Constrained Manifold Learning for Videos

Iti Chaturvedi
Information Tehchnology
James Cook University, Townsville
iti.chaturvedi@jcu.edu.au

Jin Xiang
School of Computer Science and Engineering,
Nanyang Technological University, Singapore

***Abstract*—Automatic image manipulation can be used to make subtle changes at the pixel level resulting in morphism from one domain to another. This is desirable in tasks such as creating mock expressions for an individual or dynamic scene generation in autonomous driving. This type of morphism can be achieved using an adversarial model where the generator and the discriminator compete to produce fake images of the target domain. Due to high variance among the images, it is difficult to learn an optimal loss function. Previously, manifold matching of clusters in the source domain with labeled samples and the target domain that is generated was used to overcome this limitation. To generate videos it is common to use three-dimensional convolution however, such a model has very high complexity. Instead, in this paper we use manifold constrained model selection to do a constrained clustering of the combined manifold with fixed start and end images for the morphism. We show that each step in the principal path connecting the centroids is analogous to a single time delay in the video sequence. Hence, we can construct a cascade of models using samples from a pair of connected centroids such that one model is used to initialize the next. We apply the model to smile generation from neutral face expression and for predicting the next few frames while driving on real roads. We are able to outperform the baselines in the quality of images generated and the computational cost for training the model.**

***Index Terms*—Adversarial Networks, Bayesian Model Selection, Manifold Learning, Video Generation**

I. Introduction

Automatic image manipulation can be used to generate and annotate images for several tasks such as different facial expressions of the same person [1], [2]. It may also be used for visually pleasing animations where the landscape or objects are changing. In such a model as we vary the source image x then the corresponding generated images y will also change a lot [3]. Here, the source domain is labeled image samples that are easily available. The target domain are samples we want to generate, as they are not easily available [4]. For example, we may want to generate a smiling face given a single neutral face image.

Such manipulation can be done at pixel level by training an image generator and an image discriminator adversarially in a min-max game. After several iterations of gradient descent on each pixel independently the input image devoid of the sought features will be adjusted enough to result in surreal images giving a dreaming effect. For example, an existing image can be altered so that it is 'more cat like', or we can make an animal or other patterns appear in a cloud [5].

In particular, the generative adversarial network (GAN) is a framework for estimating a generative model via an adversarial process [6]. GANs are popular networks in natural language processing research [7], [8] but also in multimodal analysis [9], [10], especially in the area of image-to-image translation. This task is defined as the transformation of a certain representation of a scene into another representation of the same scene. GAN have become popular due to their ability to generate surprisingly realistic images. Other applications include semantic segmentation of satellite and cityscape images [11].

A GAN is made up of a generator and a discriminator [12]. As shown in Figure 1, a generator aims to generate the smiling face from the input neutral face and random noise. The discriminator on the other hand will classify a generated smile image as real or fake. However, training the discriminator suffers from two limitations: firstly, the gradients often vanish during adversarial training and secondly the gradients may have large variances across samples [13], [14]. In this paper, we overcome these limitations by constraining the learning along a high density manifold. Previously, regularized k-means has shown good results in image morphism from one shape to another [15].

Figure 1 shows a sample video for smile generation. Here, we can select a pair of face images labelled as neutral and smiling. Next, the entire set of images in the training videos are clustered such that the k centroids lie on the principal path connecting the given pair of images [16]. Face images that lie far from the principal path can be discarded as noise. We train the first GAN only using images that lie close to the first centroid. Similarly, the second GAN is initialized with weights of the first GAN and then trained using images that lie close to the second centroid in the path. In this way, a cascade of GANs and the adaptive error is a weighted sum of the errors of all the models [17]. We refer to the resulting model as Constrained Adaptive Manifold Error Learning (CAMEL).

The organization of the paper is as follows: Section II reviews related works and datasets on image translation; Section III provides the preliminary concepts necessary to understand the present work; Section IV details the proposed model for generating videos; in Section V we validate our method on two real world datasets and finally we provide conclusions in Section VI.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

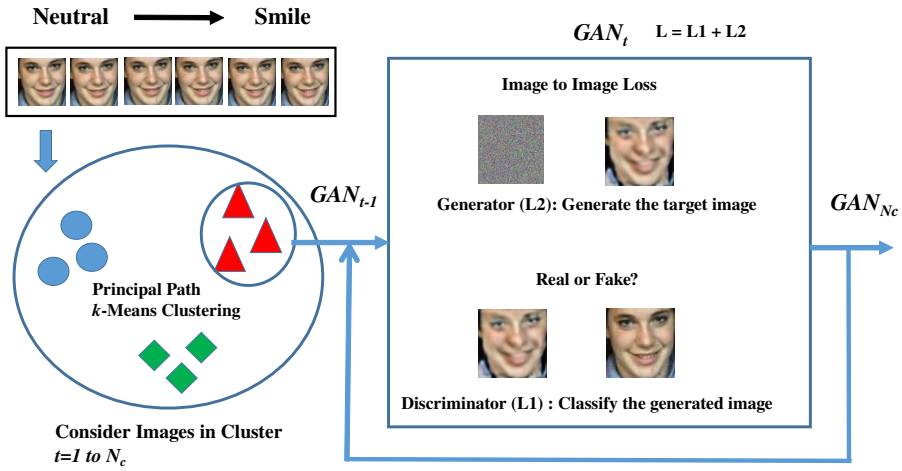

Fig. 1. Flowchart for the principal path GAN. For each cluster along the principal path $t = 1$ to N_c a virtual GAN_t is trained.

II. Related Work and Contributions

Video prediction aims to learn a nonlinear transformation function between given frames to predict subsequent frames. Learning to generate future frames in a video sequence has wide application in reinforcement learning based games and robotics. Video representation is commonly done using sub-sampling on a fixed number of input frames or pooling over frames [18]. Most previous authors describe video generation as a two-step process: first is motion generation and second is content generation. For example, ImaGINator uses spatio-temporal fusion to generate expressions from a single facial image and emotion label [19]. However, complex cityscapes cannot be modeled using a single image.

Recent work decompose a video into a static background, a mask and moving objects prior to training a GAN [20]. The traditional GAN suffers from high complexity and vanishing gradients during training. To overcome this, Wasserstein or Earth Movers GAN was proposed where the loss function is defined as the cost of transporting pixels from source to target distribution This is achieved by constraining the weights to be in a range that results in instability and slow convergence [21].

GAN's also need paired images in the source and target domain during training. CycleGAN overcome this problem by mimicking the cycle consistency in machine translation where a phrase translate from English to French should translate from French back to English. Hence, in this model the image output from one generator is used as input to the second and the output from the second generator is matched with the original input [22].

Another author divided the task of generating videos into content subspace and motion subspace [23]. Their model is able to generate videos with the same content but different motion as well as videos with different content and same motion. The work in this paper is closest to the approach described in [24] where video frames are drawn from a prior that is a function of the past few frames. In this paper, we are also inspired by the work done in [25] where spherical clusters in the manifold of the source and target domain are matched for training.

In [26], the authors used unlabeled videos to generated tiny videos from any static starting images. Such a model has very high complexity due to the additional temporal parameters. However, in this paper we want to generate videos for a particular labeled action such as a smile or driving on a specific road. We can use a smaller number of samples along the high density manifold that is obtained by clustering the data. Next, we consider a cascade of models along the principal path connecting the clusters that is able to capture the morphism and hence the temporal dynamics. For example, the starting point is a neutral face and the end point will be a smiling face. In [27], the authors leverage on the fact that temporally adjacent samples also correspond to neighbors in the latent space. They consider spatial pooling to model linear transformations. Instead, we model the temporal dependence between the source and target domain via a principal path in the latent space. Due to regularization, this will ensure smooth transformation without the need for spatial pooling [17]. We can summarize the main contributions of the paper as follows:

- Previous authors clustered source and target manifold and then matched them before training the GAN. However, we consider the principal path through the manifold from source to target domain that connects the centroids of different clusters.
- To capture temporal dependence previous authors used spatial pooling in three-dimensional convolutional networks. Instead, we consider an adaptive model where each step in the path is analogous to a single time delay in the video sequence. Hence, we can construct a cascade of models for each subsequent time delay where one model is used to initialize the next.

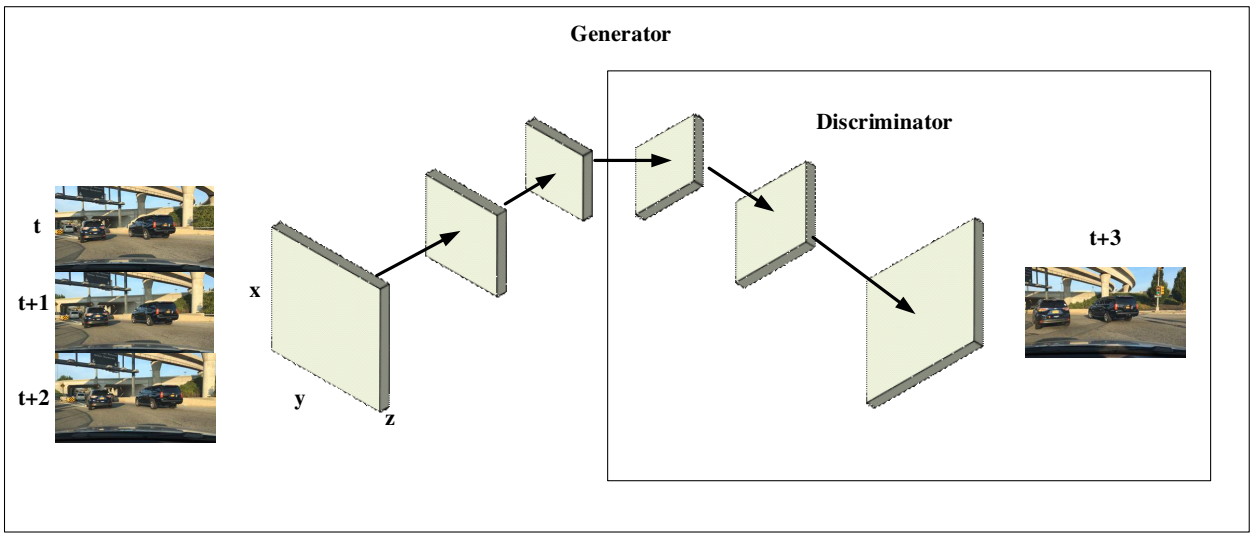

Fig. 2. Video GAN where three consecutive input frames are combined to predict the next output frame. The Generator acts as an encoder and the Discriminator acts as a decoder. Each convolutional layer has three dimensional features of the size $\{x, y, z\}$

- The principal path uses Bayesian model selection to ensure smooth morphism from source to target domain. This will eliminate the need for tuning the parameters during gradient descent.

Facial expressions such as a smile have a large variation due to age, gender, and personality. Existing methods are unable to cope with the rapidly changing intensities of a smile in a face video [28]. It is necessary to have a memory model that can adapt to the increasing or decreasing intensities of a smile. Here, we select two high intensity images from different emotions such as 'Neutral' and 'Smile' as the start and end points. Next, we cluster the data into N_c clusters such that the principal connecting path between the two points is minimal. When each centroid in k-means clustering is restricted to be a real frame the resulting clustering is known as k-medoids clustering [29]. Hence, each medoid or centroid is a frame sample in the cluster whose average dissimilarity to all the other frames in the cluster is minimal. Choosing optimal initial medoids in clustering is a challenging problem. Here, we only select two medoids at a time and then cluster the data such that the principal connecting path between the two points is minimal.

It is worth mentioning that the principal path through the manifold is in fact a principal component during dimensionality reduction. The choice of regularization parameter using Bayesian model selection ensures that the solution is consistent on different runs. The extent to which the path passes through the data in the manifold also depends on the regularization parameter. Lastly, such a minimum energy path achieves smooth morphism from one image to another by proposing moves of closing and filling gaps along the transition path.

III. Video Adversarial Nets

In the adversarial training framework, a generative model is pitted against an adversary that is a discriminative model that learns to determine whether a sample is from the model distribution or the data distribution. In the context of image data, the generative model can be thought of as analogous to a team of counterfeiters, trying to produce fake currency and use it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency. Competition in this game drives both teams to improve their methods until the counterfeits are indistinguishable from the genuine articles. Using this framework, we can train both models using only the highly successful backpropagation and dropout algorithms and sample from the generative model using only forward propagation. In this paper we employ a video GAN model where the discriminative model D takes a sequence of frames. Only the last frames are either real or generated by the generator G, the rest of the sequence is always from the dataset. This allows the discriminative model to make use of temporal information, so that G learns to produce sequences that are temporally coherent with the input [30]. Figure 2 illustrates a video GAN where three input frames are combined to predict single output frame. Each convolutional layer has three dimensional features of the size $\{x, y, z\}$. The generator is made up of an encoder and a decoder. The discriminator architecture is identical to the decoder component of the generator. Following previous authors, we showed both in the same diagram. As shown in Figure 1 there are two loss functions L1 (generator) and L2 (discriminator) that are used for training.

In addition, in order to conserve memory we divide each input frame into patches. Then, the discriminator D tries to estimate the probability that a patch comes from the dataset instead of being produced by a generative model G. The two models are simultaneously trained so that G learns to generate patches that are hard to classify by D, while D learns to discriminate patches generated by G.

The training of the pair (G, D) consists of two alternating steps, described below :

A. Training D :

Let $x_i = (x_i^1, x_i^2, \ldots, x_i^n)$ be the i^{th} sequence of input frames and $y_i = (y_i^1, y_i^2, \ldots, y_i^m)$ be a sequence of output frames. We train D to classify the input (x_i, y_i) into class 1 and the input $(x_i, G(x_i))$ into class 0. Hence, we perform gradient descent on D while keeping G fixed. The binary cross entropy loss function we use to train D is:

$$\mathcal{L}1(x_i, y_i) = L(D(x_i, y_i), 1) + L(D(x_i, G(x_i)), 0) \quad (1)$$
$$L(a, b) = -\sum_i b \log(a) + (1-b)\log(1-a)$$

where $a, b \in \{0, 1\}$.

B. Training G:

Next, we take a different sample pair (x_i, y_i) from the above update. While keeping the weights of D fixed, we can perform one gradient descent update to minimize the adversarial loss :

$$\mathcal{L}2(x_i, y_i) = L(D(x_i, G(x_i)), 1) \quad (2)$$

where L is the binary cross-entropy loss defined previously. Minimizing this loss means that the generative model G is making the discriminative model D as 'confused' as possible, in the sense that D will not discriminate the prediction correctly. However, this can make the model unstable hence in practice we minimize the combined loss $\mathcal{L}1 + \mathcal{L}2$. Figure 1 shows that a GAN will compute two different loss functions and try to minimize the total loss.

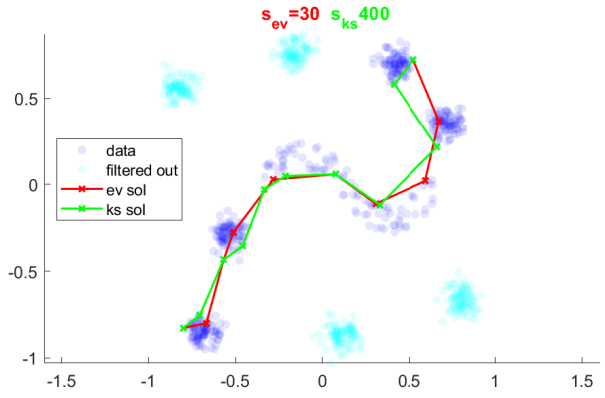

Fig. 3. The path predicted by Bayesian model selection (ev) is shown in red and the path predicted by cross-validation (ks) is shown in green. The data in light blue is far from the principal path and hence is discarded.

IV. Constrained Adaptive Error Learning

In this section, we describe the Bayesian model selection to determine the regularization parameters. Next, we introduce the backstepping algorithm for different time delays using an adaptive model. Lastly, we describe the complete framework of the proposed CAMEL and the reduction in computational cost.

A. Bayesian model selection for Principal-path

The image dataset was transformed to a low dimensional space prior to clustering. The number of dimensions was determined by Bayesian model selection. It will predict the optimal values of the parameters so that the posterior probability of the principal path clustering is maximum. Details can be found in [15]. The k-means clustering aims to partition N observations into $k = N_c$ clusters in which each observation belongs to the cluster with the nearest mean. This results in a partitioning of the data space. We can give the minimization objective of k-means as follows:

$$\min_{\hat{\mathrm{x}},\mu} \frac{\gamma}{2} \sum_{i=1}^{N} \sum_{j=1}^{N_C} ||x_i - \hat{x}_j||^2 \quad (3)$$

where γ is a scaling factor, $\mu_i \in \{1, 2, \ldots, N_c\}$ is the label associated with image x_i, the sample image at centroid of cluster j is $\hat{x}_j$. Figure III-B illustrates the well-separated clusters in a synthetic dataset manifold. The data in light blue belongs to clusters far from the principal path and hence is discarded. The images are also projected to a low dimensional space during Bayesian model selection.

Next, we can consider a principal path that connects two points in a manifold and passes through the center of mass of the data. Such a path can be viewed as a non-linear dimensionality reduction of the data. The extent to which one can pass through the data is dependent on the regularization parameter. We can use Bayesian model selection to determine the regularization parameter.

The primal minimization problem in order to learn a smooth transition path connecting a starting point w_0 to an end point w_{N_C+1} is :

$$\min_{\hat{\mathrm{x}},\mu} \frac{\gamma}{2} \sum_{i=1}^{N} \sum_{j=1}^{N_C} ||x_i - \hat{x}_j||^2 + \frac{\eta}{2} \sum_{j=0}^{N_C} ||\hat{x}_{j+1} - \hat{x}_j||^2 \quad (4)$$

where η and γ regulate the trade-off between data-fitting and smoothness of the inferred path and w_0 and w_{N_C+1} represents the starting and ending points in the inferred path. Eq 4 is a straightforward extension of k-means clustering where the first and last clusters are kept fixed, the other clusters are evolved and those are topologically connected via a series of springs in sequence.

Here we assume a Gaussian prior during model selection where the variance of each cluster j with center w_j is γ and the variance of the straight line connecting the start and end points w_0 and w_{N_C+1} is η. We can select the optimal γ and η using Bayesian model selection as follows:

$$\hat{\mathrm{x}} = \arg\max_{\hat{\mathrm{x}}} p(\hat{\mathrm{x}}|\mathrm{x}, \gamma, \eta) \quad (5)$$

Fig. 4. We show the centroids corresponding to different steps in the principal path through the manifold for a random start and end image for NEMO dataset. Next, we show the centroids in the principal path through the manifold for a pair of start and end image in DAVE2.

where $\{\hat{\mathrm{x}}\}_{N_c \times d}$ is a matrix of centroids. Figure III-B illustrates the principal paths through a manifold for two random start and end points. The path predicted by Bayesian model selection (ev) is shown in red and the path predicted by cross-validation (ks) is shown in green. The ratio of $s = \frac{\eta}{\gamma}$ is significantly higher by Bayesian model selection resulting in a smoother path. The desired path lies on a local manifold that does not involve the whole data set. For centroids that are not a part of this local manifold, we can filter out all the associated data samples in that cluster. Such a path will be consistent even when varying the number of clusters or steps in the path.

B. Backstepping temporal images

In order to model temporal sequences of images in a video we are motivated by the backstepping control design where a virtual model is created for each time delay. Then, the adaptive error of the combined model is the sum of errors of each virtual model as follows:

$$e(t) = e(t) + \lambda_1 e(t-1) + \ldots + \lambda_t e(1) \tag{6}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_t$ are the constants and $e(t)$ is the error of the model at the t^{th} time stamp. The constants allow us to capture the temporal information in video data. Hence, the error can adapt to new image samples in the video sequence depending on the previous samples seen at the previous time step.

In this paper, we consider the principal path through the manifold for a pair of start and end images from the input and the target domain to create virtual models for each time stamp. In particular, each step in the principal path through the manifold is used to construct the virtual model that captures the morphism from t to $t+1$. In each step t of the algorithm, gradient descent tries to minimize the error function E_t :

$$E_t = \frac{1}{2} e(t)^2 \ , \ \theta_t(n+1) = \theta_t(n) + \delta\theta(n) \tag{7}$$
$$\delta\theta_t(n) = \hat{\gamma}\Big[\frac{\partial E_t(n)}{\partial \theta_t(n)}\Big]$$

where n is a single iteration in the training of the t^{th} model, $\hat{\gamma}$ is the learning rate. Since, the starting point is from the input domain and the end point of the path is from the target domain. Hence, the clustering will help gradient descent, as the distance between two connecting centroids is minimal. Furthermore, the regularization term will ensure smooth morphism from input to target domain.

C. Principal-path GAN Framework

In this section, we detail the complete framework for principal-path GAN model. The first stage is to extract a sub-set of image sequences such that the distance from a centroid image is below a threshold. Hence, we select a pair of images from the two different domains as the starting and ending points of the principal-path clustering. For example, we can take a 'neutral' image as the starting point and a 'smile' image as the end point for 'smile generation' task. For the autonomous driving task, we randomly select two different road images. The clustering will result in a set of N_c centroids including the pair of images.

In order to construct a GAN at each centroid on the principal path, we extract the sub-set of training sequences that are closest to two connecting centroids in the path namely t and $t+1$ (See Figure 1). Once we train the GAN at the centroid t, we can use it to initialize the weights of the GAN at centroid $t+1$ in the path until we reach the end point. To test a new sample we can simply use the GAN at G_{N_c}. Figure 4 illustrates the principal path morphism from between a pair of random 'Neutral' and 'Smiling' faces. Each centroid in the path corresponds to a single step. The first two images are similar and the last two images are similar in each sequence.

Similarly, we can illustrate the morphism between different road scenes. This process may be repeated resulting in several principal paths in the manifold.

D. Computational Complexity

During k-means clustering we only need to compute the distance between each image and the centroid hence the complexity is $N_c \times N_p \times T$ where T is number of iterations for clustering. To train the CAMEL we only need the samples close to centroids in the principal path. Let us assume that the maximum number of centroids is N_c and we repeat the training for N_p pairs of images from source and target domain. If the dataset has N video sequences with up-to T frames, then after clustering and thresholding we only select $N^{0.25}$ sequences with an average of $T/2$ frames. Hence, the training complexity of each GAN is reduced to $N^{0.25} \times T/2 \times N_c \times N_p$. This is exponentially smaller than the original complexity of $N \times T$ for each iteration since $N_c < N_p << << N$.

V. Experiments

Validation of the proposed CAMEL (available on GitHub[1]) is done on two real world dataset : (1) NEMO Smile Videos (2) DAVE2 Driving Videos. Following previous authors, we use the PSNR and SSIM index to compare the generated images with the ground truth for test samples.

A. NEMO Smile Dataset

The model is trained on several videos of the individuals where the expression gradually changes from neutral to smiling face. For testing the model we only provide a neutral face and the model is able to generate the video of the action. Hence, the end frames will belong to a specific class and is dependent on the starting frame. We would like to clarify that generation of neutral expression from a smiling face would require training a different model.

The UvA-NEMO dataset [31] contains 1240 videos, 643 corresponding to posed smiles and 597 to spontaneous ones. The dataset comprises 400 subjects (215 male and 185 female) with different ages ranging from 8 to 76 (50 subjects wear glasses). The videos are sampled at 50 FPS and frames have a resolution of 1920×1080 pixels, with an average duration of 3.9 s. The beginning and the end of each video corresponds to a neutral expression. The intensity of a smile increases slowly until it is maximum and then decreases back to 1. Following previous authors, we consider 32 frames to capture the complete transition from neutral to smile expression.

We trained the model on 800 video sequences and tested on the remaining videos. The past six input frames were used to predict the next frame for smile generation. The first two rows in Figure 5 illustrate a sample test sequence generated by the baseline GAN and by the proposed CAMEL for the same number of training epochs. Here we trained the baseline GAN for 1500 epochs and for each of the three steps in the principal path we trained the CAMEL for 500 epochs. We repeat the training for four different principal paths. The prediction with GAN is very blurry compared with CAMEL. For CAMEL the intensity of smile increases with each time step for a male and female sample. We also compared with another baseline MOCOGAN described in [23]. The model is able to capture the smile motion; however the faces are distorted even after extensive training. Furthermore, the identity of the person cannot be specified in this model.

[1]http://github.com/SenticNet/constrained-manifold-learning-for-videos

B. DAVE2 Driving Dataset

We collected the majority of the road data in New Jersey, including two-lane roads with and without lane markings, residential streets with parked cars, tunnels and even unpaved pathways [32]. More data was collected in clear, cloudy, foggy, snowy and rainy weather, both day and night. The model is trained with time-stamped video from a front-facing camera in the car synced with the steering wheel angle applied by the human driver. The vehicle drove along paved and unpaved roads with and without lane markings and handled a wide range of weather conditions. As more training data was gathered, performance continually improve.

We consider 30000 video sequences of 50 frames each for training and another 1000 video sequences for testing. The past four input frames were used to predict the next frame for smile generation. Figure 5 illustrate a sample test sequence generated by the baseline GAN and by the proposed CAMEL for the same number of training epochs. Here we trained the baseline GAN for 15000 epochs and for each of the three steps in the principal path we trained the CAMEL for 5000 epochs. We repeat the training for four different principal paths. The prediction with GAN is very blurry compared with CAMEL. The second driving video sequence shows a car turning left as the pillar on the left side becomes visible. The third video is a car driving forward at night. The white boards on the right side get closer with each frame. The quality of images generated by MOCOGAN on driving dataset was very poor hence we did not report them.

C. Parameter Settings

We consider an identical generator and discriminator network with three convolutional up sampling and three convolutional down sampling layers. The input images are reduced to a dimension of 64×64 for the smile dataset and 320×180 for the driving dataset. Each convolutional layer has 512 kernels and each kernel is three dimensional $3 \times 3 \times 3$. The principal path algorithm was set to have a maximum of 10 clusters; however the optimal number is determined by the model. The regularization parameter s was determined by Bayesian model selection as described in Section IV-A.

D. Evaluation Metrics

To evaluate the proposed model we consider the structural similarity index (SSIM) and the peak signal to noise ratio (PSNR). The SSIM measures the difference in the visible structures in an image such as degradation due to noise, blur or flare. Similarly, PSNR is the mean square error over all the squared value differences divided by image size and by

Fig. 5. We compare the video sequence generated with simple GAN, MOCOGAN and the proposed CAMEL. The prediction with GAN is very blurry compared with CAMEL. The first frame is the input.

three. A higher PSNR or SSIM with the ground truth indicates higher quality. In Figure 6 we also show the mean and standard deviation of PSNR and SSIM [33] for both datasets. We can see that the PSNR and SSIM will decrease with each generated time frame. For NEMO dataset the proposed CAMEL has significantly higher PSNR and SSIM compared to the simple GAN model. For the case of DAVE2 driving dataset, the SSIM is slightly higher than the baseline GAN model however the PSNR is not significantly different. The standard deviations of both models remain the same with number of time frames. For NEMO data, the variance for SSIM increases significantly when predicting 8 or more consecutive frames. Due to the complexity of the image samples. It is difficult for the human eye to visualize the motion in Figure 5. However, in Figure 6 we can see that the error with the original video is reducing from t=2 to t=6.

VI. Conclusion

In this paper, we propose a novel approach to generate videos from a static starting image. We conclude that by considering samples close to a constrained path in the manifold we can show smooth morphism of images during the video. An adaptive error model where the error is accumulated over time is found to be more suitable to rapidly changing scene dynamics such as in driving. We are able to outperform the baseline significantly in the quality of generated video on two real world tasks. Lastly, the computational cost of the model is exponentially smaller compared to traditional GAN learning.

VII. Acknowledgement

This work is partially supported by the Data Science and Artificial Intelligence Center (DSAIR) at the Nanyang Technological University. This work is also partially supported by the College of Science and Engineering at James Cook University.

References

[1] S. Nam, Y. Kim, and S. J. Kim, "Text-adaptive generative adversarial networks: Manipulating images with natural language," in *NIPS*, 2018.
[2] 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.
[3] A. Chan, Y. Tay, and Y. S. Ong, "What it thinks is important is important: Robustness transfers through input gradients," 2020.
[4] I. Chaturvedi, Y. S. Ong, and R. V. Arumugam, "Deep transfer learning for classification of time-delayed gaussian networks," *Signal Processing*, vol. 110, pp. 250 – 262, 2015.
[5] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," *CVPR*, pp. 5188–5196, 2014.
[6] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017, pp. 5967–5976.
[7] Y. Li, Q. Pan, S. Wang, T. Yang, and E. Cambria, "A generative model for category text generation," *Information Sciences*, vol. 450, pp. 301–315, 2018.
[8] Y. Li, Q. Pan, S. Wang, H. Peng, T. Yang, and E. Cambria, "Disentangled variational auto-encoder for semi-supervised learning," *Information Sciences*, vol. 482, pp. 73–85, 2019.
[9] I. Chaturvedi, R. Satapathy, S. Cavallari, and E. Cambria, "Fuzzy commonsense reasoning for multimodal sentiment analysis," *Pattern Recognition Letters*, vol. 125, no. 264-270, 2019.
[10] E. Cambria and A. Hussain, "Sentic album: Content-, concept-, and context-based online personal photo management system," *Cognitive Computation*, vol. 4, no. 4, pp. 477–496, 2012.
[11] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, "St-gan: Spatial transformer generative adversarial networks for image compositing," in *CVPR*, 2018.

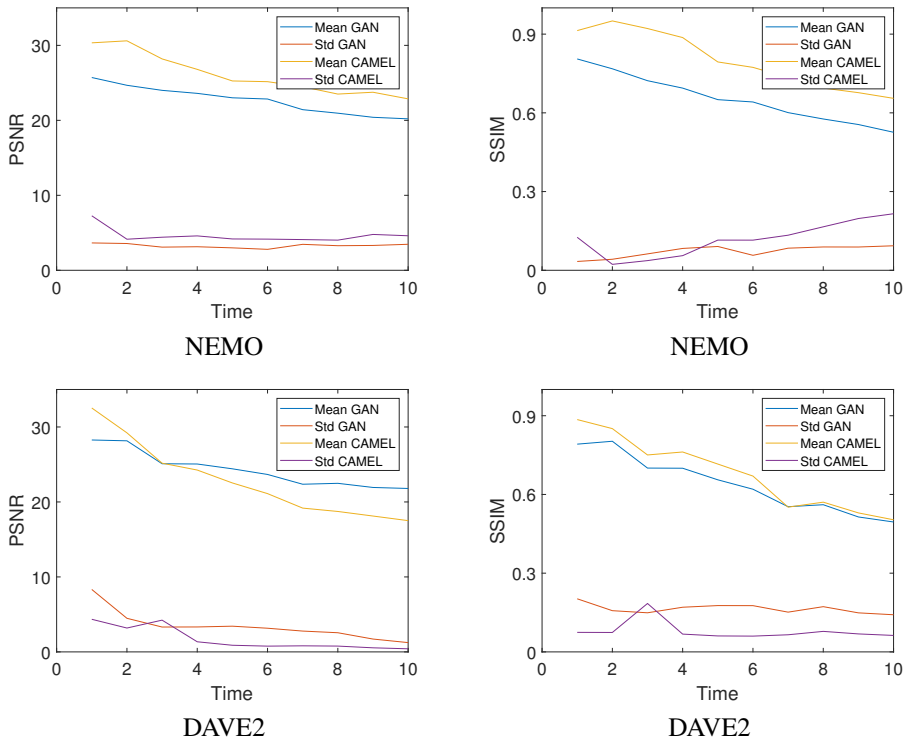

Fig. 6. Error between predicted frame and ground truth on NEMO and DAVE2 dataset

[12] Y. Luo, X. Cai, Y. Zhang, J. Xu, and X. Yuan, "Multivariate time series imputation with generative adversarial networks," in *NIPS*, 2018.

[13] X. Wang and R. Zhang, "Kdgan: Knowledge distillation with generative adversarial networks," in *NIPS*, 2018.

[14] I. Chaturvedi, E. Cambria, and D. Vilares, "Lyapunov filtering of objectivity for spanish sentiment model," in *IJCNN*, 2016, pp. 4474–4481.

[15] M. J. Ferrarotti, W. Rocchia, and S. Decherchi, "Finding principal paths in data space," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2018.

[16] H. Liu, J. Cai, Y. Wang, and Y. S. Ong, "Generalized robust bayesian committee machine for large-scale gaussian process regression," in *ICML*, vol. 80, 2018, pp. 3131–3140.

[17] B. Da, A. Gupta, and Y. S. Ong, "Curbing negative influences online for seamless transfer evolutionary optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 12, pp. 4365–4378, 2019.

[18] W. Hao, Z. Zhang, and H. Guan, "Integrating both visual and audio cues for enhanced video caption," in *AAAI*, 2018, pp. 6894–6901.

[19] Y. Wang, P. Bilinski, F. Brémond, and A. Dantcheva, "Imaginator: Conditional spatio-temporal gan for video generation," in *WACV*, 2019.

[20] Y. Li, M. R. Min, D. Shen, D. E. Carlson, and L. Carin, "Video generation from text," in *AAAI*, 2018, pp. 7065–7072.

[21] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," in *NIPS*, 2017, pp. 5769–5779.

[22] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *ICCV*, pp. 2242–2251, 2017.

[23] S. Tulyakov, M. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *CVPR*, 2018, pp. 1526–1535.

[24] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," in *ICML*, 2018, pp. 1182–1191.

[25] N. Park, A. Anand, J. R. A. Moniz, K. Lee, T. Chakraborty, J. Choo, H. Park, and Y. Kim, "Mmgan: Manifold matching generative adversarial network," in *ICPR*, 2017.

[26] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 613–621.

[27] R. Goroshin, M. F. Mathieu, and Y. LeCun, "Learning to linearize under uncertainty," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 1234–1242.

[28] C. F. Benitez-Quiroz, R. Srinivasan, and A. M. Martnez, "Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild," in *IEEE CVPR*, 2016, pp. 5562–5570.

[29] E. Cambria, T. Mazzocco, A. Hussain, and C. Eckl, "Sentic medoids: Organizing affective common sense knowledge in a multi-dimensional vector space," ser. Lecture Notes in Computer Science. Berlin Heidelberg: Springer-Verlag, 2011, vol. 6677, pp. 601–610.

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

[31] A. A. Salah and T. Gevers, "Are you really smiling at me? spontaneous versus posed enjoyment smiles," in *In ECCV*, 2012.

[32] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016.

[33] and A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.