Towards an Understanding of Our World by GANing Videos in the Wild

Bernhard Kratzwald, Zhiwu Huang, Danda Pani Paudel, Luc Van Gool
Computer Vision Lab, ETH Zürich, Switzerland
bkratzwald@ethz.ch, {zhiwu.huang, paudel, vangool}@vision.ee.ethz.ch

Abstract

Existing generative video models work well only for videos with a static background. For dynamic scenes, applications of these models demand an extra pre-processing step of background stabilization. In fact, the task of background stabilization may very often prove impossible for videos in the wild. To the best of our knowledge, we present the first video generation framework that works in the wild, without making any assumption on the videos' content. This allows us to avoid the background stabilization step, completely. The proposed method also outperforms the state-of-the-art methods even when the static background assumption is valid. This is achieved by designing a robust one-stream video generation architecture by exploiting Wasserstein GAN frameworks for better convergence. Since the proposed architecture is one-stream, which does not formally distinguish between fore- and background, it can generate and learn from videos with dynamic backgrounds. The superiority of our model is demonstrated by successfully applying it to three challenging problems: video colorization, video inpainting, and future prediction.

1. Introduction

Viewed as a digital window into the real-life physics of our world, videos capture how objects behave, move, occlude, deform, and interact with each other. Furthermore, Videos record how camera movements, scene depth or changing illumination influence a scene. Fully understanding their temporal and spatial dependencies is one of the core problems in computer vision. Teaching computers to model and interpret scene dynamics and dependencies occurring within videos is an essential step towards intelligent machines able to interact with their environment.

Compared to the domain of images, the work on supervised and unsupervised learning from videos is still in its infancy. This can be attributed to the high-dimensional nature of videos. The requirement of large and expensive amounts of labeled training data is the main limiting factor for supervised learning methods. The recent focus of research on videos; therefore, has shifted from supervised to unsupervised models. The mere endless amount of unlabeled video data available on the Internet further underpins the choice of unsupervised methods.

State-of-the-art unsupervised video models are often designed with problem-specific restrictions in mind. Generative models require stable backgrounds [55] or are tailored to work on simple datasets such as facial expression videos [54]. Such models fail to work in the wild since they are restricted to videos fulfilling those requirements. The motivation of this work is to recreate a robust, universal and unrestricted generative framework that does not impose any preconditioning on the input videos.

The task of generating videos is inherently related to modeling and understanding the scene dynamics within them. For a realistic video generation, it is essential to learn which objects move, how they move, and how they interact with each other – which *vice versa* implies an understanding of real-world semantics. A model capable of understanding these semantics is ideally not restricted to the task of video generation but can also transfer this knowledge to a broad number of other applications. Important applications include action classification, object detection, segmentation, future prediction, colorization, and inpainting.

Our paper focuses both on the robustness of our generative video framework as well as on its application to three problems. First, we design a stable architecture with no prior constraints on the training data. More precisely, we design a one-stream generation framework that does not formally distinguish between fore- and background, allowing us to handle videos with moving backgrounds/cameras. Video generation in a single-stream is a fragile task, demanding a carefully selected architecture within a stable optimization framework. We accomplish such stability by exploiting state-of-the-art Wasserstein GAN frameworks in the context of video generation. In a second step, we demonstrate the applicability of our model by proposing a general multifunctional framework dedicated to specific applications. Our extension augments the generation model with an auxiliary encoder network and an application specific loss function. With these modifications, we success-

fully conduct several experiments for unsupervised end-to-end training. The two main contributions of this paper are as follows: (i) We propose iVGAN – a robust one-stream video generation framework working in the wild. Our experiments show that iVGAN outperforms state-of-the-art generation frameworks on both raw and stabilized videos. (ii) We demonstrate the utility of the multifunctional framework of iVGAN for three challenging problems: video colorization, video inpainting, and future prediction. To the best of our knowledge, this is the first work exploiting the advantages of GANs in the domain of video inpainting or video colorization.

We provide the readers with the source code for all our models: `https://github.com/bernhard2202/improved-video-gan`

2. Related Work

Generative Adversarial Networks (GANs): GANs [17] have proven successful in the field of unsupervised learning. Generally, GANs consist of two neural networks: a generator network trained to generate samples and a discriminator network trained to distinguish between real samples drawn from the data distribution and fake samples produced by the generator. Both networks are trained in an adversarial fashion to improve each other. However, GANs are also known to be potentially unstable during training. To address this problem, *Radford et al*. [48] introduced a class of *Deep Convolutional GANs* (DCGANs) that imposes empirical constraints on the network architecture. *Salimans et al*. [51] provide a set of tools to avoid instability and mode collapsing. *Che et al*. [8] use regularization methods for the objective to avoid the problem of missing modes. *Arjovsky et al*. [2] suggest to minimize the Wasserstein-1 or Earth-Mover distance between generator and data distribution with theoretical reasoning. In a follow-up paper, *Gulrajani et al*. [19] propose an improved method for training the discriminator – termed *critic* by [2] – which behaves stably, even with very deep ResNet architectures. GANs have been mostly investigated on images, showing significant success with tasks such as image generation [11, 19, 27, 48], image super-resolution [38], style transfer [31, 65], and many others.

Video Generation: There are only few attempts like [55, 54] for video generation through GANs. In particular, *Vondrick et al*. [55] adapted the DCGAN model to generate videos, predict future frames and classify human actions. Their *Video GAN* (VGAN) model suggests the usage of independent streams for generating fore- and background. The background is generated as an image and then replicated over time. A jointly trained mask selects between foreground and background to generate videos. In order to encourage the network to use the background stream, a sparsity prior is added to the mask during learning. More recently, *Tulyakov et al*. [54] also adopted a two-stream generative model that produces dynamic motion vs. static content. In particular, the static part is modeled by a fixed Gaussian when generating individual frames within the same video clip, while the motion part is modeled by a recurrent network that represents the dynamic patterns. Although successful in their context of static backgrounds, many videos would defy these approaches. This is because videos in the wild often contain changing backgrounds, and it is non-trivial to stabilize backgrounds even when only shaking. This paper, therefore, proposes a robust video generation model that is capable to work directly on real-world videos.

Video Colorization: Works on image and video colorization can be divided into two categories: interactive colorization which requires some kind of user input [10, 26, 29, 39, 43, 60] and automatic methods [7, 20, 24, 40, 57, 62]. Our approach belongs to the latter category. Most automatic methods come with restrictions preventing them from working in the wild. For instance, [40] requires colored pictures of a similar viewing angle and [7] requires separate parameter tuning for every input picture. Methods such as [24, 57] produce undesirable artifacts. In the video domain, methods such as [20] process each frame independently, which in turn leads to temporal inconsistencies. Recently, image colorization has been combined with GANs [16, 36], but no prior research on colorizing videos has been presented.

Video Inpainting: Inpainting is a fairly well-investigated problem in the image domain [5, 35, 59]. For videos, it has been used to restore damages in vintage films [52], to remove objects [18] or to restore error concealment [12]. State-of-the-art frameworks like [37] use complex algorithms involving optical flow computation; thus demanding an optimized version to run within a feasible amount of time. Recovering big areas of an image or a video, also called *hole-filling*, is inherently a more difficult problem than the classical inpainting. Approaches like texture synthesis [4, 13] or scene completion [21] do not work for *hole-filling* [47]. While there has been some work on image inpainting with adversarial loss functions [47], we are not aware of any in the case of videos.

Future Prediction: Future prediction is the task of predicting the future frames for one/multiple given input frames. In contrast to video generation, future prediction is an elegant way of turning an unsupervised modeling problem into a supervised learning task by splitting videos into conditioning input, as well as ground-truth future. Our method builds upon recent future prediction work *e.g*. [6, 14, 15, 32, 44], especially that using generative models and adversarial losses [45, 49, 55, 56].

3. Our Model - iVGAN

For robust video generation, we propose a video generation model, called *improved Video GAN* (iVGAN), consisting of a generator and a discriminator in the GAN framework. Particularly, the designed generator network $G : Z \rightarrow \mathcal{X}$ produces a video $\boldsymbol{x}$ from a low dimensional latent code $\boldsymbol{z}$. The proposed critic network $C : \mathcal{X} \rightarrow \mathbb{R}$ is optimized in order to provide the generator updates with good gradient information, by distinguishing between real and fake samples. Distinct from existing video generation models, we design the generation framework without any prior assumptions upon the nature of the data. It is thus essential that our generator is one-stream, without separating back- and foreground. As studied in [2, 19, 48, 51] for image generation, it is non-trivial to train GAN models in a stable manner. Especially for video generation, it turns out to be much more challenging as low frequencies also span the additional temporal domain. To address this problem, we generalize the state-of-the-art Wasserstein GAN to the context of video generation for more stable convergence. Formally, we place our network within the Wasserstein GAN framework [2] optimizing

$$\min_G \max_{\|C\|_L \leq 1} V(G,C) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})}[C(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[C(G(\boldsymbol{z}))]. \tag{1}$$

In order to enforce the Lipschitz constraint on the critic function, we penalize its gradient-norm with respect to the input [19]. For this purpose we evaluate the critic's gradient $\nabla_{\hat{\boldsymbol{x}}} C(\hat{\boldsymbol{x}})$ with respect to points sampled from a distribution over the input space $\hat{\boldsymbol{x}} \sim p_{\hat{x}}$, and penalize its squared distance from one via

$$\mathcal{L}_{GP}(C) = \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim p_{\hat{x}}} \left[\left(\|\nabla_{\hat{\boldsymbol{x}}} C(\hat{\boldsymbol{x}})\|_2 - 1 \right)^2 \right]. \tag{2}$$

The distribution $p_{\hat{x}}$ is defined by uniformly sampling on straight lines between points in the data distribution and points in the generator distribution. Hence, the final unconstrained objective is given by

$$\min_G \max_C V(G,C) + \lambda \mathcal{L}_{GP}(C), \tag{3}$$

where the hyperparameter λ is used to balance the GAN objective with the gradient penalty.

3.1. Generator Network

The generator takes a latent code sampled from a 100-dimensional normal distribution $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ and produces an RGB video containing 32 frames of 64×64 pixels. We use a linear up-sampling layer in the first step, producing a tensor of size $2 \times 4 \times 4 \times 512$. The linear block is followed by four convolutional blocks of spatio-temporal [30] and fractionally-strided [61] convolutions. This combination has proven to be an efficient way to upsample, while preserving spatial and temporal invariances [55, 56]. All convolutional layers utilize $4 \times 4 \times 4$ kernels, a stride of $2 \times 2 \times 2$, and add a bias to the output. Inspired by the ResNet architecture [23], we initialize the convolutional kernels as in [22], which we found led to more stable gradients than a default initialization.

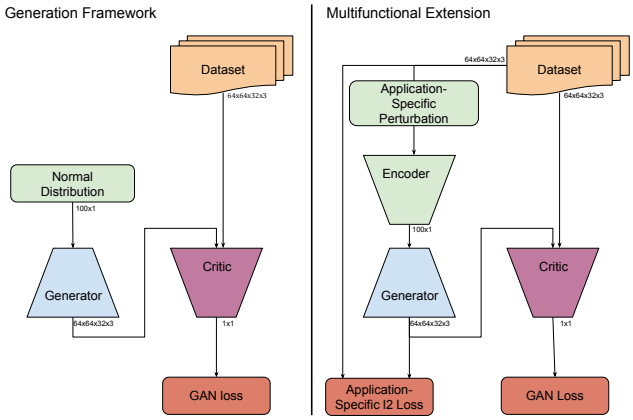

Figure 1: iVGAN video generation framework and its multifunctional extension.

Similar to DCGAN [48], all but the last layers are followed by a batch normalization layer [28]. Batch normalization stabilizes the optimization by normalizing the inputs of a layer to zero mean and unit variance, which proved critical for deep generators in early training to prevent them from collapsing [48].

The first four blocks are followed by a ReLU nonlinearity after normalization layer, while the last layer uses a hyperbolic tangent function. This is beneficial to normalize the generated videos, identically to the videos in our dataset, within the range $[-1, 1]$.

3.2. Critic Network

The critic network maps an input video to a real-valued output. It is trained to distinguish between real and generated videos, while being constrained (Eqn. 2) to yield effective gradient information for generator updates.

The critic consists of five convolutional layers and is followed by an additional linear down-sampling layer. As in [55], we use spatio-temporal convolutions with $4 \times 4 \times 4$ kernels. The kernel weights are initialized following [22]. For more expressiveness, we add a trainable bias to the output. All convolutions include a stride of $2 \times 2 \times 2$ to enable efficient down-sampling of the high-dimensional inputs.

Batch normalization correlates samples within a mini-batch by making the output for a given input $\boldsymbol{x}$ dependent on the other inputs $\boldsymbol{x}'$ within the same batch. A critic with

Figure 2: Video generation results on stabilized golf clips. Left: the two-stream VGAN model. Right: our one-stream iVGAN model.

batch normalization, therefore, maps a batch of inputs to a batch of outputs. On the other hand, in Equation 2, we are penalizing the norm of the critic's gradient with respect to each input independently. For this reason, batch normalization is no longer valid in our theoretical setting. To resolve this issue, we use layer normalization [3] following [19]. Layer normalization works equivalent to batch normalization, but mean and standard deviation is calculated independently for every single sample $\boldsymbol{x_i}$ over the hidden layers. We found that layer normalization is not necessary for convergence, but essential if we optimize the generator with additional objectives, as described in the multifunctional extension in Section 4.

All but the last layer use a leaky ReLU [58] activation. We omit using a soft-max layer, or any kind of activation in the final layer, since the critic is not trained to classify between real and fake samples, but rather trained to yield a good gradient information for generator updates.

3.3. Learning and Parameter Configuration

We optimize both networks using alternating stochastic gradient descent, more precisely we optimize the critic five times for every update step on the generator. The hyperparameter λ, controlling the trade-off between the GAN objective and the gradient penalty (Eqn. 3), is set to 10 as reported in [19]. We use Adam [34] with initial hyperparameters $\alpha = 0.0002$, $\beta_1 = 0.5$, $\beta_2 = 0.99$ and a batch size of 64 which have proven to work best for us. We divide the learning rate by two, after visual convergence. We train our network from scratch which usually takes four to six days on a *GeForce GTX TITAN X* GPU. The entire network is implemented in TensorFlow [1].

4. Multifunctional Extension

With a simple, yet powerful modification, we extend our generation architecture to a multifunctional video processing framework. We choose three challenging applications to demonstrate the semantics our framework is capable of learning. To successfully colorize grey-scale input videos our network must learn temporal consistent color semantics. Meadows *e.g*. have to be painted in a shade of green which should stay consistent over time. Inpainting, which is completing and repairing missing or damaged parts of a video, requires the network to learn spatial consistencies such as symmetries, as well as temporal consistencies. As a third application, we choose future prediction conditioned on a single input frame which has shown to be a challenging problem [55], it requires learning which objects are plausible to move and how they do so.

Fig. 1 compares the generation framework architecture with its multifunction extension. Similar to conditional GANs [46], the generator is no longer dependent on a randomly drawn latent code $\boldsymbol{z}$ but conditioned on additional application-specific information $\boldsymbol{y}$. For instance, in video colorization $\boldsymbol{y}$ is a gray-scale video, in inpainting we condition on the damaged input clip; and, for future prediction, on a single input frame. A convolutional network $E : \mathcal{Y} \rightarrow \mathcal{Z}$ encodes $\boldsymbol{y}$ to a latent code $\boldsymbol{z}$ which is in turn used to generate the resulting video. To guide this generation we extend the framework by an additional application-specific $\ell 2$ loss. For colorization, we calculate the loss between the generated video and the conditioned grey-scale clip. For inpainting, on the other hand, we use the generated video and the ground-truth, for future prediction the first frame of the generated video and the image we conditioned the generation on.

We jointly optimize for the GAN value function (Eqn. 1), the gradient penalty (Eqn. 2), and the new domain-specific $\ell 2$ loss – using two hyperparameters λ and ν to control the trade-off between them. To gain a deeper understanding of the interaction between GAN- and reconstruction loss, we conduct experiments with two variations of the colorization framework: In the *unsupervised* setting the reconstruction loss is calculated in grey-scale color space and does therefore not penalize wrong colorization, leaving the GAN-loss solely responsible for learning color semantics. In the *supervised* setting, on the other hand, the $\ell 2$-loss is calculated in RGB color space and thus penalizes both wrong colorization and wrong structure. It remains unclear what role the GAN-loss takes in the latter setting. Following Zhao *et al*. [63] we argue in Section 5.4 that the GAN-loss acts as a regularizer preventing the encoder-generator from learning plain identity functions.

4.1. Learning and Parameter Configuration

The encoder network consists of four strided convolutional layers, each of which is followed by a batch normalization layer and a ReLu activation function. We found it difficult to adjust the hyperparameter ν which controls the

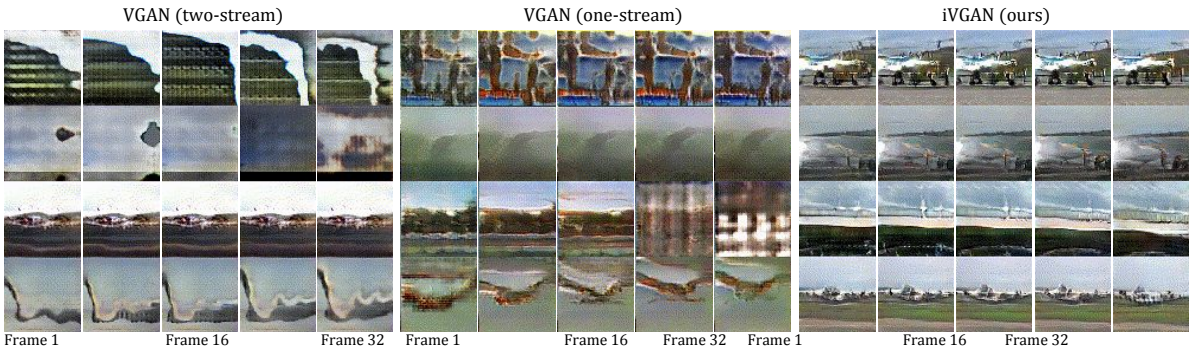

Figure 3: Comparing videos generated using the one and two stream VGAN model [55], with our iVGAN framework on unstabilized airplane videos.

trade-off between the GAN loss and the domain-specific $\ell 2$ loss. The critic output, which determines the gradient for the encoder-generator updates, is not bound, while the $\ell 2$ loss is per definition in the range $[0, 1]$. We found that the GAN loss behaves more stable when using layer normalization in the critic, allowing us to monitor the losses and empirically set $\nu = 1000$.

5. Experiments

We evaluate our generation framework on multiple datasets and compare our results with the VGAN model [55]. VGAN is – besides ours – the only large-scale generative video model we are aware of. Models such as [49] require one or more input frames, other models like [54] have only been evaluated on closed domain datasets, e.g. facial expression videos. To evaluate our multifunctional extension, we choose to colorize gray-scale videos, inpaint damaged videos, and predict the future from static images.

5.1. Datasets

We used different datasets of unlabeled but filtered video clips, which have been extracted from high-resolution videos at a natural frame rate of 25 frames per second.

Stabilized Videos: This dataset[1] was composed by [55] and contains parts of the Yahoo Flickr Creative Commons Dataset [53]. The Places2 pre-trained model [64] has been used to filter the videos by scene category *golf course*. All videos have been preprocessed to ensure a static background. Therefore, SIFT keypoints [41] were extracted to estimate a homography between frames and minimize the background motion [55]. The task of background stabilization may very often not be valid, forcing us to renounce a significant fraction of data. Discarding scenes with non-static background significantly restricts our goal of learning real-world semantics through unsupervised video understanding in the wild.

In the Wild Dataset: We compiled a second dataset of filtered, unlabeled and unprocessed video clips. Similar to the golf dataset, videos are filtered by scene category, in this case *airplanes*. We, therefore, collected videos from the YouTube-BoundingBoxes dataset [50] which have been classified containing airplanes. No pre-processing of any kind has been applied to the data and the dataset thus contains static scenes as well as scenes with moving background or moving cameras.

5.2. Generation of Stabilized Videos

Qualitative Results: Fig. 2 qualitatively compares results of the VGAN model with our iVGAN generator trained on the golf dataset.There is no formal concept of fore- or background in the iVGAN model since the entire clip is generated in a single stream. Our model nonetheless learns from the data to generate clips with a static background and moving foreground. Despite the fact that the background is not generated as an image anymore, it looks both sharp and realistic in the majority of samples. The foreground suffers from the same flaws as the VGAN model: it is blurrier than the background, people and other foreground objects turn into BLOBs. The network correctly learns which objects should move, and generates fairly plausible motions. Some samples are close to reality, but a fraction of samples collapses during training. We observe that this fraction is smaller than when using the VGAN generator. Overall the network learns correct semantics and produces scenes with a sharp and realistic looking background but blurry and only fairly realistic foreground-motion.

Quantitative Results: We used Amazon Mechanical Turk for a quantitative evaluation. We asked workers to rate how realistic videos look on a linear scale from 5 (very real-

[1] We downloaded the dataset from `http://carlvondrick.com/tinyvideo/`

istic) to 1 (very unrealistic). We generated random samples from each model and used random clips from the dataset as a reference value. The mean score for each model was calculated from more than 7000 ratings. We paid workers one cent per evaluation and required them to historically have a 95% approval rating on Amazon MTurk. The evaluation results are shown in table 1. Videos from the golf dataset have an average rating of 4.10. The frame stabilization method, used to enforce static backgrounds, sometimes distorts the clips. We witness that such videos get low ratings, hence the golf dataset does not get an optimal score of 5. Our model clearly outperforms the baseline VGAN-generator with a margin of 0.5. Workers assess our videos on average as more realistic. We attribute this to the lower number of collapsed samples and the high stability of our generation framework.

	Average Rating of Videos
Real Videos	*4.10*
VGAN [55]	2.51
iVGAN (ours)	**3.01**

Table 1: Quantitative evaluation of video generation frameworks. Average user rating of their realism from 1 (very unrealistic) to 5 (very realistic).

5.3. Generation in the Wild

We conducted four independent experiments using the VGAN generator on the airplanes dataset, varying the learning rate between 0.00005 and 0.0002, and the sparsity penalty on the foreground mask between 0.1 and 0.15. In all runs, without exception, the generator collapsed and failed to produce any meaningful results. One might argue that it is unfair to evaluate a two-stream generation model, which assumes a static background, on a dataset violating this assumption. Therefore, we repeated a series of experiments using the one-streamed VGAN model, which does not separate fore- and background. A one-stream model should theoretically be powerful enough to converge on this dataset. Regardless of that, the one-stream version of VGAN collapsed as well in all experiments and failed to generate meaningful videos.

Fig. 3 qualitatively compares generations from the two- and one-stream VGAN model with our iVGAN generator. Although the quality of our samples is lower compared to the stabilized golf videos, the generator did in no single experiment collapse. The iVGAN model – unlike any other generative model – produces both: videos with static background, as well as videos with moving background or camera motion. A fraction of the generated videos collapsed to meaningless colored noise, nonetheless. Yet, the network learns important semantics since a significant number of videos shows blurry but realistic scenes, objects, and motions.

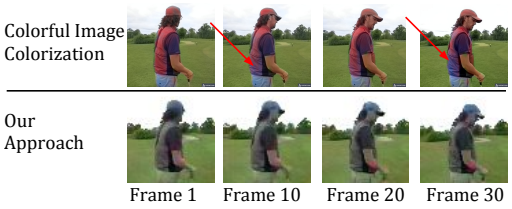

Figure 4: Color consistency over time with different colorization approaches. Red arrows mark spots where color is inconsistent over time.

Since the VGAN model collapsed and was not able to produce meaningful results in any experiment we abstain from a quantitative evaluation – it is trivial to see which results are more realistic.

5.4. Colorization

Fig. 4 qualitatively compares our framework with the state-of-the-art *Colorful Image Colorization* (CIC) model [62]. The CIC model colorizes videos in their original resolution frame by frame. Our model, on the other hand, colorizes the entire clip at once but is restricted to in- and outputs of 64×64 pixels. Frame-wise colorization is known to suffer from temporal inconsistencies [20]. Fig. 4 illustrates *e.g.* how the CIC colorized jacket changes its color over time while our colorization stays consistent. We saw that our network overall learns correct color semantics: areas in the input are selected, "classified" and then painted according to their nature. The sky *e.g.* is colorized in shades of blue or gray-white and trees are painted in a darker green than the grass. We, therefore, argue that the network not only selects the trees, but also recognizes (classifies) them as such, and paints them according to their class. The quality of the segmentation depends on the sharpness of the edges in the gray-scale input. Colorized videos are blurrier compared to the gray-scale input. This is mainly due to the fact that we dont keep the spatial resolution of the videos but encode them to a latent code, from which the colorized videos are then generated. Furthermore, using the mean squared error function to guide reconstructions is known to generate blurry results [45].

Furthermore, we investigate the interplay between the GAN-loss and encoder-generator reconstruction loss. We therefore compare two variations of our model where the reconstruction loss is calculated in RGB color space (supervised) and in grayscale color space (unsupervised), as described in section 4. Our experiments indicate that the supervised colorization network, having a stronger objective, tends to overfit. Although they perform equally well

Model	Average Rating	PSNR hold-out data	PSNR ood data
Video Colorization			
supervised	2.45	25.2 dB	23.4 dB
unsupervised	2.95	25.6 dB	24.2 dB
Video Inpainting			
salt & pepper	3.63	29.2 dB	25.4 dB
boxes (fixed)	3.37	25.3 dB	22.9 dB
boxes (random)	3.43	24.7 dB	22.7 dB

Table 2: Quantitative evaluation of video colorization and inpainting frameworks. Left: Average user rating of their realism from 1 (very unrealistic) to 5 (very realistic). Right: Peak signal to noise ratio between generated videos, and gray-scale input (colorization) or ground-truth videos (inpainting).

on the training data, the unsupervised network outperforms the supervised network on hold-out and out-of-domain data as quantitatively shown in table 2. We evaluated the sharpness of the colorizations, measured by the *Peak Signal to Noise Ratio* (PSNR) in grey-space, as well as the colorization quality judged by human ratings collected via Amazon MTurk. In both cases, the unsupervised model outperforms its supervised counterpart. The unsupervised model relies strongly on the GAN loss, which we argue – following Zhao *et al*. [63] – acts as a regularizer preventing the encoder-generator network from learning identity functions.

5.5. Inpainting

We corrupt inputs in various ways and observe the reconstruction quality of our network: 25% salt and pepper noise, 20×20 pixel holes in the center of the clip, and 20×20 pixel holes at random positions. We trained our network on stabilized golf videos, and evaluate it on the unstabilized airplane dataset as shown in Fig. 5.

Denoising salt and pepper corruptions is a well-studied problem, going back many years [9]. State-of-the-art approaches operate on noise levels as high as 70% [42]. The denoised reconstructions generated by our model are sharp and accurate. We can use our model – which has been trained on stabilized videos – to denoise clips with moving cameras or backgrounds, which would not be possible with a two-stream architecture. The reconstructed output is slightly blurrier than the ground-truth, which we attribute to the fact that we generate the entire video from a latent encoding and do not keep the undamaged parts of the input.

The task of hole-filling is more challenging since the reconstructions have to be consistent in both space and time. While we don't claim to compete with the state-of-the-art, we use it to illustrate that our network learns advanced spatial and temporal dependencies. For instance, in the second clip and second column of Fig. 5 we can see that, although the airplane's pitch elevator is mostly covered in the input, it is reconstructed almost perfectly and not split into two halves. This usually works best when the object covered is visible on more than one side of the box. We sometimes observe that such objects disappear although we could infer their existence from symmetry (*e.g*. one airplane wing is covered and not reconstructed). Our model learns temporal dependencies as objects which are covered in some – but not all frames – are reconstructed consistently over time. The overall quality does not suffer significantly when randomizing the locations of the boxes.

Our quantitative evaluations results are shown in table 2. We asked Amazon MTurkers to rate how realistic reconstructions look. Consistently with our quantitative findings, users rate the salt & pepper reconstructions with a score of 3.63 very high (real videos score 4.10). The margin between boxes at fixed and random positions is very small and not significant. Furthermore, we calculate the peak signal to noise ratio between ground-truth videos and their reconstructed counterparts. Salt and pepper reconstructions achieve again the best score. The margin between boxes at fixed and boxes at random positions is too small to rank the models. All three models perform better on hold-out data than on the out-of-domain data.

5.6. Future Prediction

We qualitatively show results of our future prediction network in Fig. 6. Future frames are blurrier, compared to the inpainting and colorization results, which we attribute to the fact that the reconstruction loss only guides the first frame of the generated clip – not the entire clip.

Although in many cases the network fails to generate a realistic future, it often learns which objects should move and generates fairly plausible motions. Since we use only one frame guiding the generation and omit to use the ground-truth future, these semantics are solely learned by the adversarial loss function. We emphasize that, to the best of our knowledge, this work and [55] are the only two approaches using a single input frame to generate multiple future frames. We suffer from the same problems as [55], such as hallucinating or omitting objects. The horse in the bottom-most clip in Fig. 6 *e.g*. is dropped in future frames. Unsupervised future prediction from a single frame is a notoriously hard task. Nonetheless, our network learns which objects are likely to move, and to generate fairly plausible motions.

6. Conclusion and Outlook

This paper proposed a robust video generation model that generalizes the state-of-the-art Wasserstein GAN technique to videos, by designing a new one-stream generative

Figure 5: Comparison of ground-truth videos with the reconstructions of salt&pepper noise, missing holes in the center and at random positions.

Figure 6: Future prediction results: Generated videos and the input frames the generations were conditioned on.

model. Our extensive qualitative and quantitative evaluation justified that our stable one-stream architecture outperforms the VideoGAN model on stabilized data. On the other hand, we demonstrated that two-stream architectures are too restricted by their design to work on raw video data. Besides, we also verified that one-stream video generation is inherently difficult and fails to work if the framework is not carefully designed. In contrast, our proposed iVGAN model is capable of learning real-world semantics while both the one-stream and two-stream VGAN models collapse when the training data is not stabilized. The proposed iVGAN model does not need to distinguish between fore- and background and is, therefore, the only one able to generate videos with moving camera/background as well as those with a static background. Although our architecture does not explicitly model the fact that our world is stationary, it correctly learns which objects are plausible to move and how.

Additionally, dropping the assumption of a static background frees our model to handle other than background-stabilized training data, thus significantly broadening its applicability. We emphasized the superiority of our model by demonstrating that our proposed multifunctional extension is applicable to several applications, each of them requiring our network to learn different semantics. The experiments on our video colorization framework indicate that the model is able to select individual parts of a scene, recognize them, and paint them according to their nature. The inpainting experiments show that our model is able to learn and recover important temporal and spatial dependencies by filling the damaged holes consistently, both in space and time. We trained our models on stabilized input frames in both applications and successfully applied them to unprocessed videos. A two-stream model would by design not be able to colorize or inpaint clips with moving background or camera.

Although unsupervised understanding of videos is still in its infancy, we have presented a more general and robust video generation model that can be used as a multifunctional framework. This said, we believe that the quality of the generated videos can be further improved by using deeper architectures like ResNet [23] or DenseNet [25], or by employing the recent, progressive growing technique of GANs [33].

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané,

R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24–1, 2009.

[5] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[6] Y.-W. Chao, J. Yang, B. Price, S. Cohen, and J. Deng. Forecasting Human Dynamics from Static Images. apr 2017.

[7] G. Charpiat, M. Hofmann, and B. Schölkopf. Automatic image colorization via multimodal predictions. *Computer Vision–ECCV 2008*, pages 126–139, 2008.

[8] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. *CoRR*, abs/1612.02136, 2016.

[9] T. Chen, K.-K. Ma, and L.-H. Chen. Tri-state median filter for image denoising. *IEEE Transactions on Image processing*, 8(12):1834–1838, 1999.

[10] A. Y.-S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin. Semantic colorization with internet images. In *ACM Transactions on Graphics (TOG)*, volume 30, page 156. ACM, 2011.

[11] E. L. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. *CoRR*, abs/1506.05751, 2015.

[12] M. Ebdelli, O. Le Meur, and C. Guillemot. Video inpainting with short-term windows: application to object removal and error concealment. *IEEE Transactions on Image Processing*, 24(10):3034–3047, 2015.

[13] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1033–1038. IEEE, 1999.

[14] C. Finn, I. J. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. *CoRR*, abs/1605.07157, 2016.

[15] K. Fragkiadaki, S. Levine, and J. Malik. Recurrent network models for kinematic tracking. *CoRR*, abs/1508.00271, 2015.

[16] Q. Fu, W.-T. Hsu, and M.-H. Yang. Colorization using convnet and gan, 2017.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[18] M. Granados, K. I. Kim, J. Tompkin, J. Kautz, and C. Theobalt. Background inpainting for videos with dynamic objects and a free-moving camera. In *European Conference on Computer Vision*, pages 682–695. Springer, 2012.

[19] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017.

[20] R. K. Gupta, A. Y.-S. Chia, D. Rajan, and H. Zhiyong. A learning-based approach for automatic image and video colorization. *arXiv preprint arXiv:1704.04610*, 2017.

[21] J. Hays and A. A. Efros. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 4. ACM, 2007.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001.

[25] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[26] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 351–354. ACM, 2005.

[27] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic. Generating images with recurrent adversarial networks. *CoRR*, abs/1602.05110, 2016.

[28] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[29] R. Ironi, D. Cohen-Or, and D. Lischinski. Colorization by example. In *Rendering Techniques*, pages 201–210, 2005.

[30] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[31] F. Jurie. A new log-polar mapping for space variant imaging.: Application to face detection and tracking. *Pattern Recognition*, 32(5):865–875, 1999.

[32] N. Kalchbrenner, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. *CoRR*, abs/1610.00527, 2016.

[33] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[34] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] N. Komodakis. Image completion using global optimization. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 442–452. IEEE, 2006.

[36] S. Koo. Automatic colorization with deep convolutional generative adversarial networks, 2016.

[37] T. T. Le, A. Almansa, Y. Gousseau, and S. Masnou. MOTION-CONSISTENT VIDEO INPAINTING. In *ICIP 2017: IEEE International Conference on Image Processing*, ICIP 2017: IEEE International Conference on Image Processing, Beijing, China, Sept. 2017.

[38] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016.

[39] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM transactions on graphics (tog)*, volume 23, pages 689–694. ACM, 2004.

[40] X. Liu, L. Wan, Y. Qu, T.-T. Wong, S. Lin, C.-S. Leung, and P.-A. Heng. Intrinsic colorization. *ACM Transactions on Graphics (TOG)*, 27(5):152, 2008.

[41] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[42] C.-T. Lu and T.-C. Chou. Denoising of salt-and-pepper noise corrupted image using modified directional-weighted-median filter. *Pattern Recognition Letters*, 33(10):1287–1295, 2012.

[43] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 309–320. Eurographics Association, 2007.

[44] Z. Luo, B. Peng, D.-A. Huang, A. Alahi, and L. Fei-Fei. Unsupervised Learning of Long-Term Motion Dynamics for Videos. jan 2017.

[45] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015.

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

[47] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. *CoRR*, abs/1604.07379, 2016.

[48] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[49] 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 preprint arXiv:1412.6604*, 2014.

[50] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. *arXiv preprint arXiv:1702.00824*, 2017.

[51] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[52] N. C. Tang, C.-T. Hsu, C.-W. Su, T. K. Shih, and H.-Y. M. Liao. Video inpainting on digitized vintage films via maintaining spatiotemporal continuity. *IEEE Transactions on Multimedia*, 13(4):602–614, 2011.

[53] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[54] S. Tulyakov, M. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation. *CoRR*, abs/1707.04993, 2017.

[55] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[56] C. Vondrick and A. Torralba. Generating the future with adversarial transformers. CVPR, 2017.

[57] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 277–280. ACM, 2002.

[58] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[59] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. *arXiv preprint arXiv:1611.09969*, 2016.

[60] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *IEEE Transactions on Image Processing*, 15(5):1120–1129, 2006.

[61] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010.

[62] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. *CoRR*, abs/1603.08511, 2016.

[63] J. J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *CoRR*, abs/1609.03126, 2016.

[64] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

[65] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.