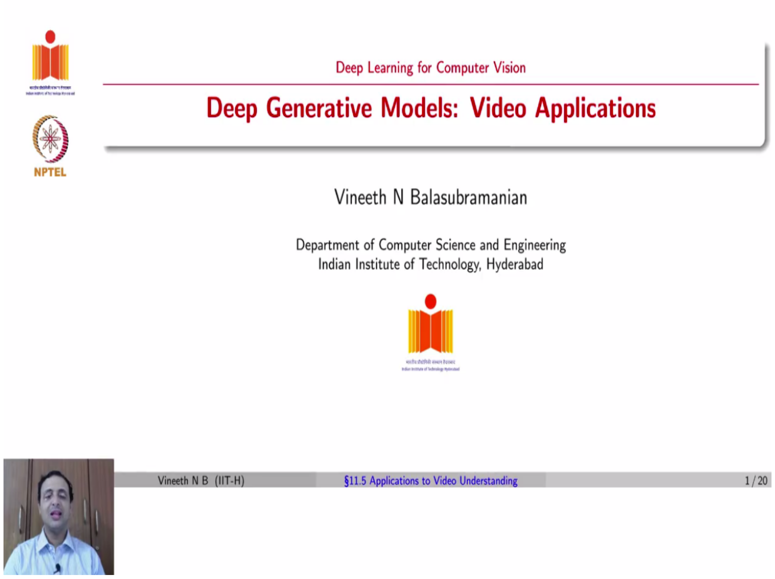
Deep Learning for Computer Vision
Professor Vineeth N Balasubramanian
Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad
Lecture 72
Deep Generative Models: Video Applications

(Refer Slide Time: 00:15)



Deep Learning for Computer Vision

Deep Generative Models: Video Applications

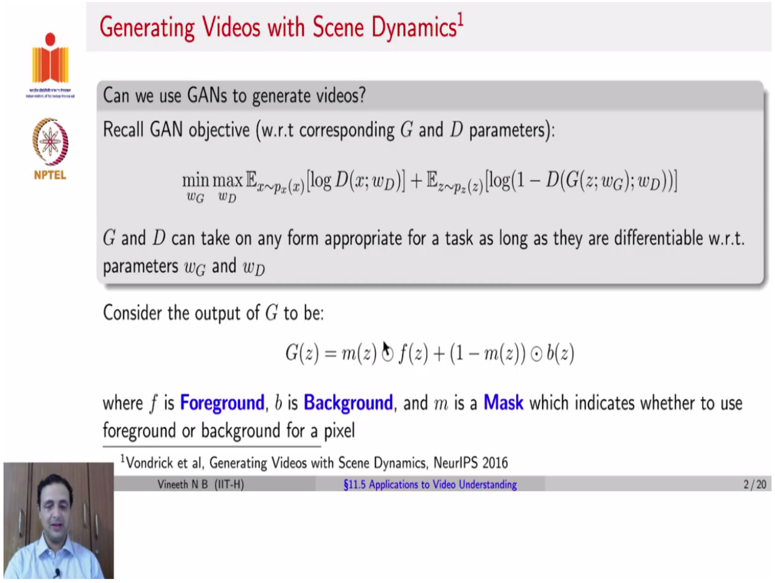
Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 1 / 20

For the last lecture of this week, we will talk about Applications of Generative Models in video understanding. In particular, we will talk about three different methods which have done different things in the scope of video understanding and generative modeling.

(Refer Slide Time: 00:38)



Generating Videos with Scene Dynamics[1]

Can we use GANs to generate videos?

Recall GAN objective (w.r.t corresponding G and D parameters):

$$\min_{w_G}\max_{w_D} \mathbb{E}_{x\sim p_x(x)}[\log D(x;w_D)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z;w_G);w_D))]$$

G and D can take on any form appropriate for a task as long as they are differentiable w.r.t. parameters w_G and w_D

Consider the output of G to be:

$$G(z) = m(z) \odot f(z) + (1 - m(z)) \odot b(z)$$

where f is **Foreground**, b is **Background**, and m is a **Mask** which indicates whether to use foreground or background for a pixel

[1] Vondrick et al, Generating Videos with Scene Dynamics, NeurIPS 2016

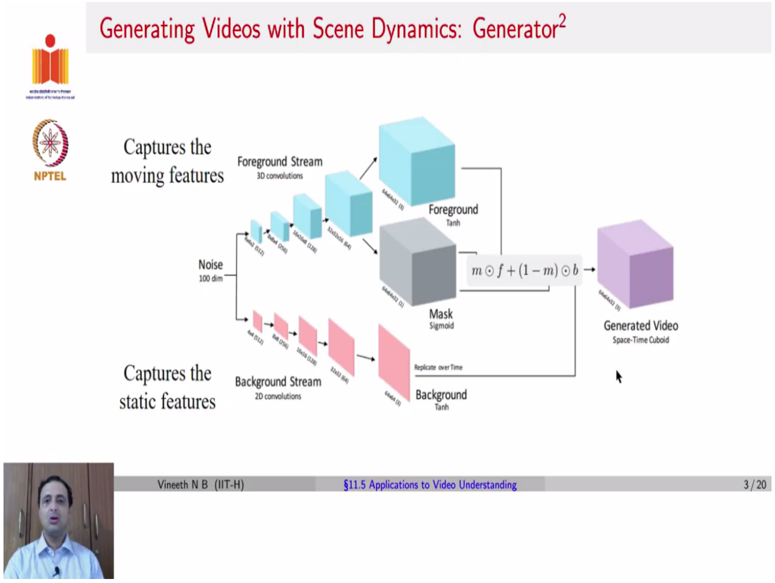
Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 2 / 20

Let us start by asking, can we use GANs to generate videos? We already answered the question partly in the previous lecture, when we talked about using GANs for 3D object generation. In a sense, videos are volumes too. More generally speaking, if you look at the GAN objective, your min max objective, where your discriminator D is parametrized by weights w_D, and your generator is parametrized by weights w_G.

G and D can take any form appropriate for a task as long as they are differentiable with respect to the parameters under consideration and this minmax objective can be solved using this objective function. Let us now see one method that extends GANs for video generation. However, with one unique change. That unique change is this method called Generating Videos with Scene Dynamics published in NeurIPS of 2016.

Considers the output of the generator to be given by $m(z) \odot f(z) + (1 - m(z)) \odot b(z)$ where z is the latent vector, which is the input to the GAN, to the generator. Here f is foreground, b is background, and m is a mask, which is also learned, which indicates whether to use foreground or background for a pixel. Let us see how this is used in a GAN architecture to generate videos.

(Refer Slide Time: 02:43)



Generating Videos with Scene Dynamics: Generator[2]

Captures the moving features

Foreground Stream
3D convolutions

4x4x2 (512)
8x8x4 (256)
16x16x8 (128)
32x32x16 (64)
64x64x32 (3)

Foreground
Tanh

Noise
100 dim

64x64x32 (1)

Mask
Sigmoid

$m \odot f + (1 - m) \odot b$

64x64x32 (3)

Generated Video
Space-Time Cuboid

Captures the static features

Background Stream
2D convolutions

4x4 (512)
8x8 (256)
16x16 (128)
32x32 (64)
64x64 (3)

Replicate over Time

Background
Tanh

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 3 / 20

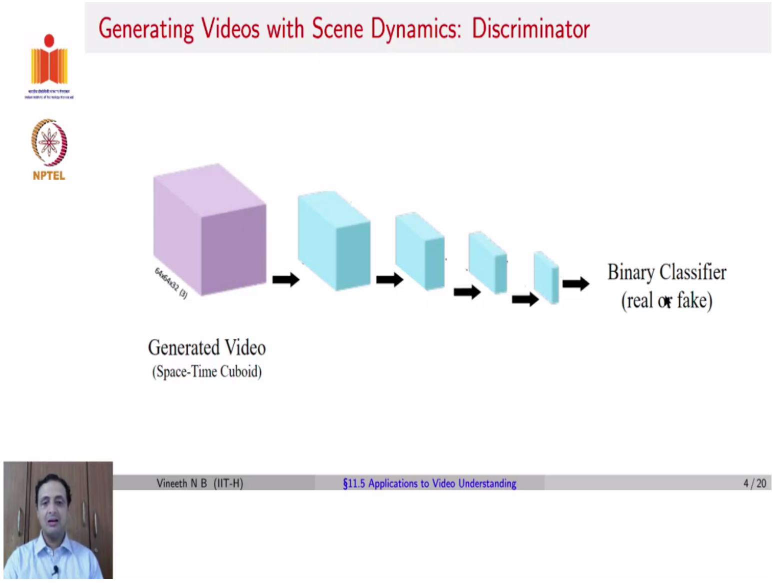
So we have this function here that we just wrote. And we have an input noise vector to the generator of a GAN which is 100 dimensional. Now, the noise is given as input to two branches, one branch which is a foreground stream which generates a volume, a 3D volume. This volume generator, this foreground stream forks into two parts at the end, where one part generates the video foreground and the other part generates a mask for every pixel in every frame of that video.

So the mask also has the same volume, same dimensions as the foreground. And the input noise also goes through another stream which generates a static background. When we talk about generating short videos, the background is not expected to change. And it is most likely that it is the foreground that changes on a static background. So in this case, the second branch of the generator generates a static image background.

And now, the foreground and the background are combined using this mask as, this mask whatever values is output by the layer here, the layer shown in blue here, $m \odot f + (1 - m) \odot b$. That is what leads to the generated output. As you can see, the foreground stream captures the moving features.

The background stream captures the static features and the background is replicated over time at every step in the frame which is the same background that is used all over again. And that leads to the final generated video which is a space-time cuboid. All of this corresponds to the generator of this model. What about the discriminator?

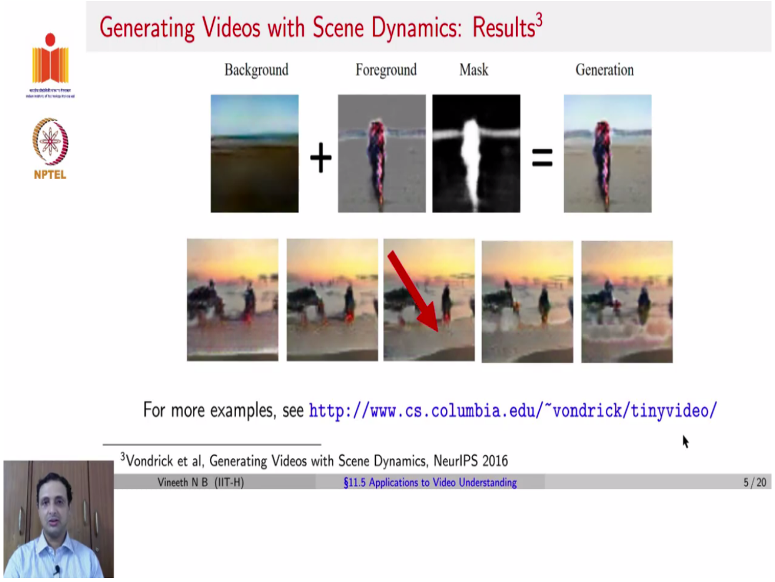
(Refer Slide Time: 05:09)



Generating Videos with Scene Dynamics: Discriminator

The discriminator is the standard discriminator in a GAN, where the generated video is down sampled over a series of layers. And the last layer has a binary classifier which says whether the video was real or fake. And the loss function remains the same as a standard GAN.

(Refer Slide Time: 05:35)



Generating Videos with Scene Dynamics: Results[3]

For more examples, see http://www.cs.columbia.edu/~vondrick/tinyvideo/

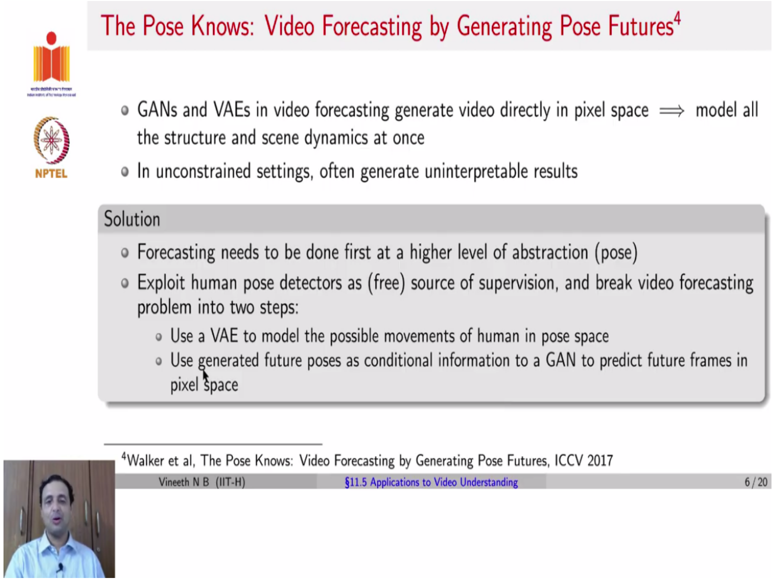
[3] Vondrick et al, Generating Videos with Scene Dynamics, NeurIPS 2016

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 5 / 20

And now we are able to generate videos with scene dynamics. Here you see snapshot examples of videos. On top, you can see a background scene that is generated, a foreground that is generated, a mask that is generated along with the foreground and together, we see the final generation to be a combination of the background and the foreground using the mask.

You see below another example, where if you observe carefully, you can see here that the waves at the bottom here seem to be receding, which shows the movement across the frames generated in the scene. If you would like to see more video examples, please go to the website of this paper right here. And you should be able to see many more examples of videos generated through this method.

(Refer Slide Time: 06:34)



The Pose Knows: Video Forecasting by Generating Pose Futures[4]

- GANs and VAEs in video forecasting generate video directly in pixel space ⟹ model all the structure and scene dynamics at once
- In unconstrained settings, often generate uninterpretable results

Solution

- Forecasting needs to be done first at a higher level of abstraction (pose)
- Exploit human pose detectors as (free) source of supervision, and break video forecasting problem into two steps:
  - Use a VAE to model the possible movements of human in pose space
  - Use generated future poses as conditional information to a GAN to predict future frames in pixel space

[4] Walker et al, The Pose Knows: Video Forecasting by Generating Pose Futures, ICCV 2017

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 6 / 20

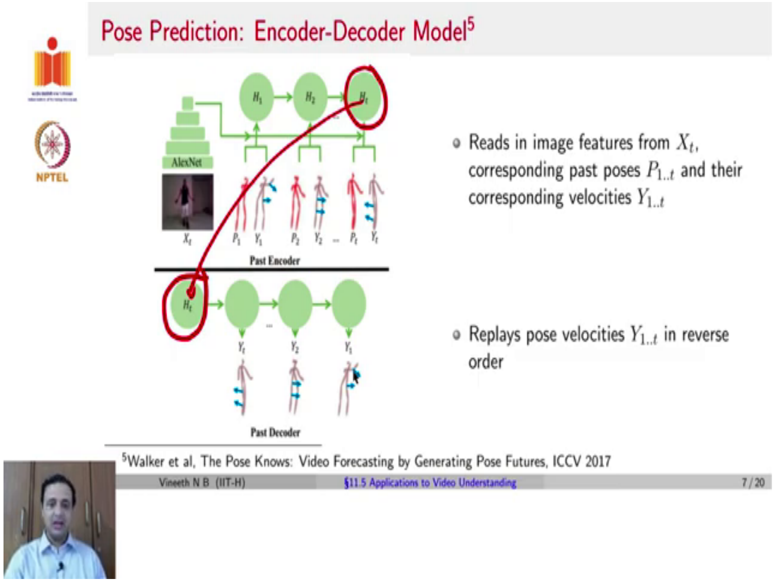
Another interesting method for generating videos in fact, a different task by itself is video forecasting. We will talk about one such method which was published in ICCV 2017, called The Pose Knows. This work argues that when GANs and VAEs are used directly for video generation they generate video directly in the pixel space each pixel is generated in every frame which means the structure and the scene dynamics are captured at the same time.

And this could often lead to uninterpretable results. To avoid this problem, this method proposes that one needs to forecast first at a higher abstraction, and then build the video based on that high level abstraction. What is that high level abstraction in this method? This method uses human pose as a high level abstraction. And it exploits human pose detectors as a free source of supervision, and thus breaks the video forecasting problem into two steps.

One, it uses a VAE to predict the possible movements of humans in pose space. And these predicted or generated human poses are used as conditional information to help the GAN predict future frames in pixel space. The task here, is given a set of frames in a video, the model needs to predict the next frame or the next set of frames. You could imagine multiple kinds of applications, you could think of a sports application.

If somebody gave you a certain configuration of football players on a football field, and you saw the last 10 seconds' video, what would happen next, whom would the person pass the ball to could be one way of looking at it. It could also have applications from a security perspective if you try to take it to a different context.

(Refer Slide Time: 09:11)

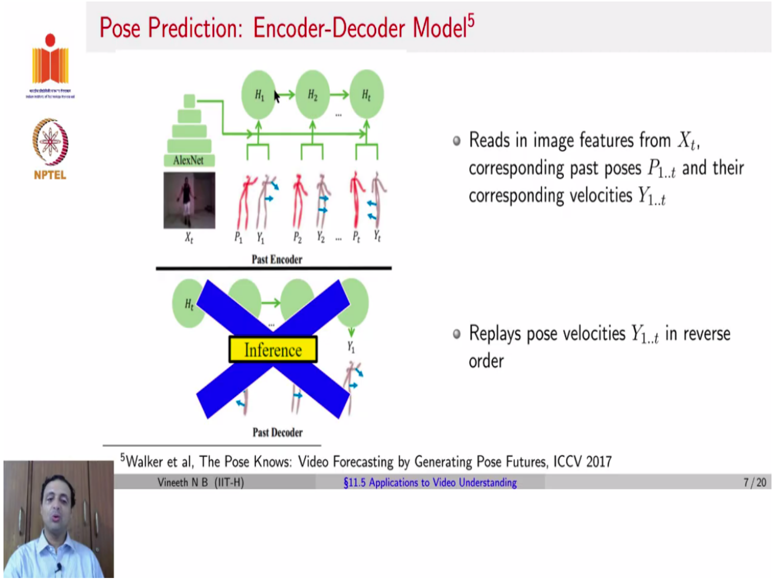


Pose Prediction: Encoder-Decoder Model[5]

- Reads in image features from X_t, corresponding past poses $P_{1..t}$ and their corresponding velocities $Y_{1..t}$
- Replays pose velocities $Y_{1..t}$ in reverse order

[5] Walker et al, The Pose Knows: Video Forecasting by Generating Pose Futures, ICCV 2017

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 7 / 20



Pose Prediction: Encoder-Decoder Model[5]

- Reads in image features from X_t, corresponding past poses $P_{1..t}$ and their corresponding velocities $Y_{1..t}$
- Replays pose velocities $Y_{1..t}$ in reverse order

[5] Walker et al, The Pose Knows: Video Forecasting by Generating Pose Futures, ICCV 2017

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 7 / 20

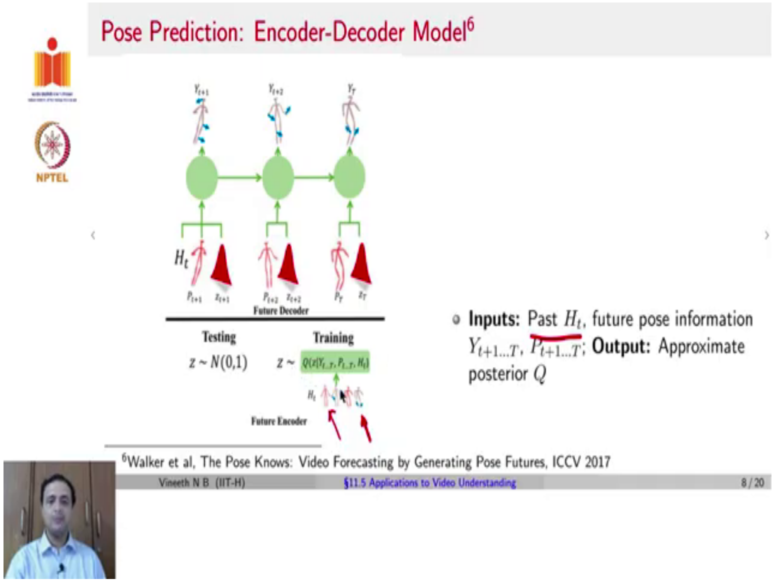
So in this method, as we just mentioned, there is a first stage where the pose of a human involved in the video is first predicted and the predicted pose or the skeleton human skeleton is then passed as conditional information to a GAN to actually generate the frames of the video. Let us first see the pose prediction part which is done using a VAE. So, in this pose prediction encoder decoder model which is a variational auto encoder in their implementation.

In the encoder, there are image features read in from $X_t$which is a particular frame at a time t, corresponding past poses, which could be P_1till P_t, those were the poses until now and also the velocities of the poses at different joint positions in each of these times 1 to t. What is velocity? It is how much is each joint location likely to move by between one frame and the next frame. You could consider these velocities as optical flow of joint locations in poses in or the skeleton pose skeletons that we get to represent human pose.

So, once the encoder reads in these image features, corresponding past poses, which in our case these past poses $P_1$to P_t, are what you see here, $P_1$to P_t and corresponding velocities $Y_1$to $y_t$give how the velocity each of the joint locations in each pose at a time t is moving towards the next time step. And the output here is the decoder which replays the pose velocities in reverse order.

So, given the output of the encoder which is H_t, which goes to the decoder, the decoder now tries to predict $Y_t$to Y_1which are the pose velocities in the reverse order. What do we do with this? Once such a model is trained, the decoder is taken away. Only the encoder is used to get these hidden states $H_1$to H_t. What do we do with that? We now use that to go to a future encoder and a future decoder.

(Refer Slide Time: 12:03)



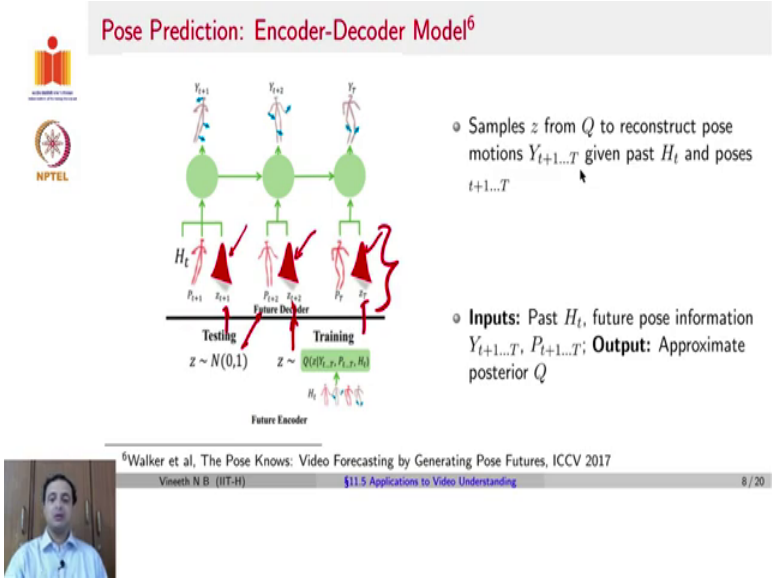
Pose Prediction: Encoder-Decoder Model[6]

- **Inputs:** Past H_t, future pose information $Y_{t+1...T}$, $P_{t+1...T}$; **Output:** Approximate posterior Q

[6] Walker et al, The Pose Knows: Video Forecasting by Generating Pose Futures, ICCV 2017

In a future encoder $H_t$is given as input. $H_t$is what we got out of the past encoder, the output of the past encoder that is given as input to the future encoder. So, the future encoder receives past H_t, future pose information $Y_{t+1\cdots T}$which are your velocities and $P_{t+1\cdots T}$which are the pose skeletons that you see here. By pose skeletons you have to recall methods like deep pose which give us an XY coordinate location for certain joint locations of a human skeleton. Since this is a Variational Auto Encoder, the encoder outputs an approximate posterior Q which is shown here. And what does the decoder do?

(Refer Slide Time: 13:06)



Pose Prediction: Encoder-Decoder Model[6]

- Samples z from Q to reconstruct pose motions $Y_{t+1...T}$ given past H_t and poses $t+1...T$
- **Inputs:** Past H_t, future pose information $Y_{t+1...T}$, $P_{t+1...T}$; **Output:** Approximate posterior Q

[6] Walker et al, The Pose Knows: Video Forecasting by Generating Pose Futures, ICCV 2017

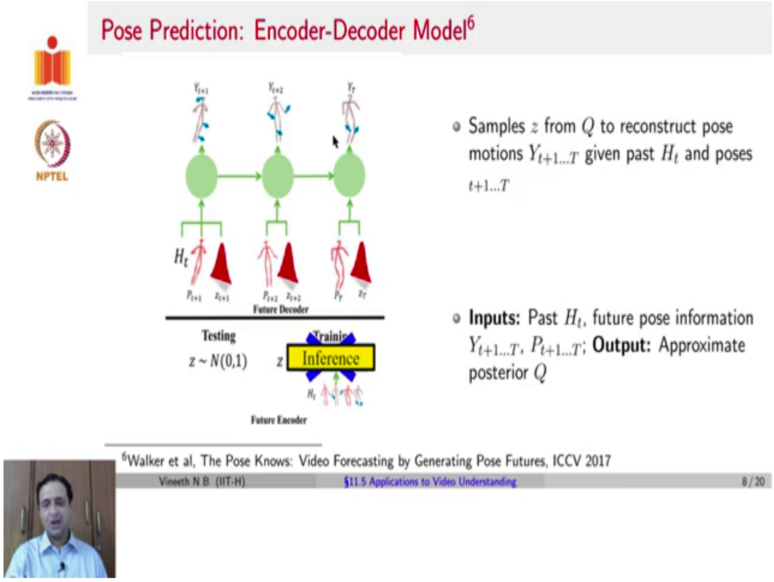
Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 8 / 20

The decoder samples from Q, not exactly Q, but from a prior that is close to the approximate posterior Q which is how the VAE learns to reconstruct the motions $Y_{t+1\cdots T}$which are your pose velocities given past $H_t$and poses $t + 1 \cdots T$.

This is shown in the input of the future decoder. So, you get a sample from your approximate from your prior sorry from your prior. Along with that you also get so, those samples here are given by z_{t+1}, z_{t+2}, $z_t$so on and so forth. Each of these red elements here are samples from a Gaussian. Along with that you also have $H_t$which comes from the encoder and not the encoder from the past encoder.

And along with that, you also have P_{t+1}, $P_{t+2}$to P_{t}. Given these, the decoder of the VAE predicts $Y_{t+1\cdots T}$which are the velocities of each joint location for times $t + 1 \cdots T$. Now, what happens once you train such a VAE?

(Refer Slide Time: 14:33)



Pose Prediction: Encoder-Decoder Model[6]

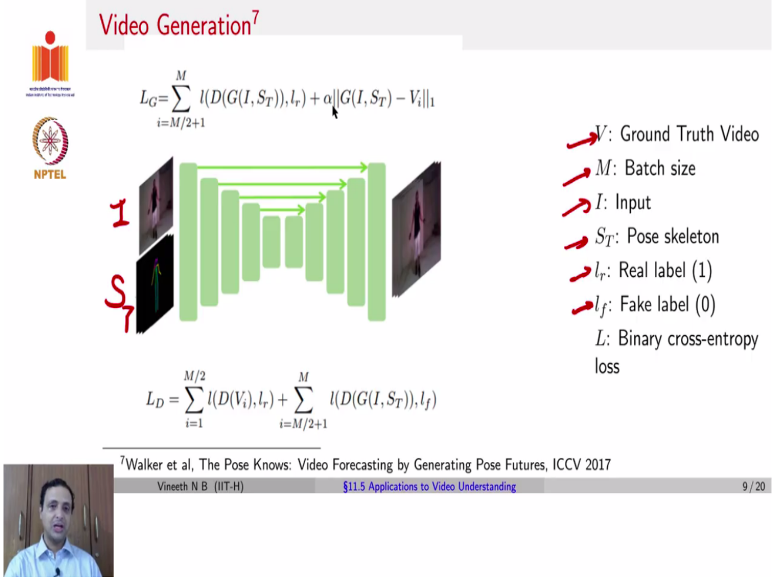
- Samples z from Q to reconstruct pose motions $Y_{t+1...T}$ given past H_t and poses $t{+}1...T$
- **Inputs:** Past H_t, future pose information $Y_{t+1...T}$, $P_{t+1...T}$; **Output:** Approximate posterior Q

[6]Walker et al, The Pose Knows: Video Forecasting by Generating Pose Futures, ICCV 2017

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 8 / 20

Like any other VAE, the encoder is not used at inference, you only sample from the prior and then let the future decoder predict how the poses will look given until a particular time $H_t$which is given by $H_t$here. The decoder tries to predict what the pose will look like from $t + 1$to T using these velocities. Remember, given a pose, and given the velocities, one can construct the pose in all of those time steps in $t + 1 \cdots T$just by adding the velocity to the current pose location.

(Refer Slide Time: 15:18)



Video Generation[7]

$$L_G = \sum_{i=M/2+1}^{M} l(D(G(I, S_T)), l_r) + \alpha \| G(I, S_T) - V_i \|_1$$

V: Ground Truth Video
M: Batch size
I: Input
S_T: Pose skeleton
l_r: Real label (1)
l_f: Fake label (0)
L: Binary cross-entropy loss

$$L_D = \sum_{i=1}^{M/2} l(D(V_i), l_r) + \sum_{i=M/2+1}^{M} l(D(G(I, S_T)), l_f)$$

[7] Walker et al, The Pose Knows: Video Forecasting by Generating Pose Futures, ICCV 2017

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 9 / 20

Now coming to the second stage, once the pose information is obtained, remember we said that the pose information is given as a conditional input to the generator which finally generates the videos. Let us see that part now. So, this is the generator which, given a set of frames, which we denote as input I at this time, and the pose information which comes from the previous module that we just discussed.

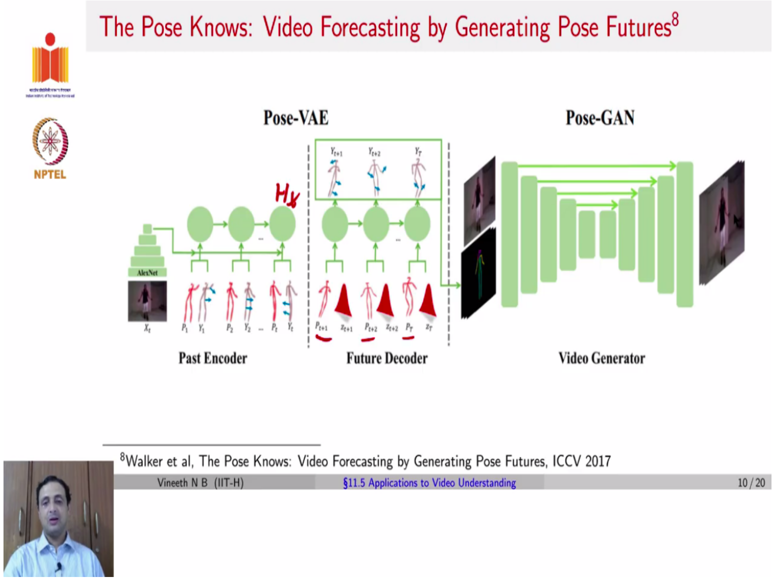
The generator generates the next few frames which is the goal of video forecasting. Now, this is a generator module. This is similar to a GAN module, so it has a loss for the discriminator and a loss for the generator. Let us see each of them. So the discriminator loss has $l(D(V_i), l_r)$. What is V_i? V_i is the ground truth video. What is l_r? Real label. So we would like the discriminator to assign the real label to the ground truth video.

And on the other hand, we would like the discriminator to consider the generation of the input and S_T denotes the pose skeleton, I denotes the input, this is input, and this is S_T the pose skeleton. Given these two, the generator generates the future frames. And the discriminator's job is to give that a fake label. What are these summations here? These summations are over a mini batch whose size is M.

So one half of the mini batch are real videos, ground truth videos and the other half of the mini batch are generated videos which are given by this term. This is what the discriminator tries to do. Let us try to see what the generator tries to do. The generator which has a counter objective wants the discriminator to look at the generator’s output and give it a real label. In addition to this generator objective, we also try to minimize the L_1 distance between the generated future frames and the future frames that are provided to us in the ground truth V_i and the L_1 term is weighted by a coefficient α.

This together helps train a model that can generate videos where in the first stage poses are predicted and the poses are then passed on for the GAN to generate videos.

(Refer Slide Time: 18:15)



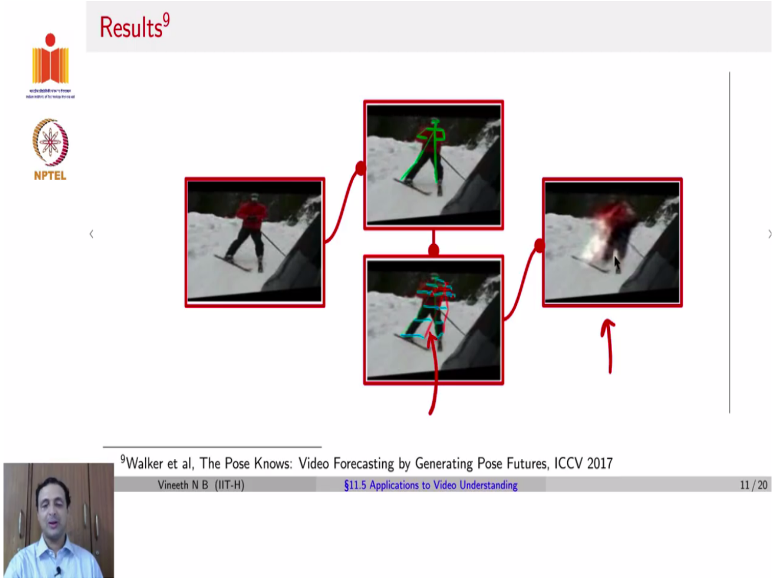
The Pose Knows: Video Forecasting by Generating Pose Futures[8]

[8] Walker et al, The Pose Knows: Video Forecasting by Generating Pose Futures, ICCV 2017

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 10 / 20

So the final end to end architecture looks like this. There is a past encoder that would generate the H_t. That H_t is given as input to the future decoder along with poses P_{t+1} to P_T. Now this generates Y_{t+1} to Y_t which are the predictions of the pose, which are given as input to the GAN which finally uses that and the input so far to be able to generate the future frames.

(Refer Slide Time: 18:56)



Results[9]

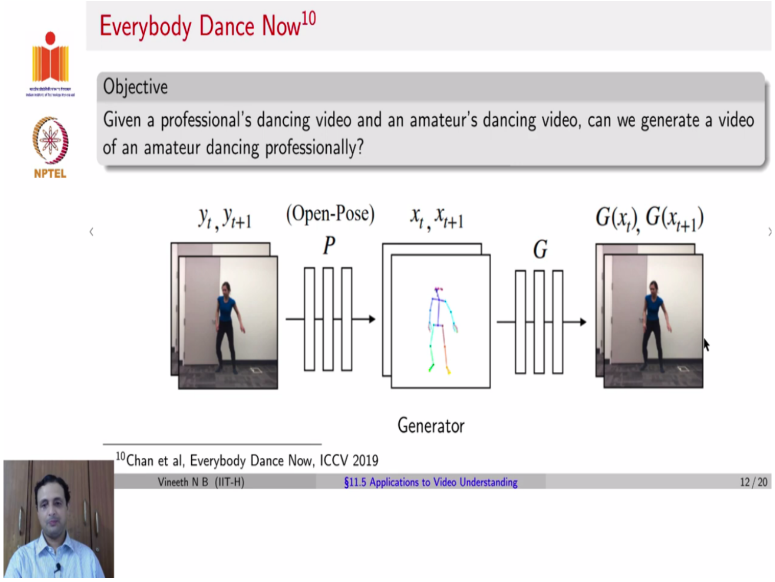
[9]Walker et al, The Pose Knows: Video Forecasting by Generating Pose Futures, ICCV 2017

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 11 / 20

With this approach, this method shows interesting results. Here is an example of a person skiing on a snow slope, snowy slope. You can see the pose skeleton here and the predicted movement here. So these blue lines represent optical flow or the velocities at each of those joint locations which also tells us if you see this red skeleton that is the skeleton where the person moves to from this green skeleton in the current location.

Now using that skeleton to condition the generation of the next frame, the GAN generates this image where the person moves to that location. Clearly, it is not very sharp in terms of the generation considering the real world situation, but it gets a fairly good estimate of where the person is likely to be in future of the video that is provided to us.

(Refer Slide Time: 20:04)



Everybody Dance Now[10]

Objective

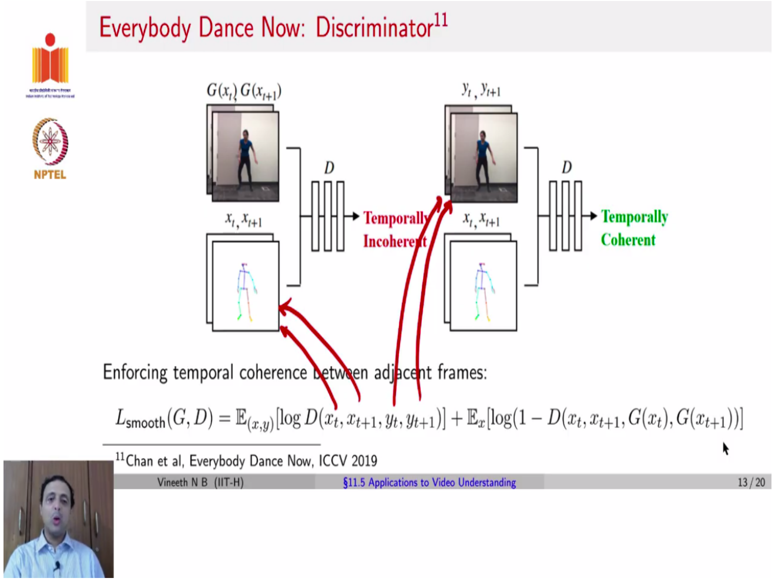
Given a professional's dancing video and an amateur's dancing video, can we generate a video of an amateur dancing professionally?

[10] Chan et al, Everybody Dance Now, ICCV 2019

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 12 / 20

A third method in Video understanding that we will discuss is an interesting one called Everybody Dance Now, a recent work published in ICCV of 2019, where the objective is, given a professional's dancing video, and an amateur's dancing video, can we generate a video of an amateur dancing professionally? And one of the basic modules of this framework tries to have a generator which captures the temporal coherence between frames using the ground truth information and the generated information.

(Refer Slide Time: 20:54)



Everybody Dance Now: Discriminator[11]

Enforcing temporal coherence between adjacent frames:

$$L_{\text{smooth}}(G,D) = \mathbb{E}_{(x,y)}[\log D(x_t, x_{t+1}, y_t, y_{t+1})] + \mathbb{E}_x[\log(1 - D(x_t, x_{t+1}, G(x_t), G(x_{t+1}))]$$

[11] Chan et al, Everybody Dance Now, ICCV 2019

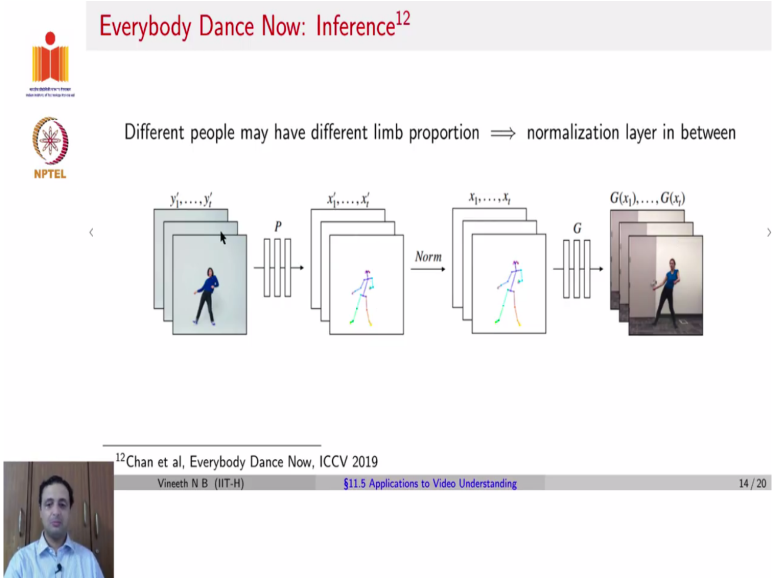
Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 13 / 20

So once we have this generated information, so you have $G(x_t)$, and $G(x_{t+1})$; $x_t$and $x_{t+1}$are the poses in two successive frames. $G(x_t)$and $G(x_{t+1})$are the generations corresponding to those poses. And $y_t$and $y_{t+1}$are the actual ground truths for that person's pose corresponding to the skeleton x_t, x_{t+1}. The job of the discriminator in this module of the framework is to look at the generated frames, $G(x_t)$and $G(x_{t+1})$and see whether they are temporally incoherent, which is equivalent to fake here.

And for real data, the label given is temporally coherent or real. So it is no more real versus fake. The discriminator tries to say, temporally coherent or temporally incoherent. So your standard GAN loss looks like you have the log likelihood of discriminator looking at x_t, x_{t+1}which are the pose information of a frame at time t and the pose information at time $t + 1$, $y_t$and $y_{t+1}$are the expected frames the ground truth frames for the same pose locations and $G(x_t)$and $G(x_t)$are the generated frames.

So this GAN module of this framework tries to ensure that consequent frames or successive frames of the video being generated is temporally coherent, like a given input video. This is one module of this work.

(Refer Slide Time: 22:56)



Everybody Dance Now: Inference[12]

Different people may have different limb proportion ⟹ normalization layer in between

[12] Chan et al, Everybody Dance Now, ICCV 2019

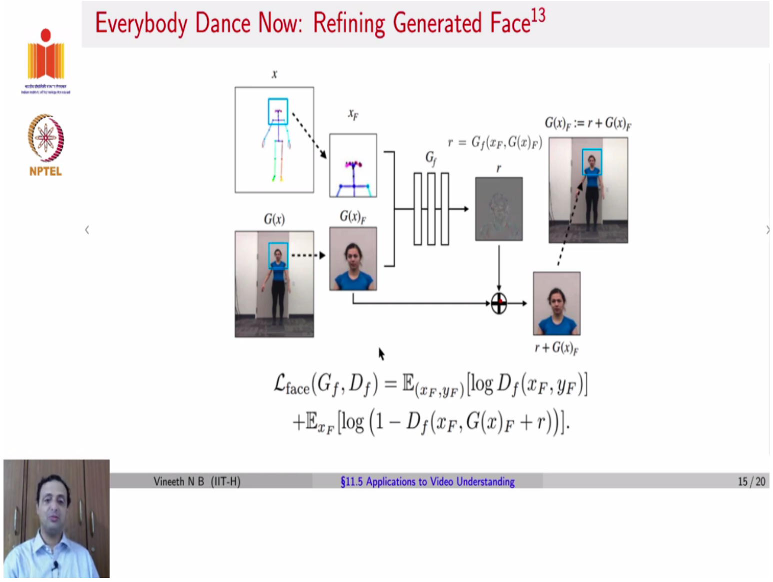
Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 14 / 20

Another module that is part of this work is to ensure that when you have a source person's dance moves, so in this case, you have $y_1', \cdots, y_t'$which are the video frames of a source person, a professional dancer. So the pose of the professional dancer is obtained to get $x_1', \cdots, x_t'$. Since this has to be overlaid on the target persons pose. This is normalized to get the pose information in the dimensions of the person that is being considered.

So different people may have different limp proportions. So this normalization step moves the skeleton of the source person that is the professional dancer to the skeleton of the amateur dancer in terms of sizes and distances between the joints. Given this $x_1$to x_t, the generator module in the previous step we had trained a generator that could take pose information and generate frames.

Now the pose information is obtained using this architecture of coming from a professional's skeleton normalized into the dimensions of the target person and the generator now generates the generator trained, as in the previous step now generates videos of the target person with the same pose moves of the professional person.

(Refer Slide Time: 24:41)



Everybody Dance Now: Refining Generated Face[13]

$$\mathcal{L}_{\text{face}}(G_f, D_f) = \mathbb{E}_{(x_F, y_F)}[\log D_f(x_F, y_F)]$$
$$+\mathbb{E}_{x_F}[\log\left(1 - D_f(x_F, G(x)_F + r)\right)].$$

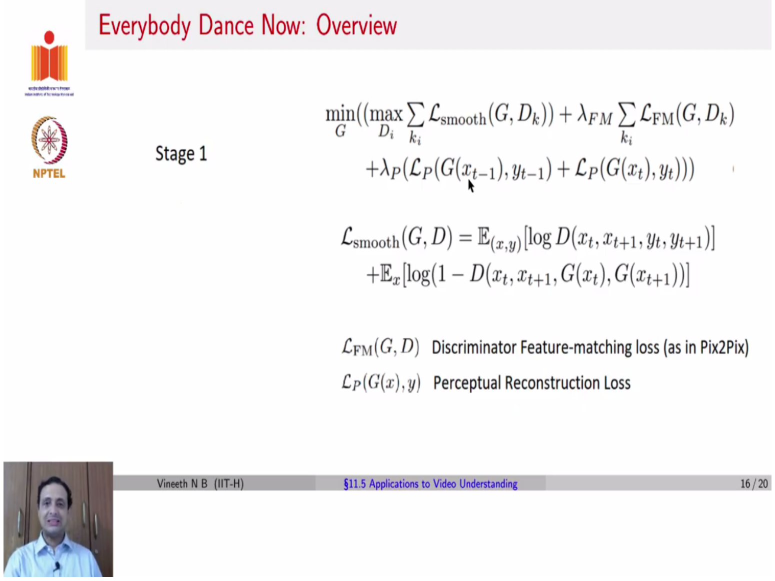
Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 15 / 20

Finally, because in the process of transposing someone else's moves on to a target person there could be distortions and loss of detail on the face of the target person, this method also introduces one module to refine the generated face. So this in this module, this is another GAN within the overall framework. In this module, the face portion of the generated image. So this $G(x)$that was generated in the previous step, a certain region around the face around the nose of the person is cropped out which is given by $G(x)_F$.

And $x_F$is also the pose drop cropped around the same area. These two are given to a new generator module which outputs a residual that needs to be added to the original face to add more detail. This residual is then added to the original generated face to give the final outcome which is then transposed onto the final generation. This is a separate GAN by itself where the discriminator tries to maximize the likelihood of x_F, y_F.

Where $x_F$is the pose skeleton around the face, $y_F$is the actual face region in the target image and $G(x)_F + r$is the generated face through this generator where the residual is the output of the generator. And this helps this GAN helps refine the face.

(Refer Slide Time: 26:38)



Everybody Dance Now: Overview

Stage 1

$$\min_{G}((\max_{D_i}\sum_{k_i}\mathcal{L}_{\text{smooth}}(G,D_k))+\lambda_{FM}\sum_{k_i}\mathcal{L}_{\text{FM}}(G,D_k)$$
$$+\lambda_P(\mathcal{L}_P(G(x_{t-1}),y_{t-1})+\mathcal{L}_P(G(x_t),y_t)))$$

$$\mathcal{L}_{\text{smooth}}(G,D)=\mathbb{E}_{(x,y)}[\log D(x_t,x_{t+1},y_t,y_{t+1})]$$
$$+\mathbb{E}_x[\log(1-D(x_t,x_{t+1},G(x_t),G(x_{t+1}))]$$

$\mathcal{L}_{\text{FM}}(G,D)$ Discriminator Feature-matching loss (as in Pix2Pix)

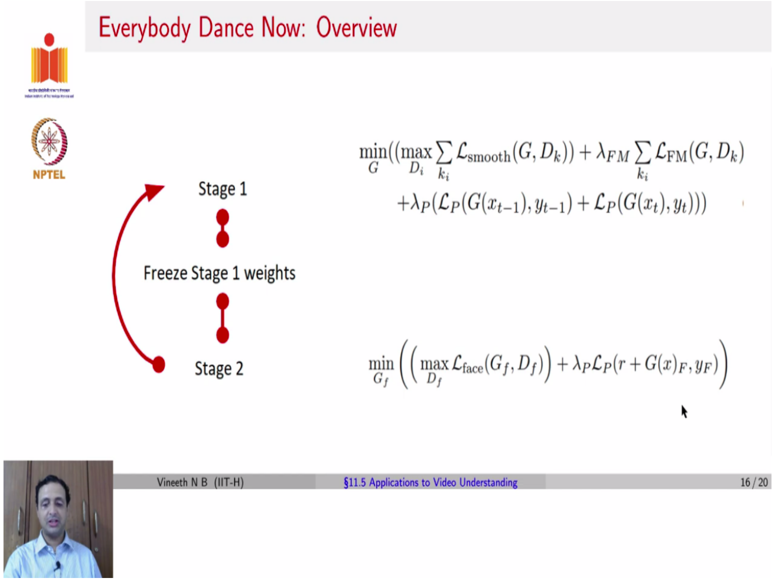
$\mathcal{L}_P(G(x),y)$ Perceptual Reconstruction Loss

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 16 / 20

Putting these together in stage 1 in this particular framework. The first GAN is trained and this GAN has a few components. It has an $\backslash L_{smooth}(G,\ D_k)$, which is the standard GAN loss that we already saw. So this is your standard GAN loss that ensured temporal coherence. L_{FM}gives a discriminator feature matching loss, very similar to pix2pix between the generated image and the real image that comes from a discriminator.

And there is also an $L_P$loss, which gives a perceptual reconstruction loss, once again comes from pix2pix. And this is for $G(x_{t-1})$matching to $y_{t-1}$and $G(x_t)$matching to y_t. Remember the perceptual loss also tries to measure loss at an intermediate feature space feature space level. And this is stage 1.

(Refer Slide Time: 27:50)



Everybody Dance Now: Overview

Stage 1

Freeze Stage 1 weights

Stage 2

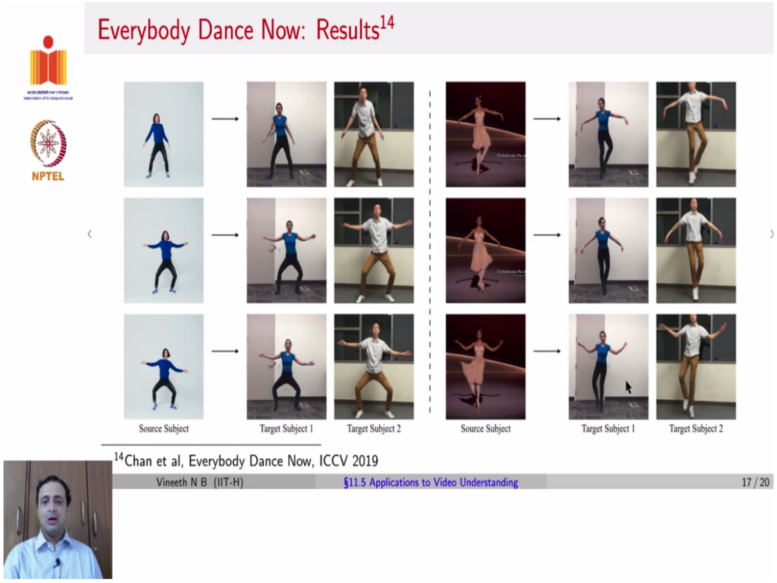
$$\min_G((\max_{D_i}\sum_{k_i}\mathcal{L}_{\text{smooth}}(G,D_k))+\lambda_{FM}\sum_{k_i}\mathcal{L}_{\text{FM}}(G,D_k)$$
$$+\lambda_P(\mathcal{L}_P(G(x_{t-1}),y_{t-1})+\mathcal{L}_P(G(x_t),y_t)))$$

$$\min_{G_f}\left(\left(\max_{D_f}\mathcal{L}_{\text{face}}(G_f,D_f)\right)+\lambda_P\mathcal{L}_P(r+G(x)_F,y_F)\right)$$

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 16 / 20

Once this GAN is trained the stage 1 weights are frozen, and then comes stage 2, where the face GAN is trained based on the L face which is the standard GAN loss for the face region and a perceptual loss also defined for the face region. Once stage 2 is trained, stage 1 is again retrained. And this goes on in iterations to be able to get the final generations.

(Refer Slide Time: 28:23)



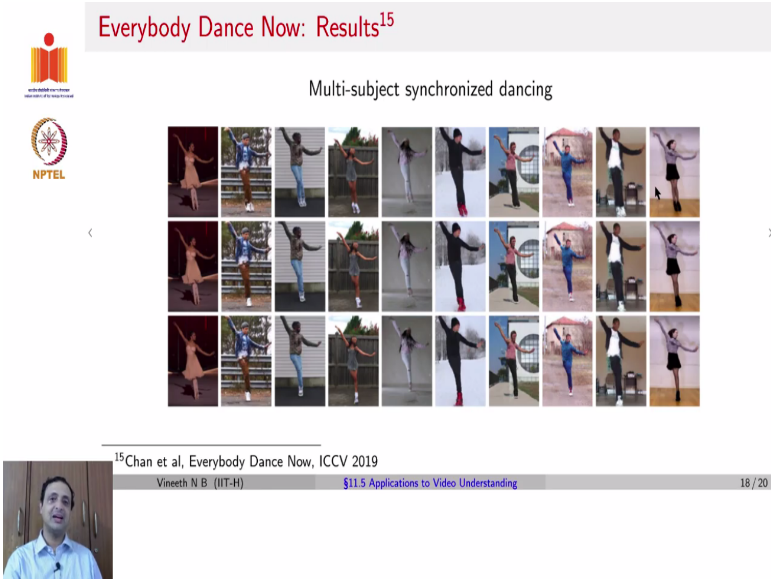
Everybody Dance Now: Results[14]

[14] Chan et al, Everybody Dance Now, ICCV 2019

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 17 / 20

Now here are some interesting results. Here is a source subject and the source subjects corresponding poses. And these poses are now translated to a target subject who now show the same pose in different scales, in different backgrounds with different body structures, including male and female. A similar example is also seen on the right side here.

(Refer Slide Time: 28:55)



Everybody Dance Now: Results[15]

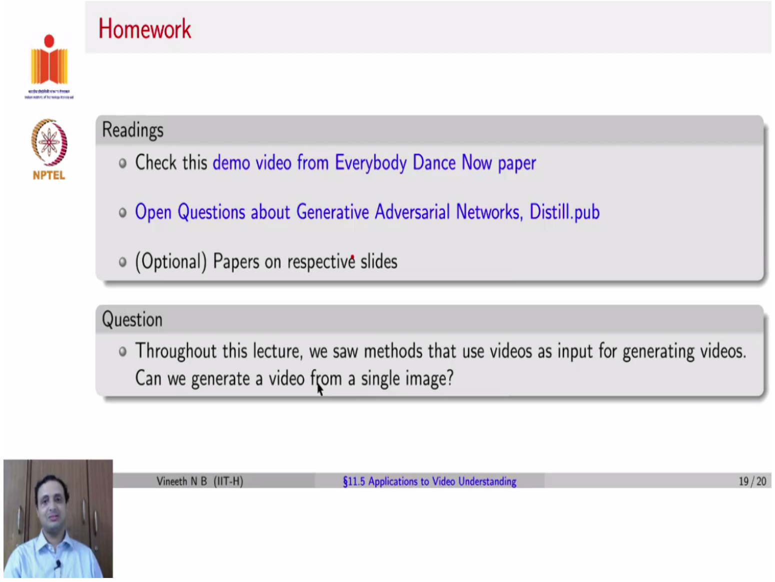
Multi-subject synchronized dancing

[15] Chan et al, Everybody Dance Now, ICCV 2019

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 18 / 20

The paper also shows interesting results of multi-subject synchronized dancing, where the source subjects poses are transposed onto multiple different subjects of different backgrounds, different sizes, different genders. And the same pose is now shown by all of those people to give a sense of synchronized dancing.

(Refer Slide Time: 29:22)



Homework

Readings

- Check this demo video from Everybody Dance Now paper
- Open Questions about Generative Adversarial Networks, Distill.pub
- (Optional) Papers on respective slides

Question

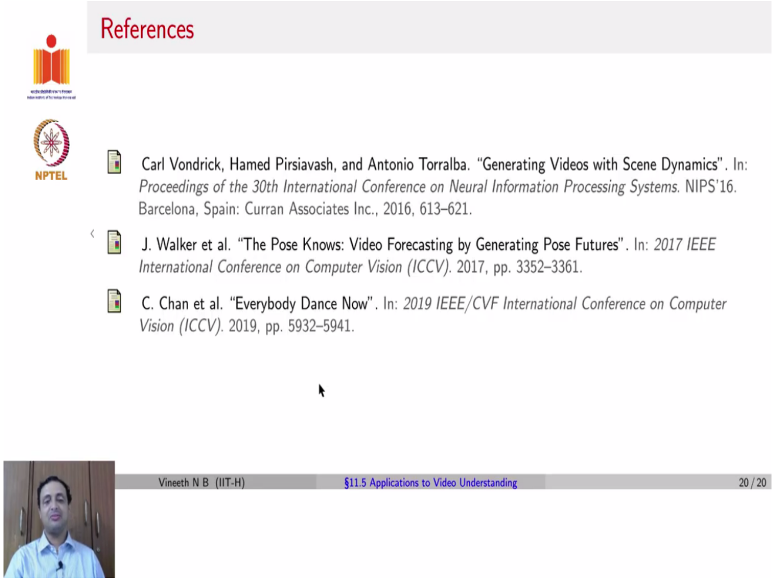
- Throughout this lecture, we saw methods that use videos as input for generating videos. Can we generate a video from a single image?

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 19 / 20

Your homework for this lecture is to check out this demo video from everybody dance now paper, which is fairly interesting and also a very nice article on open questions on GANs to wrap up our discussions so far on generative models. And of course, if you are interested, please do read the papers on the respective slides.

Let us end this lecture with one question. Through this lecture, we saw methods that used videos as input where the discriminator looked at the real video and the generated video and said whether it was real or fake or temporally coherent or incoherent. Can you generate a video from a single image? Think about it, and we will discuss soon.

(Refer Slide Time: 30:18)



References

- Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. "Generating Videos with Scene Dynamics". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2016, 613–621.
- J. Walker et al. "The Pose Knows: Video Forecasting by Generating Pose Futures". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 3352–3361.
- C. Chan et al. "Everybody Dance Now". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 5932–5941.

Vineeth N B (IIT-H) §11.5 Applications to Video Understanding 20 / 20

Here are references.