology. Human perception and performance*, vol. 30 3, 2004.

[31] Y. Bengio, A. C. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *Trans. on PAMI*, vol. 35, no. 8, 2013.

[32] X. Wang and A. Gupta, "Unsupervised Learning of Visual Representations Using Videos," in *ICCV*, 2015.

[33] P. Agrawal, J. Carreira, and J. Malik, "Learning to see by moving," in *ICCV*, 2015.

[34] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, and J. Carlos Niebles, "What makes a video a video: Analyzing temporal information in video understanding models and datasets," in *CVPR*, June 2018.

[35] L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Schölkopf, and W. T. Freeman, "Seeing the arrow of time," in *CVPR*, 2014.

[36] D. Wei, J. J. Lim, A. Zisserman, and W. T. Freeman, "Learning and using the arrow of time," in *CVPR*, 2018.

[37] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: Unsupervised learning using temporal order verification," in *ECCV*, 2016.

[38] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," in *ICLR*, 2018.

[39] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Computational Imaging*, vol. 3, no. 1, 2017.

[40] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *arXiv:1702.05659*, 2017.

[41] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *CVPR*, 2017.

[42] J.-J. Hwang, T.-W. Ke, J. Shi, and S. X. Yu, "Adversarial structure matching for structured prediction tasks," in *CVPR*, 2019.

[43] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *ICLR (Poster)*, 2016.

[44] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *NIPS*, 2016.

[45] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*, vol. 9906, 2016.

[46] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *CVPR*, 2017.

[47] M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch, "Enhancenet: Single image super-resolution through automated texture synthesis," in *ICCV*, 2017.

[48] J. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *ECCV*, ser. Lecture Notes in Computer Science, vol. 9909, 2016.

[49] W. Lotter, G. Kreiman, and D. D. Cox, "Unsupervised learning of visual structure using predictive generative networks," *arXiv:1511.06380*, 2015.

[50] X. Chen, W. Wang, J. Wang, and W. Li, "Learning object-centric transformation for video prediction," in *ACM-MM*, ser. MM '17. New York, NY, USA: ACM, 2017.

[51] X. Jin, X. Li, H. Xiao, X. Shen, Z. Lin, J. Yang, Y. Chen, J. Dong, L. Liu, Z. Jie, J. Feng, and S. Yan, "Video Scene Parsing with Predictive Feature Learning," in *ICCV*, 2017.

[52] N. Wichers, R. Villegas, D. Erhan, and H. Lee, "Hierarchical long-term video prediction without supervision," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 80, 2018.

[53] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *ICML*, 2017.

[54] J. Walker, K. Marino, A. Gupta, and M. Hebert, "The pose knows: Video forecasting by generating pose futures," in *ICCV*, 2017.

[55] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion GAN for future-flow embedded video prediction," in *ICCV*, 2017.

[56] P. Luc, N. Neverova, C. Couprie, J. Verbeek, and Y. LeCun, "Predicting Deeper into the Future of Semantic Segmentation," in *ICCV*, 2017.

[57] Z. Hu and J. Wang, "A novel adversarial inference framework for video prediction with action control," in *ICCV Workshops*, Oct 2019.

[58] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, Nov 1998.

[59] V. Jain, J. F. Murray, F. Roth, S. C. Turaga, V. P. Zhigulin, K. L. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung, "Supervised learning of image restoration with convolutional networks," in *ICCV*, 2007.

[60] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," in *NeurIPS*, 2015.

[61] F. Yu, V. Koltun, and T. A. Funkhouser, "Dilated residual networks," in *CVPR*. IEEE, 2017.

[62] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *TPAMI*, vol. 40, no. 4, 2018.

[63] W. Luo, Y. Li, R. Urtasun, and R. S. Zemel, "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks," in *NeurIPS*, 2016.

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[65] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," in *ICLR*, 2017.

[66] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, and L. Fei-Fei, "Eidetic 3d LSTM: A model for video prediction and beyond," in *ICLR*, 2019.

[67] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating Videos with Scene Dynamics," in *NeurIPS*, 2016.

[68] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "MoCoGAN: Decomposing motion and content for video generation," in *CVPR*, June 2018.

[69] S. Aigner and M. Körner, "Futuregan: Anticipating the future frames of video sequences using spatio-temporal 3d convolutions in progressively growing autoencoder gans," *arXiv:1810.01325*, 2018.

[70] J. Zhang, Y. Wang, M. Long, W. Jianmin, and P. S. Yu, "Z-order recurrent neural networks for video prediction," in *ICME*, July 2019.

[71] J. R. van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala, "Transformation-based models of video sequences," *arXiv:1701.08435*, 2017.

[72] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, 1986.

[73] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, "Video (language) modeling: a baseline for generative models of natural videos," *arXiv:1412.6604*, 2014.

[74] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised Learning of Video Representations using LSTMs," in *ICML*, 2015.

[75] W. Lotter, G. Kreiman, and D. Cox, "Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning," in *ICLR (Poster)*, 2017.

[76] W. Byeon, Q. Wang, R. K. Srivastava, and P. Koumoutsakos, "Contextvp: Fully context-aware video prediction," in *CVPR (Workshops)*, 2018.

[77] V. Patraucean, A. Handa, and R. Cipolla, "Spatio-temporal video autoencoder with differentiable memory," *(ICLR) Workshop*, 2015.

[78] C. Lu, M. Hirsch, and B. Schölkopf, "Flexible Spatio-Temporal Networks for Video Prediction," in *CVPR*, 2017.

[79] E. L. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from video," in *NeurIPS*, 2017.

[80] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. P. Singh, "Action-Conditional Video Prediction using Deep Networks in Atari Games," in *NeurIPS*, 2015.

[81] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," in *ICML*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80, 2018.

[82] S. shahabeddin Nabavi, M. Rochan, and Y. Wang, "Future Semantic Segmentation with Convolutional LSTM," in *BMVC*, 2018.

[83] S. Vora, R. Mahjourian, S. Pirk, and A. Angelova, "Future segmentation using 3d structure," *arXiv:1811.11358*, 2018.

[84] J. Sun, J. Xie, J. Hu, Z. Lin, J. Lai, W. Zeng, and W. Zheng, "Predicting future instance segmentation with contextual pyramid convLSTMs," in *ACM Multimedia*. ACM, 2019.

[85] M. Minderer, C. Sun, R. Villegas, F. Cole, K. P. Murphy, and H. Lee, "Unsupervised learning of object structure and dynamics from videos," in *NeurIPS*, 2019.

[86] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, 1997.

[87] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *EMNLP*, pp. 1724–1734, 2014.

[88] A. Graves, S. Fernández, and J. Schmidhuber, "Multi-dimensional recurrent neural networks," in *ICANN*, vol. 4668, 2007.

[89] C. Finn, I. J. Goodfellow, and S. Levine, "Unsupervised Learning for Physical Interaction through Video Prediction," in *NeurIPS*, 2016.

[90] E. Zhan, S. Zheng, Y. Yue, L. Sha, and P. Lucey, "Generating multi-agent trajectories using programmatic weak supervision," in *ICLR*, 2019.

[91] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel Recurrent Neural Networks," in *ICML*, 2016.

[92] R. M. Neal, "Connectionist learning of belief networks," *Artif. Intell.*, vol. 56, no. 1, 1992.

[93] Y. Bengio and S. Bengio, "Modeling high-dimensional discrete data with multi-layer neural networks," in *NeurIPS*, 1999.

[94] A. van den Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, and A. Graves, "Conditional image generation with pixelcnn decoders," in *NIPS*, 2016.

[95] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu, "Video pixel networks," in *ICML*, 2017, pp. 1771–1779.

[96] K. Fragkiadaki, J. Huang, A. Alemi, S. Vijayanarasimhan, S. Ricco, and R. Sukthankar, "Motion prediction under multimodality with conditional stochastic networks," *arXiv:1705.02082*, 2017.

[97] L. Castrejon, N. Ballas, and A. Courville, "Improved conditional vrnns for video prediction," in *ICCV*, 2019.

[98] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *NIPS*, 2015.

[99] M. Henaff, J. J. Zhao, and Y. LeCun, "Prediction under uncertainty with error-encoding networks," *arXiv:1711.04994*, 2017.

[100] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[101] Y.-H. Kwon and M.-G. Park, "Predicting future frames using retrospective cycle gan," in *CVPR*, 2019.

[102] C. Vondrick and A. Torralba, "Generating the Future with Adversarial Transformers," in *CVPR*, 2017.

[103] Y. Zhou and T. L. Berg, "Learning Temporal Transformations from Time-Lapse Videos," in *ECCV*, 2016.

[104] P. Bhattacharjee and S. Das, "Temporal coherency based criteria for predicting video frames using deep multi-stage generative adversarial networks," in *NIPS*, 2017.

[105] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *ICCV*, 2017.

[106] B. Chen, W. Wang, and J. Wang, "Video imagination from a single image with transformation generation," in *ACM Multimedia*, 2017.

[107] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv:1411.1784*, 2014.

[108] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," *arXiv:1804.01523*, 2018.

[109] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016.

[110] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *ICLR*, 2017.

[111] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *ICPR*. IEEE, 2004.

[112] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Trans. on PAMI*, vol. 29, no. 12, 2007.

[113] H. Kuehne, H. Jhuang, E. Garrote, T. A. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *ICCV*, 2011.

[114] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *ICCV*, 2013.

[115] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv:1212.0402*, 2012.

[116] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *ICCV*, 2013.

[117] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *Trans. on PAMI*, vol. 36, no. 7, 2014.

[118] H. Idrees, A. R. Zamir, Y. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, "The THUMOS challenge on action recognition for videos "in the wild"," *CVIU*, vol. 155, 2017.

[119] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *CVPR*, 2009.

[120] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *IJRR*, vol. 32, no. 11, 2013.

[121] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[122] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," *arXiv: 1803.06184*, 2018.

[123] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. Li, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[124] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv:1609.08675*, 2016.

[125] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li, "YFCC100M: the new data in multimedia research," *Commun. ACM*, vol. 59, no. 2, 2016.

[126] I. Sutskever, G. E. Hinton, and G. W. Taylor, "The recurrent temporal restricted boltzmann machine," in *NIPS*, 2008.

[127] C. F. Cadieu and B. A. Olshausen, "Learning intermediate-level representations of form and motion from natural movies," *Neural Computation*, vol. 24, no. 4, 2012.

[128] R. Memisevic and G. Exarchakis, "Learning invariant features by harnessing the aperture problem," in *ICML*, vol. 28, 2013.

[129] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections," in *CoRL*, ser. Proceedings of Machine Learning Research, vol. 78, 2017.

[130] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "Robonet: Large-scale multi-robot learning," *arXiv:1910.11215*, 2019.

[131] R. Vezzani and R. Cucchiara, "Video surveillance online repository (visor): an integrated framework," *Multimedia Tools Appl.*, vol. 50, no. 2, 2010.

[132] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: parallel robust online simple tracking," in *CVPR*, 2010.

[133] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *J. Artif. Intell. Res.*, vol. 47, 2013.

[134] G. Seguin, P. Bojanowski, R. Lajugie, and I. Laptev, "Instance-level video segmentation from object tracks," in *CVPR*, 2016.

[135] Z. Bauer, F. Gomez-Donoso, E. Cruz, S. Orts-Escolano, and M. Cazorla, "UASOL, a large-scale high-resolution outdoor stereo dataset," *Scientific Data*, vol. 6, no. 1, 2019.

[136] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV*, vol. 5302, 2008.

[137] E. Santana and G. Hotz, "Learning a driving simulator," *arXiv:1608.01230*, 2016.

[138] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *CVPR*, 2004.

[139] P. Martinez-Gonzalez, S. Oprea, A. Garcia-Garcia, A. Jover-Alvarez, S. Orts-Escolano, and J. Garcia-Rodriguez, "UnrealROX: An extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation," *Virtual Reality*, 2019.

[140] D. Jayaraman and K. Grauman, "Look-ahead before you leap: End-to-end active recognition by forecasting the effect of motion," in *ECCV*, vol. 9909, 2016.

[141] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An Uncertain Future: Forecasting from Static Images Using Variational Autoencoders," in *ECCV*, 2016.

[142] Z. Hao, X. Huang, and S. J. Belongie, "Controllable video generation with sparse trajectories," in *CVPR*, 2018.

[143] Y. Ye, M. Singh, A. Gupta, and S. Tulsiani, "Compositional video prediction," in *ICCV*, October 2019.

[144] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. A. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behaviour prediction for autonomous driving applications: A review," *arXiv:1912.11676*, 2019.

[145] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, 2010.

[146] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[147] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *ECCV*, T. Pajdla and J. Matas, Eds., vol. 3024, 2004.

[148] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *ICLR*, 2018.

[149] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *NIPS*, 2017.

[150] B. Jin, Y. Hu, Q. Tang, J. Niu, Z. Shi, Y. Han, and X. Li, "Exploring spatial-temporal multi-frequency analysis for high-fidelity and temporal-consistency video prediction," *arXiv:2002.09905*, 2020.

[151] O. Shouno, "Photo-realistic video prediction on natural videos of largely changing frames," *arXiv:2003.08635*, 2020.

[152] R. Hou, H. Chang, B. Ma, and X. Chen, "Video prediction with bidirectional constraint network," in *FG*, May 2019.

[153] M. Oliu, J. Selva, and S. Escalera, "Folded recurrent neural networks for future video prediction," in *ECCV*, 2018.

[154] W. Yu, Y. Lu, S. Easterbrook, and S. Fidler, "Efficient and information-preserving future frame prediction and beyond," in *ICLR*, 2020.

[155] F. A. Reda, G. Liu, K. J. Shih, R. Kirby, J. Barker, D. Tarjan, A. Tao, and B. Catanzaro, "SDC-Net: Video prediction using spatially-displaced convolution," in *ECCV*, 2018.

[156] R. Memisevic and G. E. Hinton, "Learning to represent spatial transformations with factored higher-order boltzmann machines," *Neural Computation*, vol. 22, no. 6, 2010.

[157] R. Memisevic, "Gradient-based learning of higher-order image features," in *ICCV*, 2011.

[158] ——, "Learning to relate images," *Trans. on PAMI*, vol. 35, no. 8, 2013.

[159] V. Michalski, R. Memisevic, and K. Konda, "Modeling deep temporal dependencies with recurrent grammar cells," in *NeurIPS*, 2014.

[160] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," in *NeurIPS*, 2015.

[161] B. Klein, L. Wolf, and Y. Afek, "A dynamic convolutional layer for short rangeweather prediction," in *CVPR*, 2015.

[162] B. D. Brabandere, X. Jia, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *NeurIPS*, 2016.

[163] A. Clark, J. Donahue, and K. Simonyan, "Adversarial video generation on complex datasets," 2019.

[164] P. Luc, A. Clark, S. Dieleman, D. de Las Casas, Y. Doron, A. Cassirer, and K. Simonyan, "Transformation-based adversarial video prediction on large-scale data," *arXiv:2003.04035*, 2020.

[165] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Trans. on PAMI*, vol. 37, no. 9, 2015.

[166] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NeurIPS*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014.

[167] H. Gao, H. Xu, Q. Cai, R. Wang, F. Yu, and T. Darrell, "Disentangling propagation and generation for video prediction," in *ICCV*, 2019.

[168] Y. Wu, R. Gao, J. Park, and Q. Chen, "Future video synthesis with object motion prediction," 2020.

[169] J. Hsieh, B. Liu, D. Huang, F. Li, and J. C. Niebles, "Learning to decompose and disentangle representations for video prediction," in *NeurIPS*, 2018.

[170] S. Chiappa, S. Racanière, D. Wierstra, and S. Mohamed, "Recurrent environment simulators," in *ICLR*, 2017.

[171] K. Fragkiadaki, P. Agrawal, S. Levine, and J. Malik, "Learning visual predictive models of physics for playing billiards," in *ICLR (Poster)*, 2016.

[172] A. Dosovitskiy and V. Koltun, "Learning to Act by Predicting the Future," in *ICLR*, 2017.

[173] P. Luc, "Self-supervised learning of predictive segmentation models from video," Theses, Université Grenoble Alpes, Jun. 2019. [Online]. Available: https://tel.archives-ouvertes.fr/tel-02196890

[174] H.-k. Chiu, E. Adeli, and J. C. Niebles, "Segmenting the future," *arXiv:1904.10666*, 2019.

[175] X. Jin, H. Xiao, X. Shen, J. Yang, Z. Lin, Y. Chen, Z. Jie, J. Feng, and S. Yan, "Predicting Scene Parsing and Motion Dynamics in the Future," in *NeurIPS*, 2017.

[176] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *NIPS*, 2014.

[177] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv:1503.02531*, 2015.

[178] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," in *CVPR*, 2015.

[179] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017.

[180] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed, "Variational approaches for auto-encoding generative adversarial networks," *arXiv:1706.04987*, 2017.

[181] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *ICCV*, 2017.

[182] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee, "Deep visual analogy-making," in *NIPS*, 2015.

[183] N. Fushishita, A. Tejero-de-Pablos, Y. Mukuta, and T. Harada, "Long-term video generation of multiple futures using human poses," *arXiv:1904.07538*, 2019.

[184] J. Tang, H. Hu, Q. Zhou, H. Shan, C. Tian, and T. Q. S. Quek, "Pose guided global and local gan for appearance preserving human video prediction," in *ICIP*, Sep. 2019.

[185] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, no. null, p. 11371155, Mar. 2003.

[186] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NeurIPS*, 2014.

[187] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *CVPR*, 2015.

[188] R. Chalasani and J. C. Príncipe, "Deep predictive coding networks," in *ICLR (Workshop Poster)*, 2013.

[189] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber, "Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation," in *NeurIPS*, 2015.

[190] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *AAAI*, 2017.

[191] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fründ, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic, “The ”something something” video database for learning and evaluating visual common sense,” in *ICCV*, 2017.

[192] Z. Yi, H. R. Zhang, P. Tan, and M. Gong, “Dualgan: Unsupervised dual learning for image-to-image translation,” in *ICCV*, 2017.

[193] J. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *ICCV*, 2017.

[194] W. Luo, W. Liu, and S. Gao, “A revisit of sparse coding based anomaly detection in stacked RNN framework,” in *ICCV*. IEEE, 2017.

[195] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. S. Regazzoni, and N. Sebe, “Abnormal event detection in videos using generative adversarial nets,” in *ICIP*, 2017.

[196] L. Dinh, D. Krueger, and Y. Bengio, “NICE: non-linear independent components estimation,” in *ICLR (Workshop)*, 2015.

[197] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, “Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms,” in *NeurIPS*, 2017.

[198] “Traffic4cast: Traffic map movie forecasting,” https://www.iarai.ac.at/traffic4cast/, accessed: 2020-04-14.

[199] S. Niklaus, L. Mai, and F. Liu, “Video frame interpolation via adaptive separable convolution,” in *ICCV*. IEEE, 2017.

[200] ——, “Video frame interpolation via adaptive convolution,” in *CVPR*. IEEE, 2017.

[201] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, “A short note about kinetics-600,” *arXiv:1808.01340*, 2018.

[202] T. Xue, J. Wu, K. L. Bouman, and B. Freeman, “Visual Dynamics: Probabilistic Future Frame Synthesis via Cross Convolutional Networks,” in *NeurIPS*, 2016.

[203] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. A. Funkhouser, “Semantic scene completion from a single depth image,” in *CVPR*. IEEE, 2017.

[204] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in *CVPR*, 2015.

[205] J. Janai, F. Güney, A. Ranjan, M. J. Black, and A. Geiger, “Unsupervised learning of multi-frame optical flow with occlusions,” in *ECCV*, vol. 11220, 2018.

[206] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, “Learning what and where to draw,” in *NIPS*, 2016.

[207] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in *ECCV*, vol. 9912, 2016.

[208] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, “Recurrent Network Models for Human Dynamics,” in *ICCV*, 2015.

[209] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi, “Conditional image generation for learning the structure of visual objects,” *arXiv:1806.07823*, 2018.

[210] Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, and H. Lee, “Unsupervised discovery of object landmarks as structural representations,” in *CVPR*, 2018.

[211] R. Goroshin, M. Mathieu, and Y. LeCun, “Learning to linearize under uncertainty,” in *NeurIPS*, 2015.

[212] G. E. Hinton, A. Krizhevsky, and S. D. Wang, “Transforming auto-encoders,” in *ICANN*, vol. 6791. Springer, 2011.

[213] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun, “Unsupervised learning of spatiotemporally coherent metrics,” in *ICCV*, 2015.

[214] T. Brox and J. Malik, “Object segmentation by long term analysis of point trajectories,” in *ECCV*, vol. 6315, 2010.

[215] J. Schmidhuber, “Learning complex, extended sequences using the principle of history compression,” *Neural Computation*, vol. 4, no. 2, 1992.

[216] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine, “Learning to poke by poking: Experiential learning of intuitive physics,” in *NeurIPS*, 2016, p. 50925100.

[217] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in *ICML*, vol. 48, 2016.

[218] J. Zhang and K. Cho, “Query-efficient imitation learning for end-to-end simulated driving,” in *AAAI*, 2017.

[219] S. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. Maier-Hein, S. A. Eslami, D. J. Rezende, and O. Ronneberger, “A probabilistic u-net for segmentation of ambiguous images,” in *NeurIPS*, 2018.

[220] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, “BDD100K: A diverse driving video database with scalable annotation tooling,” *arXiv:1805.04687*, 2018.

[221] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, “The mapillary vistas dataset for semantic understanding of street scenes,” in *ICCV*, 2017.

[222] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *ICLR*, 2014.

[223] X. Yan, J. Yang, K. Sohn, and H. Lee, “Attribute2image: Conditional image generation from visual attributes,” in *ECCV*, vol. 9908, 2016.

[224] H. Wu, M. Rubinstein, E. Shih, J. V. Guttag, F. Durand, and W. T. Freeman, “Eulerian video magnification for revealing subtle changes in the world,” *ToG*, vol. 31, no. 4, 2012.

[225] R. Villegas, A. Pathak, H. Kannan, D. Erhan, Q. V. Le, and H. Lee, “High fidelity video prediction with large stochastic recurrent neural networks,” in *NeurIPS*, 2019, pp. 81–91.

[226] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, “Ladder variational autoencoders,” in *NIPS*, 2016.

[227] R. Pottorff, J. Nielsen, and D. Wingate, “Video extrapolation with an invertible linear embedding,” *arXiv:1903.00133*, 2019.

[228] D. P. Kingma and P. Dhariwal, “Glow: Generative flow with invertible 1x1 convolutions,” in *NeurIPS*, 2018.

[229] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Processing*, vol. 13, no. 4, 2004.

[230] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *CVPR*, 2018.

[231] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, “Towards accurate generative models of video: A new metric & challenges,” *arXiv:1812.01717*, 2018.

[232] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in *NIPS*, 2016.

[233] O. Breuleux, Y. Bengio, and P. Vincent, “Quickly generating representative samples from an rbm-derived process,” *Neural Computation*, vol. 23, no. 8, 2011.

[234] E. Hildreth, “Theory of edge detection,” *Proc. of Royal Society of London*, vol. 207, no. 187-217, 1980.

[235] Y. Wang, Z. Gao, M. Long, J. Wang, and P. S. Yu, “Predrnn++: Towards A resolution of the deep-in-time dilemma in spatiotemporal predictive learning,” in *ICML*, ser. Proceedings of Machine Learning Research, vol. 80, 2018.

[236] F. Cricri, X. Ni, M. Honkala, E. Aksu, and M. Gabbouj, “Video ladder networks,” *arXiv:1612.01756*, 2016.

[237] I. Prémont-Schwarz, A. Ilin, T. Hao, A. Rasmus, R. Boney, and H. Valpola, “Recurrent ladder networks,” in *NIPS*, 2017.

[238] B. Jin, Y. Hu, Y. Zeng, Q. Tang, S. Liu, and J. Ye, “Varnet: Exploring variations for unsupervised video prediction,” in *IROS*, 2018.

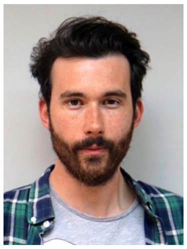
[239] J. Lee, J. Lee, S. Lee, and S. Yoon, “Mutual suppression network for video prediction using disentangled features,” *arXiv:1804.04810*, 2018.

[240] D. Weissenborn, O. Tckstrm, and J. Uszkoreit, “Scaling autoregressive video models,” in *ICLR*, 2020.

[241] L. Theis, A. van den Oord, and M. Bethge, “A note on the evaluation of generative models,” in *ICLR*, 2016.

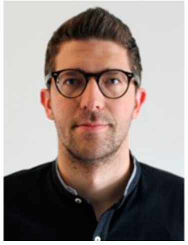
Sergiu Oprea is a PhD student at the Department of Computer Technology (DTIC), University of Alicante. He received his MSc (Automation and Robotics) and BSc (Computer Science) from the same institution in 2017 and 2015 respectively. His main research interests include video prediction with deep learning, virtual reality, 3D computer vision, and parallel computing on GPUs.

Pablo Martinez Gonzalez is a PhD student at the Department of Computer Technology (DTIC), University of Alicante. He received his MSc (Computer Graphics, Games and Virtual Reality) and BSc (Computer Science) at the Rey Juan Carlos University and University of Alicante, in 2017 and 2015, respectively. His main research interests include deep learning, virtual reality and parallel computing on GPUs.



Alberto Garcia Garcia is a Postdoctoral Researcher at the Institute of Space Sciences (ICE-CSIC, Barcelona) where he leads the efforts in code optimization, machine learning, and parallel computing on the MAGNESIA ERC Consolidator project. He received his PhD (Machine Learning and Computer Vision), MSc (Automation and Robotics) and BSc (Computer Science) from the same institution in 2019, 2016 and 2015 respectively. Previously he was an intern at NVIDIA Research/Engineering, Facebook Reality Labs, and Oculus Core Tech. His main research interests include deep learning (specially convolutional neural networks), virtual reality, 3D computer vision, and parallel computing on GPUs.

John Alejandro Castro Vargas is a PhD student at the Department of Computer Technology (DTIC), University of Alicante. He received his MSc (Automation and Robotics) and BSc (Computer Science) from the same institution in 2017 and 2016 respectively. His main research interests include human behavior recognition with deep learning, virtual reality and parallel computing on GPUs.



Sergio Orts-Escolano received a BSc, MSc and PhD in Computer Science from the University of Alicante in 2008, 2010 and 2014 respectively. His research interests include computer vision, assistive robotics, 3D sensors, GPU computing, virtual/augmented reality and deep learning. He has authored +50 publications in top journals and conferences like CVPR, SIGGRAPH, 3DV, BMVC, CVIU, IROS, UIST, RAS, etcetera. He is also a member of European Networks like HiPEAC and Eucog. He has experience as a professor in academia and industry, working as a research scientist for companies such as Google and Microsoft Research.

Jose Garcia-Rodriguez received his Ph.D. degree, with specialization in Computer Vision and Neural Networks, from the University of Alicante (Spain). He is currently Full Professor at the Department of Computer Technology of the University of Alicante. His research areas of interest include: computer vision, computational intelligence, machine learning, pattern recognition, robotics, man-machine interfaces, ambient intelligence, computational chemistry, and parallel and multicore architectures.

Antonis Argyros is a professor of computer science at the Computer Science Department, University of Crete and a researcher at the Institute of Computer Science, FORTH, in Heraklion, Crete, Greece. His research interests fall in the areas of computer vision and pattern recognition, with emphasis on the analysis of humans in images and videos, human pose analysis, recognition of human activities and gestures, 3D computer vision, as well as image motion and tracking. He is also interested in applications of computer vision in the fields of robotics and smart environments.