Literature - My Notes

Nagabhushan S N

June 30, 2020

Contents

1 Study of Subjective and Objective Quality Assessment of Video (LIVE VQA) 4

2 Video (Language) Modeling: A Baseline for Generative Models of Natural Videos 5

3 Unsupervised Learning of Video Representations using LSTMs 6

4 Action-Conditional Video Prediction using Deep Networks in Atari Games 8

5 Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting 9

6 Sports Videos in the Wild (SVW): A Video Dataset for Sports Analysis 10

7 Deep Multi-Scale Video Prediction Beyond Mean Square Error 11

8 Learning Visual Predictive Models of Physics for Playing Billiards 14

9 Dynamic Filter Networks 16

10 Improved Techniques for Training GANs 17

11 Learning to Poke by Poking: Experiential Learning of Intuitive Physics (NIPS 2016) 19

12 Unsupervised Learning for Physical Interaction through Video Prediction 21

13 Visual Dynamics: Probabilistic Future Frame Synthesis via Cross Convolutional Networks 23

14 Generating Videos with Scene Dynamics 24

15 Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (DC GAN) 25

16 Flexible Spatio-Temporal Networks for Video Prediction 26

17 Generating the Future with Adversarial Transformers 27

18 Self-Supervised Visual Planning with Temporal Skip Connections 28

19 Dual Motion GAN for Future-Flow Embedded Video Prediction 29

20 Least Squares Generative Adversarial Networks (LS GAN) - ICCV 2017 30

21 Video Frame Synthesis using Deep Voxel Flow (DVF) 32

22 Decomposing Motion and Content for Natural Video Sequence Prediction 33

23 Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning 36

24 Learning to Generate Long-term Future via Hierarchical Prediction 37

25 Video Pixel Network (VPN) 38

26 Learning to See Physics via Visual De-animation 39

27 Transformation-based Models of Video Sequences 42

28 Deep Image Prior 43

29 Future Frame Prediction for Anomaly Detection - A New Baseline 44

30 Structure Preserving Video Prediction 45

31 The Unreasonable Effectiveness of Deep Features as a Perceptual Metric 46

32 ContextVP: Fully Context-Aware Video Prediction 47

33 DYAN: A Dynamical Atoms-Based Network For Video Prediction 48

34 Folded Recurrent Neural Networks for Future Video Prediction (FRNN) 50

35 Probabilistic Video Generation using Holistic Attribute Control (VideoVAE) 51

36 SDC-Net: Video prediction using spatially-displaced convolution 52

37 Stochastic Variational Video Prediction (SV2P) 53

38 Hierarchical Long-term Video Prediction without Supervision 56

39 Stochastic Video Generation with a Learned Prior (SVG-LP) - ICML 2018 57

40 Video Prediction with Appearance and Motion Conditions (AMC-GAN) 58

41 Learning to Decompose and Disentangle Representations for Video Prediction (DDPAE) 59

42 Video Prediction via Selective Sampling (VPSS) (2018-NIPS) 60

43 FutureGAN: Anticipating the Future Frames of Video Sequences using Spatio-Temporal 3d Convolutions in Progressively Growing GANs 62

44 Reduced-Gate Convolutional LSTM Using Predictive Coding for Spatiotemporal Prediction 63

45 Stochastic Adversarial Video Prediction (SAVP) 64

46 Stochastic Dynamics for Video Infilling (SDVI) 68

47 TGANv2: Efficient Training of Large Models for Video Generation with Multiple Subsampling Layers 69

48 Quality Assessment of In-the-Wild Videos 70

49 Order Matters: Shuffling Sequence Generation for Video Prediction (SEE Net) 71

50 Predicting Future Frames using Retrospective Cycle GAN 72

51 Disentangling Propagation and Generation for Video Prediction 73

52 Spatio-Temporal Measures Of Naturalness 74

53 Bounce and Learn - Modeling Scene Dynamics with Real-World Bounces (ICLR 2019, CMU) 76

54 FVD - A new Metric for Video Generation 78

55 Reasoning about Physical Interactions with Object-Oriented Prediction and Planning (O2P2) 80

56 Time Agnostic Prediction (Conference Paper - ICLR 2019) 83

57 Deep Compressed Sensing 84

58 ChainQueen: A Real-Time Differentiable Physical Simulator for Soft Robotics 87

59 High Fidelity Video Prediction with Large Stochastic Recurrent Neural Networks 88

60 Large-Scale Study of Perceptual Video Quality (LIVE VQC Database) 89

61 Predicting the Quality of Images Compressed After Distortion in Two Steps 91

62 Event-driven Video Frame Synthesis 94

63 From Here to There: Video Inbetweening Using Direct 3D Convolutions 95

64 Physics-as-Inverse-Graphics: Joint Unsupervised Learning of Objects and Physics from Video 96

65 Scaling Autoregressive Video Models 97

66 VideoFlow: A Flow-Based Generative Model for Video 98

67 Probabilistic Video Prediction from Noisy Data with a Posterior Confidence (BP-Net) 99

68 Learning Human Objectives by Evaluating Hypothetical Behavior 100

1 Study of Subjective and Objective Quality Assessment of Video (LIVE VQA)

Paper Details

- Authors: Kalpana Seshadrinathan, Rajiv Soundararajan, Alan Bovik, Lawrence Cormack
- Institutions: UT Austin
- Project website: https://live.ece.utexas.edu/research/Quality/live_video.html
- Code:
- Published in: TIP 2010 June
- Citations: 956 (As of 12-02-2020)

Abstract

1.1 Introduction

- The only reliable method to assess the video quality perceived by a human observer is to ask human subjects for their opinion, which is termed subjective video quality assessment (VQA).
- 10 uncompressed reference videos, 150 distorted videos, 38 subjects

1.2 Details of Subjective Study

- Reference videos are taken from Technical University of Munich (TUM)

1.3 Processing of Subjective Scores

1.4 Objective VQA Algorithms

2 Video (Language) Modeling: A Baseline for Generative Models of Natural Videos

Paper Details

- Authors: Marc' Aurelio Ranzato, Michael Mathieu
- Institutions: Facebook AI Research
- Project website: https://research.fb.com/publications/video-language-modeling-a-baseline-for-gener
- Code:
- Published in: 2014
- Citations: 242 (As of 20-10-2019)

Abstract

- Strong baseline for unsupervised feature learning using video data. Features are learnt by learning to predict missing frames or extrapolate future frames.

2.1 Introduction

- Motivation:
- Contributions:
  - –

2.2 Model

2.3 Experiments

- Datasets: UCF-101, van Hateren
- Baselines:
- Evaluation Metrics: RMSE,

3 Unsupervised Learning of Video Representations using LSTMs

Paper Details

- Authors: Nitish Srivastava
- Institutions: University of Toronto
- Project website:
- Code: https://github.com/mansimov/unsupervised-videos
- Published in: ICML 2015
- Citations: 1109 (As of 19-10-2019)

Abstract

- LSTMs to learn representations of video sequences.
- Encoder maps input sequence to fixed length representation.
- Decoder decodes this to reconstruct the input sequence or predict future sequence.
- **These video representations help improve classification accuracy (eg: action recognition), especially when there are only few training examples.**

3.1 Introduction

- Motivation:
- Contributions:
  -

3.2 Model Description

3.2.1 LSTM Autoencoder Model

3.2.2 LSTM Future Predictor Model

3.2.3 Conditional Decoder

3.2.4 Composite Model

- The two tasks: Reconstructing the input and predicting the future are combined in the composite model.
- This composite model tries to overcome the shortcomings that each model suffers on its own.
- A high-capacity autoencoder would suffer from the tendency to learn trivial representations that just memorize the inputs. However, this memorization is not useful at all for predicting the future. Therefore, the composite model cannot just memorize information.
- On the other hand, the future predictor suffers form the tendency to store information only about the last few frames since those are most important for predicting the future. But if we ask the model to also predict all of the input sequence, then it cannot just pay attention to the last few frames.

3.3 Experiments

3.3.1 Training

- Optimization: RMSProp

3.3.2 Datasets

- Moving MNIST
- UCF-101: 13,320 videos; Average length 6.2s; 101 different action categories; 9500 videos in train split.
- HMDB-51: 5100 videos; Average length 3.2s; 51 different action categories; 3570 videos in train split.
- Sports-1M: Used to train only unsupervised models; Resolution: 240x320; Center-cropped to 224x224.

3.3.3 Visualization and Qualitative Analysis

- Moving MNIST: Context: 10 frames; Prediction: 10 frames; Logistic output units; Cross-Entropy Loss;
- UCF-101: Conditional Future Predictor; 32x32 patches; ReLU output units; Squared Loss; Context: 16 frames; Prediction: 13 frames;

3.3.4 Action Recognition on UCF-101/HMDB-51

3.3.5 Comparison of Different Model Variants

- Composite Model always does a better job of predicting the future compared to the Future Predictor.
- Baselines:
- Evaluation Metrics: Cross-Entropy for MNIST and Squared loss for UCF-101.

4 Action-Conditional Video Prediction using Deep Networks in Atari Games

Paper Details

- Authors: Junhyuk Oh
- Institutions: University of Michigan
- Project website: https://sites.google.com/a/umich.edu/junhyuk-oh/action-conditional-video-predicti
- Code: https://github.com/junhyukoh/nips2015-action-conditional-video-prediction
- Published in: NIPS 2015
- Citations: 434 (As of 22-10-2019)

Abstract

- Approximately 100-step action-conditional futures can be predicted.

4.1 Introduction

- Motivation:
- Contributions:
  - –

4.2 Related Work

4.3 Proposed Architectures and Training Method

- Loss Functions: MSE

4.4 Experiments

- Datasets:
- Baselines:
- Evaluation Metrics: MSE

5 Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting

Paper Details

- Authors:
- Institutions: Hong Kong University of Science and Technology, China
- Project website:
- Code:
- Published in: NIPS 2015
- Citations: 1161 (As of 28-09-2019)

Abstract

5.1 Introduction

- **Highlights**:
  - **One of the first Video Prediction papers**
  - **Introduced ConvLSTM**
- Motivation: Weather Forecasting
- Contributions:
  - ConvLSTM

5.2 Preliminaries

5.3 The Model

- Loss Functions:

5.4 Experiments

- Datasets: Moving MNIST, Radar Echo Dataset.
- Baselines: ROVER, FC-LSTM
- Evaluation Metrics: Rainfall-MSE, Critical Success Index (CSI), False Alarm Rate (FAR), Probability Of Detection (POD), and correlation.

6 Sports Videos in the Wild (SVW): A Video Dataset for Sports Analysis

Abstract

- 4200 videos
- Captured in smartphone
- Coordinates of a rectangle enclosing action is provided.

7 Deep Multi-Scale Video Prediction Beyond Mean Square Error

Paper Details

- Authors: Michael Mathieu, Camille Couprie, Yann LeCun
- Institutions: New York University, Facebook Artificial Intelligence Research
- Project website: https://cs.nyu.edu/~mathieu/iclr2016.html
- Code: https://github.com/coupriec/VideoPredictionICLR2016
- Published: ICLR-2016.
- Citations: 703 (As of 11-Aug-2019)

Abstract

Pixel-space video prediction

- Multi-scale architecture
- Adversarial Training Method
- Image Gradient Difference Loss Function

7.1 Introduction

- Motivation: Unsupervised Learning, Robotics, Video compression, Inpainting
- Action recognition algorithm need Supervised Training for months with heavily labelled data. This can be reduced by unsupervised learning.
- Contributions:
  - Adversarial Loss
  - Image Gradient Difference Loss
  - Multi-scale architecture
  - Sharpness measure

7.2 Models

- ConvNet: Alternating Convolutions and ReLU layers
- Notations:
  - X: Sequence of Input Frames.
  - G(X): Sequence of predicted frames.
  - Y: Sequence of actual future frames.
- $\mathcal{L}_p(X, Y) = l_p(G(X), Y) = \|G(X) - Y\|_p^p$
- Two major flaws:
  1. Convolutions only account for short-range dependencies, limited by the size of their kernels.
     - Pooling can only be part of solution: Pooling reduces resolution but we want the output to be of same resolution as input.
     - Simplest and oldest: No pooling, just many convolution layers.
     - Skip connections.
     - Multi-scale architecture using Laplacian Pyramid: This is followed in the paper.
  2. Using l_2 loss produces blurry predictions:

- Suppose there are 2 possibilities: v_1, v_2. Using either of them adds $(v_1 - v_2)^2$ (Or 0) to loss but using $v_{avg} = (v_1 + v_2)/2$ adds a loss of $(v_1 - v_2)^2/4$ every time. Thus, v_{avg} minimizes l_2 loss, but creates blurry predictions.
- Using l_1 norm minimizes this since l_1 norm picks median of the set of equally likely values.

7.2.1 Multi-Scale Network

- To tackle problem 1 i.e. Convolutions account only for short range dependencies.
- W_G: Set of trainable parameters.
- SGD is used for minimization.

7.2.2 Adversarial Training

- To tackle problem 2 i.e. blurry predictions.
- Discriminator can easily identify $v_{avg} = (v_1 + v_2)/2$ as fake. Thus blurry predictions are avoided.
- Discriminative model is a multi-scale ConvNet with single scalar output.
- SGD is used to train the discriminative model.
- Loss functions:

$$\mathcal{L}^D_{adv}(X, Y) = \sum_{k=1}^{N_{scales}} L_{\text{bce}}(D_k(X_k, Y_k), 1) + L_{\text{bce}}(D_k(X_k, G_k(X)), 0)$$
$$L_{\text{bce}}(Y, \hat{Y}) = -\sum_i \hat{Y}_i \log(Y_i) + (1 - \hat{Y}_i)\log(1 - Y_i)$$
$$\mathcal{L}^G_{adv}(X, Y) = \sum_{k=1}^{N_{scales}} L_{\text{bce}}(D_k(X_k, G_k(X_k)), 1)$$

7.2.3 Image Gradient Difference Loss (GDL)

- To tackle problem 2 i.e. blurry predictions.
- Penalize the differences of image gradient predictions in the generative loss function.
  $\mathcal{L}_{gdl}(X, Y) = \sum_{i,j} ||X_{i,j} - X_{i-1,j}| - |Y_{i,j} - Y_{i-1,j}||^\alpha + ||X_{i,j-1} - X_{i,j}| - |Y_{i,j-1} - Y_{i,j}||^\alpha$
  where $\alpha \geq 1$

7.2.4 Combining Losses

$$\mathcal{L}(X, Y) = \lambda_{adv}\mathcal{L}^G_{adv}(X, Y) + \lambda_{l_p}\mathcal{L}_p(X, Y) + \lambda_{gdl}\mathcal{L}_{gdl}(X, Y)$$

This is used as a loss function for the Generator Network

7.3 Experiments

- Two models:
  1. 4 context frames, 1 predicted frame.
  2. 8 context frames, predicted frames.

7.3.1 Datasets

- Trained on Sports1M, tested on UCF-101.

7.3.2 Network Architecture

7.3.3 Quantitative Evaluations

- PSNR, SSIM and sharpness are calculated between true frame Y and predicted frame $\hat{Y}$.
- Evaluation of the accuracy of future frames prediction only takes the moving areas of the images into account.
- To extract moving areas, EpicFlow method is used.
  http://vision.middlebury.edu/flow/code/flow-code-matlab.zip
- Different quality measures are computed in the areas only in the areas where the optical flow is higher than a fixed threshold.
- Interesting fact: Training on l_2 gives worst PSNR.

7.3.4 Comparison to Ranzato et al.

7.4 Conclusion

8 Learning Visual Predictive Models of Physics for Playing Billiards

Paper Details

- Paper: arXiv
- Authors: Pulkit Agrawal, Sergey Levine
- Institutions: UC Berkeley
- Project website:
- Code:
- Published in: ICLR 2016
- Citations: 156 (As of 24-06-2020)

Abstract

8.1 Introduction

- Visual predictive models of physics: models that can enable the agents to visually anticipate the future states of the world.
- Motivation:
  - A visual predictive model of physics equips an agent with the ability to generate potential future states of the world in response to an action without actually performing that action.
  - By running multiple internal simulations to imagine the effects of different actions, the agent can perform planning, choosing the action with the best outcome and executing it in the real world.
- Contributions:
  - Learning dynamical model of the external world directly from visual inputs.

8.2 Previous Work

8.3 Learning Predictive Visual Models

- Input to model is 4 past frames and the force exercised on an object. Output is the predicted velocity of the object
- Loss Functions: Weighted MSE between ground truth and predicted velocities of the object.
- How did they generate the training data?

8.4 Model Evaluation

- Datasets:
- Baselines:
- Evaluation Metrics: Error in the angle and magnitude of the predicted velocities.
- Error is high near collisions.

8.5 Generating Visual Imaginations

- Next frame is generating by translating each ball by its predicted velocity. The predicted frames are fed back for long term predictions.

8.6 Using Predictive Visual Models for Action Planning

- Task: Compute force to move a ball to desired location. Force is computing by sampling different values (CMA-ES method) and checking its output using the internal simulator. The optimal force is the one that produces the world state that is closest to the target state. Not clear if error is computed in pixel space.

8.7 My Summary

-

9 Dynamic Filter Networks

Paper Details

- Authors: Bert De Brabandere
- Institutions: iMinds
- Project website:
- Code: https://github.com/dbbert/dfn
- Published in: NIPS 2016
- Citations: 230 (As of 20-10-2019)

Abstract

- In CNNs, after training, filters are fixed. In DFN, filters are generated dynamically conditioned on the input.

9.1 Introduction

- Motivation:
- Contributions:
  - –
- Dynamic Filter Module consists of 2 parts: Filter generating network and dynamic filtering layer.

9.2 Related Work

9.3 Dynamic Filter Networks

- Loss Functions:

9.4 Experiments

- Datasets: Moving MNIST, Highway Driving Dataset
- Baselines: Some ablation
- Evaluation Metrics:

10 Improved Techniques for Training GANs

Abstract

- Semi-supervised classification on MNIST, CIFAR-10, SVHN datasets.

10.1 Introduction

- Training GANs requires finding a Nash equilibrium of a non-convex game with continuous, high-dimensional parameters.
- GANs are typically trained using gradient descent techniques that are designed to find a low value of a cost function, rather than to find the Nash equilibrium of a game. When used to seek for a Nash equilibrium, these algorithms may fail to converge.
- Code: https://github.com/openai/improved-gan

10.2 Related Work

10.3 Toward Convergent GAN Training

- A Nash equilibrium is a point such that the Discriminator cost function is at a minimum w.r.t Discriminator parameters and Generator cost function is at a minimum w.r.t Generator parameters.
- Modification (update) to Discriminator parameters to reduce Discriminator loss function can increase Generator loss function and vice-versa. Thus, methods like Gradient Descent may fail to converge (achieve Nash equilibrium) for many games.
- Eg: One player minimizing xy w.r.t. x and the other player minimizing $-xy$ w.r.t y. Nash equilibrium in this case is $x = y = 0$, but Gradient Descent fails to achieve it.

10.3.1 Feature matching

- Genertor objective is changed from '*maximizing the output of discriminator*' to '*matching the statistics of real data* i.e. match the expected value of features on an intermediate layer of discriminator'.
- Let $f(x)$ denote the activations on an intermediate layer of the discriminator. New objective for generator is to minimize
$$\|\mathbb{E}_{x\sim p_{\text{data}}}[f(x)] - \mathbb{E}_{z\sim p_z(z)}[f(G(z))]\|_2^2$$
- This prevents a generator from overtraining on the current discriminator. Thus addresses the instability of GANs.
- No theoretical guarantee, but empirical results support this.

10.3.2 Minibatch discrimination

- Discriminator uses other examples in mini-batch while classifying an input as real or fake.
- Shown to prevent Mode-collapse.

10.3.3 Historical Averaging

10.3.4 One-sided label smoothing

- Label smoothing: Replace the '0 or 1' targets for the classifier with smoothed values '0.1 or 0.9'.
- Replacing positive classification targets with α and negative targets with β, the optimal discriminator becomes $D(x) = \frac{\alpha p_{\text{data}}(x) + \beta p_{\text{model}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)}$
- We smooth only positive labels to α and set $\beta = 0$

10.3.5 Virtual batch normalization

- Each example x is normalized based on the statistics collected on a reference batch of examples that are chosen once and fixed at the start of training, and on x itself.

10.4 Assessment of image quality

- Inception Score:
  - Apply the Inception model to every generated image to get the conditional label distribution $p(y|x)$. Images that contain meaningful objects should have a conditional label distribution p(yjx) with low entropy.
  - Then find marginal $p(y) = \int p(y|x = G(z))dz$. To have diversity in generated images, $p(y)$ should have high entropy.
  - Combining these two requirements, Inception Score = $\exp(\mathbb{E}_x[\mathrm{KL}(p(y|x)||p(y))])$. The values are exponentiated for easier comparision.
  - Inception Score is not a good objective for training, but a good evaluation metric.
  - Number of samples should be large enough $\sim$ 50K
  - Inception model used: http://download.tensorflow.org/models/image/imagenet/inception-2015-12-0 tgz

10.5 Semi-supervised learning

10.6 Experiments

10.6.1 MNIST

10.6.2 CIFAR-10

10.6.3 SVHN

10.6.4 ImageNet

10.7 Conclusion

11 Learning to Poke by Poking: Experiential Learning of Intuitive Physics (NIPS 2016)

Abstract

11.1 Introduction

- Predicting the value of every pixel in the next image is a challenging task.
- Moreover, in most cases it is not the precise pixel values that are of interest, but the occurrence of a more abstract event.
- For example, predicting that a glass jar will break when pushed from the table onto the ground is of greater interest (and easier) than predicting exactly how every piece of shattered glass will look.
- A forward model predicts the next state from the current state and action, and an inverse model predicts the action given the initial and target state.
- In joint training, the inverse model objective provides supervision for transforming image pixels into an abstract feature space, which the forward model can then predict.
- Baxter Robot is used to poke at objects.
- 400 hours of training, 100K pokes, 16 distinct objects.

11.2 Data

- The robot is equipped with a Kinect camera and a gripper for poking objects kept on a table in front of it.
- Only 3 objects at any given time (during training).
- Coordinate system:
  - X axis: Horizontal axis of robot.
  - Y axis: Vertical axis of robot.
  - Z axis: Pointed away from the robot.
- Robot moves its finger along XZ plane at fixed height from the table.
- Point-cloud data from Kinect depth camera is used to prevent random poking (where object is not present). Only during training. While testing, RGB data is enough.
- Parameters in a poke:
  - Point to poke p (random)
  - Poke direction θ (random)
  - Length l
  - Action $u_t = (p, \theta, l)$
  - p_1: $\frac{l}{2}$ distance from p along direction θ
  - p_2: $\frac{l}{2}$ distance from p along direction $(\theta + 180^o)$

  Robot executes poke by moving its finger from p_1 to p_2
- Sometimes objects move in unexpected ways.

11.3 Method

- Notation:
  - I_t: Image at time t
  - u_t: Action at time t
  - x_t: Feature (state representation) of image at time t
  - W_{fwd}, W_{inv}: Forward and Inverse model parameters
  - F, G: Forward and Inverse models
- Prediction in feature space is less challenging than prediction in pixel space.
- Feature extracted by a Image Classification Net may not be useful for object manipulation.
- Inverse problem formulation prevents degenerate solution of all features reducing to zeros. Didn't understand how.
- Didn't understand second challenge in forward models.

11.3.1 Model

- Each training sample consists of (I_t, I_{t+1}, u_t)

11.3.2 Evaluation Procedure

- Greedy approach is followed for repeated iterative pokes. Limitation: Can't push around obstacles.
- Error Metrics: Location error and Pose error.

11.3.3 Blob Model

11.4 Results

11.5 Related Work

11.6 Discussion and Future Work

11.7 My Summary

11.8 My Inferences

- Moves to correct location, but not to correct pose.

12 Unsupervised Learning for Physical Interaction through Video Prediction

Paper Details

- Authors: Chelsea Finn, Ian Goodfellow, Sergey Levine
- Institutions: UC Berkeley, OpenAI, Google Brain
- Project website: https://sites.google.com/site/robotprediction/
- Code: https://github.com/tensorflow/models/tree/master/research/video_prediction
- Published in: NIPS 2016
- Citations: 400 (As of 19-10-2019)
- Pre-trained Models: Not available. Even authors don't have.

Abstract

- Learning about physical objects' motion by predicting frames (i.e. in pixel space), enables to generalize previously unseen objects.

12.1 Introduction

- Motivation: Predicting the effect of physical interactions is a critical challenge for learning agents acting in the world, such as robots, autonomous cars, and drones.
- Contributions:
  - Making long-range predictions in real-world videos by predicting pixel motion.
- Three motion prediction modules:
  1. DNA: Outputs a distribution over locations in the previous frame for each pixel in the new frame. The predicted pixel value is computed as an expectation under this distribution.
  2. CDNA: Outputs the parameters of multiple normalized convolution kernels to apply to the previous image to compute new pixel values. The idea is that pixels on the same rigid object will move together, and therefore can share the same transformation.
  3. STP: Outputs the parameters of multiple affine transformations to apply to the previous image.
- Dataset of 59,000 robot pushing motions, consisting of 1.5 million frames and the corresponding actions at each time step.

12.2 Related Work

12.3 Motion-Focused Predictive Models

- Model computes the next frame by first predicting the motions of image segments, then merges these predictions via masking.

12.3.1 Pixel Transformations for Future Video Prediction

12.3.2 Composing Object Motion Predictions

- CDNA and STP produce multiple object motion predictions, which need to be combined into a single image.
- For each model, including DNA, we also include a "background mask" where we allow the models to copy pixels directly from the previous frame. Besides improving performance, this also produces interpretable background masks

12.3.3 Action-conditioned Convolutional LSTMs

12.4 Robotic Pushing Dataset

- Dataset: 10robotic arms; 57000 interaction sequences; 1.5 million video frames; 2 test sets - each with 1250 recorded motions; First test set contains 2 subsets of objects seen during training; Second test set contains 2 subsets of objects unseen during training.
https://sites.google.com/site/brainrobotdata/home/push-dataset

12.5 Experiments

- Datasets: Robotic Pushing Dataset and Human Actions 3.6M dataset.
- Baselines: Copying the previous frame, Beyond MSE, Action-conditional video prediction in atari games.
- Evaluation Metrics: PSNR and SSIM

12.5.1 Action-conditioned prediction for robotic pushing

- Action conditioned prediction
- 2 context frames, 8 prediction frames. While testing, 18 prediction frames.

12.5.2 Human motion prediction

- Videos are subsampled to 10fps.
- 10 context frames, 10 prediction frames. While testing, 20 prediction frames.

12.6 Conclusion and Future Directions

13 Visual Dynamics: Probabilistic Future Frame Synthesis via Cross Convolutional Networks

Paper Details

- Authors: Tianfan Xue, Jiajun Wu, William T Freeman
- Institutions: Massachusetts Institute of Technology, Google Research
- Project website: http://visualdynamics.csail.mit.edu/
- Code: https://github.com/tfxue/visual-dynamics
- Published in: NIPS 2016
- Citations: 227 (As of 20-10-2019)

Abstract

- In contrast to traditional methods, which have tackled this problem in a deterministic or non-parametric way, we propose a novel approach that models future frames in a probabilistic manner.

13.1 Introduction

- Motivation:
- Contributions:
  - Conditional Variational Autoencoder
  - Intrinsic representation of difference image (Eulerian motion)
  - Motion modelling using a set of image-dependent convolution kernels operating over an image pyramid.
- In this work, we study the problem of visual dynamics: modeling the conditional distribution of future frames given an observed image.

13.2 Related Work

13.3 Formulation

- Loss Functions: MLE, Variational upper bound

13.4 Method

13.5 Evaluations

- Datasets: Movement of 2D Shapes, Movement of Video Game Sprites, PENN
- Baselines:
- Evaluation Metrics: MSE

14 Generating Videos with Scene Dynamics

Abstract

- Generates tiny videos up to a second at full frame rate.

14.1 Introduction

- Interested in the fundamental problem of learning how scenes transform with time.

15 Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (DC GAN)

15.1 Introduction

- One way to build good image representations is by training GANs
- Later, parts of Generator and Discriminator can be reused as feature extractors for supervised tasks.
- GANs provide an alternative to Maximum Likelihood Techniques.
- GANs have been known to be unstable to train, often resulting in generators that produce nonsensical outputs.

15.2 Related Work

15.3 Approach and Model Architecture

Proposed changes to CNN architecture:

1. Use all Convolutional Net. Replace deterministic spatial pooling functions with strided convolutions. It allows network to learn its own spatial downsampling.
   - For generator, replace pooling layers by fractional-strided convolutional layers.
   - For discriminator, replace pooling layers by strided convolutional layers.
2. Eliminate fully connected layers on top of convolutional features. Global average pooling increases model stability, but slows down convergence. Use a middle ground between them.
   - For generator, connect the noise vector input (noise distribution is uniform) is fully connected to the next layer. The result is then reshaped into a 4-dimensional tensor.
   - For discriminator, the last convolutional layer is flattened and then fed into a single sigmoid output.
3. Use Batch Normalization.
   - Stabilizes learning by normalizing the input to each unit to have zero mean and unit variance.
   - This prevents generator from collapsing all samples to a single point
   - Directly applying batchnorm to all layers resulted in sample oscillation and model instability.
   - This was avoided by not applying batchnorm to the generator output layer and the discriminator input layer.
4. For generator, use ReLU activation except for output layer. Use Tanh activation for output layer. For discriminator, use Leaky ReLU activation (Original GAN uses maxout activation).

15.4 Details of Adversarial Training

- Trained on 3 datasets:
  1. LSUN: Large-scale Scene Understanding
  2. Imagenet-1k
  3. Faces dataset
- Preprocessing: Images are scaled to [-1,1] i.e. range of tanh activation
- Optimizer: SGD with mini-batch-size=128
- Weights Initialization $\sim \mathcal{N}(0, 0.02)$
- LeakyReLU slope = 0.2 in all models
- Adam Optimizer: learning rate=0.0002; momentum term $\beta_1 = 0.5$

16 Flexible Spatio-Temporal Networks for Video Prediction

Paper Details

- Authors: Chaochao Lu
- Institutions: University of Cambridge, Max Planck Institute for Intelligent Systems
- Project website:
- Code:
- Published in: CVPR 2017
- Citations: 33 (As of 22-10-2019)

Abstract

16.1 Introduction

- Motivation:
- Contributions:
  - Versatile and flexible framework for video extrapolation and interpolation.
  - Novel objective function.
  - Different optimization strategies.
  - Comprehensive comparison of recent state-of-the-art video prediction methods.

16.2 Related Work

16.3 Model Description

- Loss Functions: l_2 loss, Huber loss, DeePSiM loss, adversarial loss

16.4 Training

16.5 Experiments

- Datasets: UCF-101, Sports-1M, PROST, ViSOR, Moving MNIST
- Baselines: Spatio-Temporal Video Autoencoder, Beyond MSE, Video (language) modeling baseline
- Evaluation Metrics: PSNR, sharpness

17 Generating the Future with Adversarial Transformers

Paper Details

- Authors: Carl Vondrick, Antonio Torralba
- Institutions: MIT
- Project website:
- Code:
- Published in: CVPR 2017
- Citations: 97 (As of 12-04-2020)

Abstract

17.1 Introduction

- Motivation:
- Contributions:
  - –

17.2 Related Work

17.3 Dataset

- Motion stabilized flickr videos

17.4 Method

- Loss Functions:
- 4 past frames, 12 future frames.

17.5 Experiments

- Resolution: 64×64
- Datasets:
- Baselines:
- Evaluation Metrics: 2AFC experiment

18 Self-Supervised Visual Planning with Temporal Skip Connections

Paper Details

- Authors: Frederik Ebert, Chelsea Finn, Alex X Lee, and Sergey Levine
- Institutions: UC Berkeley, Technical University of Munich
- Project website: https://sites.google.com/view/sna-visual-mpc
- Code:
- Published in: CoRL 2017
- Citations: 60 (As of 20-10-2019)

Abstract

18.1 Introduction

- Motivation: Robotics
- Contributions:
  - Video Prediction with object occlusions
  - Long-term planning.
  - Planning with both discrete and continuous actions with video prediction models.
- Suppose in a video, if a moving ball suddenly disappears (when in centre of frame, not edges), then the video doesn't look natural. But if there is an occlusion, then it is possible in natural videos as well.

18.2 Related Work

18.3 Preliminaries

18.4 Skip Connection Neural Advection Model

18.5 Visual MPC with Pixel Distance Costs

18.6 Sampling-Based MPC with Continuous and Discrete Actions

18.7 Experiments

- Datasets: BAIR
- Baselines: Dynamic Neural Advection (DNA)
- Evaluation Metrics:

19 Dual Motion GAN for Future-Flow Embedded Video Prediction

Paper Details

- Authors: Xiaodan Liang
- Institutions: Carnegie Mellon University, Petuum Inc.
- Project website:
- Code:
- Published in: ICCV 2017
- Citations: 93 (As of 21-10-2019)

Abstract

19.1 Introduction

- Motivation: Unsupervised video representation learning.
- Contributions:
  - –
- Dual Motion GAN learns by explicitly enforcing future-frame predictions to be consistent with the pixel-wise flows in the video through a dual learning mechanism.
- The primal future-frame prediction and dual future-flow prediction form a closed loop, generating informative feedback signals to each other for better video prediction.

19.2 Related Work

19.3 Dual Motion GAN

- Loss Functions: VAE loss, Adversarial loss

19.4 Experiments

- Datasets: KITTI, Caltech Pedestrian, UCF-101, THUMOS-15
- Baselines: Copying previous frame, Beyond MSE, PredNet, DVF, EpicFlow, NextFlow
- Evaluation Metrics: MSE, PSNR, SSIM

20 Least Squares Generative Adversarial Networks (LS GAN) - ICCV 2017

Abstract

- Sigmoid Cross-Entropy loss function causes vanishing gradient problem.
- LS GAN uses $\mathcal{L}_2$ distance as loss function.
- This is equivalent to minimizing Pearson Divergence
- LS GANs generate better quality images compared to normal GANs
- LS GANs are more stable during training.

20.1 Introduction

20.2 Related Work

20.3 Method

20.3.1 Generative Adversarial Networks

20.3.2 Least Squares Generative Adversarial Networks

The objective functions for LSGANs is defined as follows:

$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{x\sim p_{\text{data}}(x)}\left[(D(x)-b)^2\right] + \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[(D(G(z))-a)^2\right]$$
$$\min_D V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}\left[(D(G(z))-c)^2\right]$$

- a: Label for fake data
- b: Label for real data
- c: What Generator wants Discriminator to output for fake data.

20.3.2.1 Benefits of LSGANs

20.3.2.2 Relation to Pearson Divergence

20.3.2.3 Parameters Selection

- a, b, c need to satisfy the following equations: $b - c = 1$ and $b - a = 2$
- Set $a = -1, b = 1, c = 0$
- Another method: Set $c = b$ i.e. $a = 0, b = 1, c = 1$. Results in the paper use this.

20.4 Experiments

20.4.1 Datasets and Implementation Details

- Datasets:
  - LSUN
  - CIFAR-10
  - HWDB1.0
- Adam Optimizer: $\beta_1 = 0.5$
- Learning Rate:

  - LSUN: 0.001
  - CIFAR-10: 0.0002
  - HWDB1.0: 0.0002
- Code: https://github.com/xudonmao/LSGAN

20.4.2 Qualitative Evaluation

20.4.3 Quantitative Evaluation

20.4.4 Stability Comparison

20.4.4.1 Suggestions in Practice

- Sometimes LSGAN suffers from mode collapse at the end of training.
- Quality of generated images shift between good and bad during training process.
- Based on the above two observations, it is suggested to keep a record of generated images at every thousand or hundred iterations and select the model manually by checking the image quality.

20.4.5 Handwritten Chinese Characters

20.5 Conclusions and Future Work

21 Video Frame Synthesis using Deep Voxel Flow (DVF)

Paper Details

- Authors: Ziwei Liu
- Institutions: The Chinese University of Hong Kong, Google
- Project website: https://liuziwei7.github.io/projects/VoxelFlow
- Code: https://github.com/liuziwei7/voxel-flow, https://github.com/lxx1991/pytorch-voxel-flow
- Published in: ICCV 2017
- Citations: 169 (As of 20-10-2019)

Abstract

21.1 Introduction

- Motivation: Film production (Interpolation)
- Contributions:
  - –

21.2 Related Work

21.3 Our Approach

- Loss Functions:

21.4 Experiments

- Datasets: UCF-101, THUMOS-15, KITTI
- Baselines: EpicFlow (Interpolation), Beyond MSE (Extrapolation)
- Evaluation Metrics: PSNR, SSIM

22 Decomposing Motion and Content for Natural Video Sequence Prediction

Paper Details

- Authors: Ruben Villegas
- Institutions: University of Michigan, Adobe Research (San Jose), Google Brain (Mountain View)
- Project website: https://sites.google.com/a/umich.edu/rubenevillegas/iclr2017
- Code: https://github.com/rubenvillegas/iclr2017mcnet
- 2017-ICLR conference paper.
- Citations: 142 (As of 10-Aug-2019)

Abstract

- Decompose motion and content.
- Model: Encoder-Decoder ConvNet and ConvLSTM.
- Datasets: KTH, Weizmann, UCF-101, Sports-1M.

22.1 Introduction

- Motivation: Existing video recognition models can be adopted on top of the predicted frames to infer various semantics of the future.
- Predicting high level semantics like action, motion are often specific to the particular task. They provide only a partial description of the future. Additionally, they require labelled data for training.
- Contributions: MCnet, decomposing video into motion and content without separate training (for video prediction task).

22.2 Related Work

22.3 Algorithm Overview

- Objective: Given $x_{1:t}$, predict $\hat{x}_{t+1}$
- Motion Encoder: Takes difference of consecutive frames and produces hidden representation d_t that encodes temporal dynamics of the scene components.
- Content Encoder: Takes last observed frame x_t and outputs hidden representation s_t that encodes spatial layout of the scene.
- Multi-Scale Motion-Content Residual: Takes features from both encoders at every scale before pooling and computes residuals r_t
- Combination Layers and Decoder: Takes d_t, s_t, r_t and produces $\hat{x}_{t+1}$
- Multiple frames can be predicted by recursively feeding the predicted frames as input.

22.4 Architecture

22.4.1 Motion Encoder

- $[d_t, c_t] = f^{\text{dyn}}(x_t - x_{t-1}, d_{t-1}, c_{t-1})$
- c_t is a memory cell that retains information of the dynamics observed through time.
- f^{dyn} is a fully conv net. Encoder CNN with a Convolutional LSTM layer on top.

22.4.2 Content Encoder

- Extracts important spatial features from a single frame.
- $s_t = f^{\text{cont}}(x_t)$
- f^{cont} is a CNN.
- Assymetric architecture for motion and content encoder.

22.4.3 Multi-Scale Motion-Content Residual

- The residual feature at layer l is computed by $r_t^l = f^{\text{res}}([s_t^l, d_t^l])^l$
- $[\cdot, \cdot]$ represents concatenation along depth dimension.
- $f^{\text{res}}(\cdot)^l$ is implemented as consecutive convolution layers and rectification with a final linear layer.

22.4.4 Combination Layers and Decoder

- First combines the motion and content back into a unified representation by $f_t = g^{\text{comb}}([d_t, d_t])$
- g^{comb} is implemented by a CNN with bottleneck layers.
- $x_{t+1} = g^{\text{dec}}(f_t, r_t)$
- g^{dec} is a deconvolution network. The output layer is passed through a tanh activation function.

22.5 Inference and Training

22.5.1 Multi-Step Prediction

22.5.2 Training Objective

- Loss:

$$
\begin{aligned}
\mathcal{L} &= \alpha \mathcal{L}_{\text{img}} + \beta \mathcal{L}_{\text{GAN}} \\
\mathcal{L}_{\text{img}} &= \mathcal{L}_p(x_{t+k}, \hat{x}_{t+k}) + \mathcal{L}_{gdl}(x_{t+k}, \hat{x}_{t+k}) \\
\mathcal{L}_p(y, z) &= \sum_{k=1}^{T} \|y - z\|_p^p \\
\mathcal{L}_{gdl}(y, z) &= \sum_{i,j}^{h,w} |(|y_{i,j} - y_{i-1,j}| - |z_{i,j} - z_{i-1,j}|)|^\lambda + |(|y_{i,j-1} - y_{i,j}| - |z_{i,j-1} - z_{i,j}|)|^\lambda \\
\mathcal{L}_{\text{GAN}} &= -\log D([x_{1:t}, G(x_{1:t})]) \\
G(x_{1:t}) &= \hat{x}_{t+1:t+T} \\
\mathcal{L}_{\text{disc}} &= -\log D([x_{1:t}, x_{t+1:t+T}]) - \log(1 - D([x_{1:t}, G(x_{1:t})]))
\end{aligned}
$$

22.6 Experiments

- Evaluated on datasets: KTH, Weizmann action, UCF-101
- Baselines: ConvLSTM, Deep Multi-Scale Video Prediction Beyond Mean Square Error (Mathieu, Yann LeCun)
- Hyper-parameters: $\alpha = 1, \lambda = 1, p = 2$

22.6.1 KTH and Weizmann Action Datasets

- KTH split: train(1-16), test(17-25).
- Frames resized to 128x128
- Hyper-parameters: $\beta = 0.02$ for training.
- Evaluation: PSNR, SSIM

22.6.2 UCF-101 Dataset

- Resized frames to 240x320.
- Observes 4 frames and predicts next frame.
- Hyper-parameters: $\beta = 0.001$ for training.
- Trained on Sports-1M and tested on UCF-101.

22.7 Conclusion

23 Deep Predictive Coding Networks for Video Prediction and Unsupervised Learning

Abstract

23.1 Introduction

- Computer vision models are typically trained using static images. But in real visual world, there is movement in both viewer and objects.
- Many have suggested that temporal experience with objects as they move and undergo transformations can serve as an important signal for learning about the structure of objects.
- PredNet: A deep, recurrent convolutional neural network to continually predict the appearance of future video frames
- Inspired from the concept of "Predictive Coding" in Neuroscience.Predictive coding posits that the brain is continually making predictions of incoming sensory stimuli.

23.2 The PredNet Model

23.3 Experiments

23.3.1 Rendered Image Sequences

- Quantitative evaluation of generative models is a difficult, unsolved problem. Here, MSE and SSIM are used.
- The rotating faces were generated using the FaceGen software package.
- To understand the information contained in the trained models, we decoded the latent parameters from the representation neurons (R_l) in different layers, using a ridge regression.

23.3.2 Natural Image Sequences

- Car-mounted camera videos are chosen since these videos span across a wide range of settings and are characterized by rich temporal dynamics, including both self-motion of the vehicle and the motion of other objects in the scene.
- Models were trained using the raw videos from the KITTI dataset. Tested on the CalTech Pedestrian dataset.
- Training dataset: 128x160 resolution 41K frames.
- Though trained for 1 frame prediction, can be used to predict upto 9 frames by feeding back the predicted frames. Approx 2fps. So, given a frame, 5s video is generated.

23.4 Discussion

23.5 Appendix

24 Learning to Generate Long-term Future via Hierarchical Prediction

Paper Details

- Authors: Ruben Villegas
- Institutions: Google Brain, University of Michigan, Adobe Research
- Project website: https://sites.google.com/a/umich.edu/rubenevillegas/hierch_vid
- Code: https://github.com/rubenvillegas/icml2017hierchvid
- Published in: ICML 2017
- Citations: 130 (As of 22-10-2019)

Abstract

- To avoid inherent compounding errors in recursive pixel level prediction, we propose to first estimate high level structure in the input frames, then predict how that structure evolves in the future, and finally by observing a single frame from the past and the predicted high-level structure, we construct the future frames without having to observe any of the pixel-level predictions.

24.1 Introduction

- Motivation: Robotics, Autonomous cars.
- Contributions:
  - Hierarchical approach for video prediction

24.2 Related Work

24.3 Overview

24.4 Architecture

24.5 Training

- Loss Functions: MSE, MSE in feature space (AlexNet and Hourglass Network), adversarial loss, MSE between poses.

24.6 Experiments

- Datasets: Human 3.6M, Penn Action (10 context, 32 predictions, 64 predictions for testing)
- Baselines:
- Evaluation Metrics: 2AFC (using AMT), PSNR

25 Video Pixel Network (VPN)

Paper Details

- Authors: Nal Kalchbrenner
- Institutions: Google Deep Mind
- Project website:
- Code:
- Published in: ICML 2017
- Citations: 177 (As of 20-10-2019)

Abstract

- VPN estimates discrete joint distribution of the raw pixel values in a video.

25.1 Introduction

- Motivation:
- Contributions:
  - –

25.2 Method

25.3 Architecture

25.4 Network Building Blocks

Experiments

- Datasets: Moving MNIST, PUSH
- Baselines:
- Evaluation Metrics:

26 Learning to See Physics via Visual De-animation

Paper Details

- Paper: NIPS
- Authors: Jiajin Wu, Pushmeet Kohli, William Freeman, Joshua Tenenbaum
- Institutions: MIT, Deep Mind
- Project website:
- Code:
- Published in: NIPS 2017
- Citations: 100 (As of 25-06-2020)

Abstract

26.1 Introduction

- Motivation: Humans can predict what happens next, infer physical properties from visual input and make an unstable system, stable by applying forces.
- Contributions:
  - Generative pipeline for physical scene understanding.
  - Learning of physical scene representations without human supervision.

26.2 Related Work

26.3 Visual De-animation

- Components:
  - Perception Module: Does inverse physics i.e. generates object representations and estimates their physical state.
  - Physics Engine: Can be neural network based differentiable engine or classical non-differentiable engines. REINFORCE is used for non-differentiable engines.
  - Graphics Engine: Renders pixel images from object representations.

26.4 Evaluation

- Framework is evaluated in 3 scenarios.
  - Synthetic Billard Tables
  - Real world Billiards videos
  - Block Towers

26.4.1 Billiard Tables: A Motivating Example

- 3 types:
  - same appearance, same physics
  - different appearance, different physics
  - same appearance, different physics
- Physical Properties:
  - Intrinsic properties:
    - Mass

    - Friction
  - Extrinsic Properties:
    - 2D position
    - Velocity

- Flow fields are computed using SPyNet
- Perceptual model is a ResNet-18 which takes masked 3 RGB frames and 2 flow images and recovers object's physical state.
- A differential neural physics engine predicts objects' extrinsic properties in the next frame.
- Graphics engine renders final images.
- Loss functions:
  - MSE in pixel space for reconstruction.
  - From rendered images, objects positions are computed and error in positions is used.
- Training:
  - Perceptual module and neural physics engine are pretrained on synthetic data (supervised).
  - End-to-end fine tuning without annotations.
- Quantitative measures:
  - MSE in pixel space for reconstruction
  - Manhattan distance for position and velocity prediction
  - Behavioral study: Subjects are asked to predict future trajectory. Ratio of errors in trajectories predicted by model and humans is computed.

26.4.2 Billiard Tables: Transferring to Real Videos

- Same as with synthetic billiards, but perception module takes flow images to abstract away appearance changes.
- Perception module is retrained.
- Neural physics engine trained on synthetic billiards is used.
- Predicted frames are rendered using Blender by providing the object states
- No quantitative evaluation.

26.4.3 The Blocks World

- Focus is on reasoning of object states in 3D world rather than physical properties like mass.
- All objects have same physical properties (mass, friction etc), just differ in color.
- Physical state is the pose (3D position and 3D rotation euler angles)
- Perception module is ResNet-18 which takes silhouettes of blocks and recovers physical state.
- Bullet physics engine is used.
- Blender is used to render predicted frames.
- Input is a static image.
- Loss functions:
  - MSE between rendered silhouettes and observations.
  - Binary Cross-Entropy between predicted and ground truth stability.

- Training:
  - Perception module is trained on synthetic data (supervised).
  - End-to-end finetuning with only ground truth values for stability.
- 3 tasks:
  - Scene reconstruction
  - Stability prediction (Future prediction)
  - Making an unstable tower stable
- Evaluation: Accuracy of stability prediction.
- Generalization: trained on towers with 2/4 blocks and tested on towers with 3 blocks.

26.5 Discussion

26.6 My Summary

- Main contribution is the framework integrating perception module, physics engine and rendering engine.
- Can be used for either estimating physical properties or for video prediction.
- Drawbacks:
  - Doesn't evaluate the perception module on ground truth for synthetic data.
  - May not generalize to unseen objects.
  - There is a scale factor ambiguity in mass and friction coefficient estimated for billiards case - it appears impossible to find out the exact mass and friction coefficient in the current setup.
  - However, if an external force is applied using a robotic arm or using gravity, then mass can be accurately estimated.
  - Not clear what use is estimating mass and friction coefficient which is correct only upto a scale factor.
  - Doesn't fine tune the end-to-end model for real world Billiards case.
  - MSE may not be an appropriate loss function in fine-tuning for real world Billiards. SSIM structure term might be better.
  - Stability prediction is not an exhaustive evaluation measure for perception task. The perception module might have estimated the pose wrongly and still be able to predict the stability correctly. The binary score sweeps a lot of factors under the carpet.
  - Since improvement in accuracy is not significant, it raises the question of effectiveness of this framework.

27 Transformation-based Models of Video Sequences

Paper Details

- Authors: Marc' Aurelio Ranzato, Soumith Chintala
- Institutions: Facebook AI Research
- Project website:
- Code:
- Published in: ICLR 2017 Reject
- Citations: 32 (As of 12-04-2020)

Abstract

27.1 Introduction

- Motivation:
- Contributions:
  - –
- Transformation based approach: uses affine transformations.
- New evaluation method

27.1.1 Related Work

27.2 Model

- Loss Functions:
- 4 past frames, 8 predicted frames.

27.3 Experiments

- Datasets: Moving MNIST, UCF-101
- Baselines:
- Evaluation Metrics: Classification accuracy for action recognition.

27.4 Conclusions

28 Deep Image Prior

Paper Details

- Authors: Dmitry Ulyanov, Andrea Vedaldi, Victor Lempitsky
- Institutions: Skolkovo Institute of Science and Technology, University of Oxford
- Project website: https://dmitryulyanov.github.io/deep_image_prior
- Code: https://github.com/DmitryUlyanov/deep-image-prior
- Published in: CVPR 2018
- Citations: 353 (As of 07-01-2020)

Abstract

28.1 Introduction

28.2 Method

28.3 Applications

- Blind Image Denoising
- Super Resolution
- Inpainting
- Natural Pre Image
  - It is a diagnostic tool to study the invariances of a lossy function, such as deep network.
  - Let ϕ be the first several layers of a neural network trained to perform, say, image classification. The pre-image is the set of images, that result in same representation $\phi(x_0)$

$$\phi^{-1}(\phi(x_0)) = \{x \in \mathcal{X} : \phi(x) = \phi(x_0)\}$$

  - Pre-images can be found by minimizing

$$E(x; x_0) = \|\phi(x) - \phi(x_0)\|^2$$

  - Optimizing this function directly may lead to non-natural images. By restricting pre-image to set $\mathcal{X}$ of natural images, gives natural pre image.
- Flash - No flash reconstruction

28.4 Related Work

28.5 Discussion

28.6 My Summary

- Use a randomly initialized conditional generator network.
- Update network parameters (weights) to minimize loss function, for the given test image.
- Overtraining doesn't give good results, but optimal number of iterations at optimal learning rate, gives good images.

29 Future Frame Prediction for Anomaly Detection - A New Baseline

Paper Details

- Authors: Wen Liu, Weixin Luo, Dongze Lian, Shenghua Gao
- Institutions: ShanghaiTech University
- Project website:
- Code: https://github.com/StevenLiuWen/ano_pred_cvpr2018
- Published in: CVPR 2018
- Citations: 90 (As of 12-04-2020)

Abstract

29.1 Introduction

- Motivation:
  - Video Prediction can better capture anomalies than video reconstruction.
- Apply temporal constraint by minimizing error in optical flow (optical flow is not predicted).
- Contributions:
  -

29.2 Related Work

29.3 Method

- Unet based architecture
- Loss Functions:
  - MSE in pixel space
  - Gradient difference loss
  - Flow l_1 error
  - Adversarial loss
- 4 past frames, 1 future frame

29.4 Experiments

- Datasets: CUHK Avenue, UCSD Pedestrian, ShanghaiTech (Anomaly detection datasets)
- Baselines: BeyondMSE
- Evaluation Metrics: Area under the curve

30 Structure Preserving Video Prediction

Paper Details

- Authors: Jingwei Xu
- Institutions: Shanghai Institute for Advanced Communication and Data Science
- Project website:
- Code:
- Published in: CVPR 2018
- Citations: 18 (As of 09-04-2020)

Abstract

30.1 Introduction

- Motivation:
- Contributions:
  - –

30.2 Related Work

30.3 Methods

- Loss Functions:

30.4 Experiments

- Datasets: UCF-101, Human 3.6M, CityScapes
- Baselines:
- Evaluation Metrics: PSNR, SSIM
- 10 past frames, 10 future frames.

31 The Unreasonable Effectiveness of Deep Features as a Perceptual Metric

Abstract

- Features extracted by VGG network (pretrained on ImageNet-1k dataset) have been remarkably useful as a training loss for image synethesis. But how "perceptual" are these "perceptual losses"?
- A new dataset of human perceptual similarity judgements is introduced.
- Deep features (not restricted to ImageNet trained VGG, all levels of supervision i.e. supervised, self-supervised and unsupervised) perform better than PSNR and SSIM in perceptual quality.
- Models and data: https://www.github.com/richzhang/PerceptualSimilarity

31.1 Motivation

- What we would really like is a "perceptual distance", which measures how similar are two images in a way that coincides with human judgment.
- Human judgments of similarity
  - depend on high-order image structure
  - are context-dependent
  - may not actually constitute a distance metric
- Deep features perform far better than SSIM and FSIM when there is spatial ambiguity
- Randomly initialized networks do not achieve good performance.
- Our study is based on a newly collected perceptual similarity dataset, using a large set of distortions and real algorithm outputs. It contains both traditional distortions, such as contrast and saturation adjustments, noise patterns, filtering, and spatial warping operations, and CNN-based algorithm outputs, such as autoencoding, denoising, and colorization, produced by a variety of architectures and losses.
- Contributions:
  - Dataset: **484k** human judgements.
  - Deep features model low level perceptual similarity
  - Network architecture alone is insufficient. Training is important.
  - Performance can be improved by "calibrating" feature responses from a pre-trained network using the dataset.

31.2 Berkeley-Adobe Perceptual Patch Similarity (BAPPS) Dataset

- We collect a large-scale highly diverse dataset of perceptual judgments using two approaches.
  - Mainly Two Alternative Forced Choice (**2AFC**).
  - Validated by Just Noticeable Difference (**JND**).

31.3 Deep Feature Spaces

-

32 ContextVP: Fully Context-Aware Video Prediction

Paper Details

- Authors: Wonmin Byeon
- Institutions: NVIDIA
- Project website: https://wonmin-byeon.github.io/publication/2018-eccv
- Code:
- Published in: ECCV 2018
- Citations: 23 (As of 21-10-2019)

Abstract

32.1 Introduction

- Motivation: Unsupervised learning of features.
- Contributions:
  - Highlighting a blind spot problem of uncertain future leading to blurry predictions.
  - Simple baseline.
  - New architecture for video prediction.

32.2 Related Work

32.3 Missing Contexts in Other Network Architectures

32.4 Method

- Loss Functions: $\mathcal{L}_p$ Loss, Image Gradient Difference Loss

32.5 Experiments

- Datasets:
  - Human 3.6M: 10 past frames.
  - KITTI (train) + CalTech Pedestrian (test): 10 past frames.
  - UCF101: 4 past frames used.
- Baselines: Copying previous frame, ConvLSTM20, Beyond MSE, PredNet, DVF, Dual Motion GAN
- Evaluation metrics: PSNR, SSIM

33 DYAN: A Dynamical Atoms-Based Network For Video Prediction

Paper Details

- Authors: Wenqian Liu, Abhishek Sharma
- Institutions:Northeastern University, Boston
- Project website:
- Code: https://github.com/liuem607/DYAN
- Published in: ECCV 2018
- Citations: 2 (As of 25-09-2019)

Abstract

33.1 Introduction

- Motivation: Autonomous Driving
- Contributions:
  - A novel auto-encoder network that captures long and short term temporal information and explicitly incorporates dynamics-based affine invariants;
  - The proposed network is shallow, with very few parameters. It is easy to train and it does not take large disk space to save the learned model.
  - The proposed network is easy to interpret and it is easy to visualize what it learns, since the parameters of the network have a clear physical meaning.
  - The proposed network can predict future frames accurately and efficiently without introducing blurriness.
  - The model is differentiable, so it can be fine-tuned for another task if necessary.
- RNNs are hard to train for long term predictions because of vanishing and exploding gradients.
- LSTMs and GRUs are easier to use.
- GANs are hard to reportedly train, since training requires finding a Nash equilibrium of a game, which might be hard to get using gradient descent techniques.
- DYAN is similar to LSTMs.

33.2 Related Work

- Most Optical Flow methods focus on Lagrangian Optical Flow: Flow field represents the displacement between corresponding pixels or features across frames.
- In Eulerian optical flow, where the motion is captured by the changes at individual pixels, without requiring finding correspondences or tracking features.
- Eulerian flow has been shown to be useful for tasks such as motion enhancement and video frame interpolation.

33.3 Background

- Loss Functions: Sparse optimization, l_2 loss.

33.4 DYAN: A dynamical atoms-based network

33.5 Implementation Details

33.6 Experiments

- Datasets: KITTI/Caltech and UCF-101.
- Baselines: CopyLastFrame, DualMoGAN, BeyondMSE, PredNet and DVF.
- Evaluation Metrics: MSE and SSIM.

33.6.1 Car Mounted Camera Videos Dataset

- Based on 10 past frames, predicts the next 1 frame.
- Videos center-cropped and resized to 128x160.
- Baselines: CopyLastFrame, DualMoGAN, BeyondMSE and PredNet.

33.6.2 Human Action Videos Dataset

- Based on 4 past frames, predicts the next 1 frame.
- Videos of resolution 320x240.
- Baselines: CopyLastFrame, BeyondMSE and DVF.

34 Folded Recurrent Neural Networks for Future Video Prediction (FRNN)

Paper Details

- Authors: Marc Oliu
- Institutions: Universitat Oberta de Catalunya, Barcelona, Spain
- Project website:
- Code: https://github.com/moliusimon/frnn
- Published in:
- Citations: 23 (As of 09-04-2020)

Abstract

34.1 Introduction

- Motivation:
  - Action and gesture recognition
  - Task planning
  - Weather prediction
  - Optical flow estimation
  - New view synthesis
- Contributions:
  -

34.2 Related Work

34.3 Proposed Method

- Loss Functions:

34.4 Experiments

- Datasets: Moving MNIST, UCF-101, KTH
- Baselines: RLadder, PredNet, Srivastava, BeyondMSE, MCnet
- Evaluation Metrics: MSE

35 Probabilistic Video Generation using Holistic Attribute Control (VideoVAE)

Paper Details

- Authors:
- Institutions: Disney Research, Pittsburgh, USA; California Institute of Technology, Pasadena, USA; Simon Fraser University, Burnaby, Canada; The University of British Columbia, Vancouver, Canada.
- Project website:
- Code:
- Published in: ECCV 2018
- Citations: 12 (As of 26-09-2019)

Abstract

35.1 Introduction

- Motivation: Video Generation model are useful for building spatio-temporal priors, forecasting , and unsupervised feature learning.
- Contributions:
  - Novel generative video model: VideoVAE
- A generative video model should have the following properties:
  - It should be able to model diversity of future predictions.
  - Each future prediction, which corresponds to a sample from the generative model, should be self-consistent.

35.2 Related Work

35.3 Probabilistic Video Generation

- Loss Functions: VAE, cross-entropy loss (for holistic attribute classifiers)
- VAE as spatial model and LSTM as temporal model.

35.4 Learning and Synthesis

35.5 Experiments

- Conditioned on first frame, next frames are predicted.

35.5.1 Datasets

- Chair CAD: Resolution 64x64.
- Weizmann Human Action
- YFCC - MIT Flickr

35.5.2 Evaluation Metrics

- Inception Score: Individual Classifiers are pretrained on each of the datasets.
- Intra entropy and inter entropy: Individual terms in Inception Score.

35.5.3 Video Synthesis

- Baselines: Ablation Models, Deep Rotator, VGAN, MoCoGAN.

36 SDC-Net: Video prediction using spatially-displaced convolution

Paper Details

- Authors: Fitsum A Reda
- Institutions: NVIDIA
- Project website: https://nv-adlr.github.io/publication/2018-SDCNet
- Code:
- Published in: ECCV 2018
- Citations: 17 (As of 21-10-2019)

Abstract

36.1 Introduction

- Motivation: Self driving vehicles, video analysis.
- Contributions:
  - Deep model for high-resolution frame prediction from a sequence of past frames.
  - Spatially Displaced Convolutional (SDC) module for effective frame synthesis via transformation learning.
  - Comparison of SDC module with kernel-based, vector-based and state-of-the-art approaches.
- spatially-displaced convolutional (SDC) module is used for video frame prediction. A motion vector and a kernel for each pixel are learnt and a pixel is synthesized by applying the kernel at a displaced location in a source image, defined by the predicted motion vector.

36.2 Method

- Loss Functions: $\mathcal{L}_1 loss$,Perceptual $\mathcal{L}_1$ loss, Style loss

36.3 Experiments

- Datasets: Caltech Pedestrian, YouTube-8M
- Baselines: Copying previous frame, Beyond MSE, PredNet, MCnet, DualGAN,
- Evaluation Metrics: $\mathcal{L}_1$ distance, MSE, PSNR, SSIM

37 Stochastic Variational Video Prediction (SV2P)

Paper Details

- Authors: Mohammad Babaeizadeh, Chelsea Finn, Sergey Levine
- Institutions: University of Illinois at Urbana-Champaign, University of California Berkeley, Google Brain.
- Project website: https://sites.google.com/site/stochasticvideoprediction/
- Code: https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/video/sv2p.py
- Published in: ICLR 2018
- Citations: 105 (As of 19-10-2019)

Abstract

37.1 Introduction

- Motivation: Representations acquired by a video prediction model can be used for a variety of visual perception tasks, such as object tracking and action recognition. Such models will also be useful to allow an autonomous agent or robot to decide how to interact with the world to bring about a desired outcome.
- Contributions:
  - Stochastic variational method for video prediction (SV2P)
  - Stable training procedure for training a neural network based implementation of SV2P.
- Modeling future distributions over images is a challenging task.
- Hence, it is common to make various simplifying assumptions.
- One particularly common assumption is that the environment is deterministic and that there is only one possible future.
- However real world prediction tasks are stochastic.
- So deterministic models predict a statistic (such as expected value) of all possible outcomes.

37.2 Related Work

37.3 Stochastic Variational Video Prediction (SV2P)

- Loss Functions: MLE, KL Divergence
- Predictions are conditioned on a set of c context frames $x_0, \ldots, x_{c-1}$
- Goal is to sample from $p(x_{c:T}|x_{0:c-1})$
- Prior: $z \sim p(z)$
- Model using the prior:

$$p(x_{c:T}|x_{0:c-1}, z) = \prod_{t=c}^{T} p_\theta(x_t|x_{0:t-1}, z)$$

- True posterior: $p(z|x_{0:T})$ - Intractable
- Approximate posterior with an inference network $q_\phi(z|x_{0:T})$. This network outputs parameters of a conditionally gaussian distribution $\mathcal{N}(\mu_\phi(x_{0:T}), \sigma_\phi(x_{0:T}))$
- Inference network is trained using reparameterization trick: $z = \mu_\phi(x_{0:T}) + \sigma_\phi(x_{0:T}) \cdot \epsilon$ where $\epsilon \sim \mathcal{N}(0, I)$

- Optimize variational lower bound:

$$\mathcal{L}(x) = -\mathbb{E}_{q_\phi(z|x_{0:T})}[\log p_\theta(x_{t:T}|x_{0:t-1}, z)] + D_{KL}(q_\phi(z|z_{0:T})||p(z))$$

  where prior is $p(z) = \mathcal{N}(0, I)$

- Approximated posterior is conditioned on all of the frames, including the future frames $x_{t:T}$. This is feasible during training. While testing, latent variables are sampled from assumed prior.

- Two variants:

  1. Time Invariant: z is sampled once for the entire video.
  2. Time Variant: z is sampled once for every frame to be predicted. Generative model becomes

$$p(z_t)\prod_{t=c}^{T} p_\theta(x_t|x_{0:t-1}, z_t)$$

     and inference model becomes

$$q_\phi(z_t|x_{0:T})$$

- The main benefit of time-variant latent variable is better generalization beyond T, since the model does not have to encode all the events of the video in one vector z.

- In action-conditioned settings, we modify the generative model to be conditioned on action vector a_t.

$$p(z_t)\prod_{t=c}^{T} p_\theta(x_t|x_{0:t-1}, z_t, a_t)$$

37.3.1 Model Architecture

- CNN is used for $q_\phi(z|x_{0:T})$

- For $p(x_t|x_{0:t-1}, z)$, CDNA architecture proposed in Unsupervised Video Prediction (Finn et al) is used.

37.3.2 Training Procedure

- Training is done in 3 phases:

  1. Training the generative network: Inference network is disabled. z will be sampled from $\mathcal{N}(0, I)$
  2. Training the inference network: The inference network is trained to estimate the approximate posterior $q_\phi(z|x_{0:T})$; KL Loss is set to 0.
  3. Divergence reduction: KL Loss is added.

- Transition from second phase to third phase is done gradually.

37.4 Stochastic Movement Dataset

37.5 Experiments

- Baselines: Copy previous frame, CDNA, auto-regressive stochastic model, video pixel networks (VPN)

37.5.1 Datasets

- BAIR: 2 context frames, 10 predicted frames.

- Human 3.6M: Subsampled to 10fps, 10 context frames, 10 predicted frames.

- PUSH: 2 context frames, 10 predicted frames.

37.5.2 Quantitative Comparison

- PSNR, SSIM (best out of 100), Confidence of an object detector
- Evaluation Metrics:

37.5.3 Qualitative Comparison

37.6 Conclusion

38 Hierarchical Long-term Video Prediction without Supervision

Paper Details

- Authors: Nevan Wichers, Ruben Villegas
- Institutions: Google Brain, University of Michigan
- Project website: https://sites.google.com/view/vid-pred-without-supervision/home
- Code: https://github.com/brain-research/long-term-video-prediction-without-supervision
- Published in: ICML 2018
- Citations: 20 (As of 22-10-2019)

Abstract

38.1 Introduction

- Motivation: Intelligent agents
- Contributions:
  - An unsupervised approach for discovering high-level features necessary for long-term future prediction.
  - A joint training strategy for generating high-level features from low-level features and low-level features from high-level features simultaneously.
  - Use of adversarial training in feature space for improved high-level feature discovery and generation.
  - Long-term pixel-level video prediction for about 20 seconds into the future for the Human 3.6M dataset.

38.2 Related Work

38.3 Background

38.4 Method

- Loss Functions: l_2 loss, l_2 loss in feature space

38.5 Experiments

- Datasets: Human 3.6M
- Baselines: CDNA, SVGLP
- Evaluation Metrics: SSIM, 2AFC

39 Stochastic Video Generation with a Learned Prior (SVG-LP) - ICML 2018

Abstract

39.1 Introduction

- **Motivation**: Learning to generate future frames of a video sequence is a challenging research problem with great relevance to reinforcement learning, planning and robotics.
- Uncertainty in the dynamics of the world is the main issue in video prediction.

39.2 Related Work

39.3 Approach

- Model has 2 components:
  - Prediction model p_θ: Given previous frames ($\mathbf{x}_{1:t-1}$) and latent variable ($\mathbf{z}_t$), predicts next frame ($\hat{\mathbf{x}}_t$)
  - Loss: MSE + β KL$[q_\phi \| p]$

40 Video Prediction with Appearance and Motion Conditions (AMC-GAN)

Paper Details

- Authors: Yunseok Jang
- Institutions: University of Michigan
- Project website: https://sites.google.com/vision.snu.ac.kr/icml2018-video-prediction
- Code: https://github.com/YunseokJANG/amc-gan
- Published in: ICML 2018
- Citations: 16 (As of 09-04-2020)

Abstract

40.1 Introduction

- Motivation:
  - Video representation learning
- Contributions:
  -

40.2 Related Work

40.3 Approach

- Loss Functions:

40.4 Experiments

- Datasets: MUG facial expression, NATOPS human action
- Baselines: CDNA, BeyondMSE, MCnet
- Evaluation Metrics: Motion classifier accuracy
- Resolution: 64×64
- 4 context frames, 28 predicted frames (to be verified).

41 Learning to Decompose and Disentangle Representations for Video Prediction (DDPAE)

Paper Details

- Authors: Jun-Ting Hsieh
- Institutions: Stanford
- Project website:
- Code: https://github.com/jthsieh/DDPAE-video-prediction
- Published in: NIPS 2018
- Citations: 51 (As of 09-04-2020)

Abstract

41.1 Introduction

- Motivation:
- Contributions:
  - –

41.2 Related Work

41.3 Methods

- Loss Functions:

41.4 Experiments

- Datasets: Moving MNIST and Bouncing Balls
- Baselines:
- Evaluation Metrics: BCE, MSE, Cosine Similarity

42 Video Prediction via Selective Sampling (VPSS) (2018-NIPS)

Paper Details

- Authors: Jingwei Xu, Bingbing Ni, Xiaokang Yang
- Institutions: Shanghai Jiao Tong University
- Project website:
- Code: https://github.com/xjwxjw/VPSS. Tensorflow Implementation.
- Published in: NIPS 2018
- Citations: 1 (As of 05/09/2019)

Abstract

42.1 Introduction

- Motivation: Future Decision, Robot Manipulation, Autonomous driving.
- Using a combination of regression loss and adversarial loss results in competition between them instead of collaboration.
- Contributions:
  - Sample multiple future frames stochastically and then choose one among them or a combination. This reduces blur in the predicted videos.
  - To encourage collaboration between loss functions, design dedicated sub-networks for adversarial and regression loss respectively.
- Sampling module produces multiple high quality video frame proposals. Trained with adversarial loss.
- Selection module selects high possibility candidates from proposals and combines to produce the final prediction, according to the criteria of better position matching. Trained with regression loss.

42.2 Related Work

42.3 Method

- Losses used:
  - Adversarial Loss
  - Regression Loss (L2 loss)
- When using a loss function as a combination of adversarial and regression loss with a balancing hyper-parameter, only one of them can be reduced i.e. reducing one of the loss automatically increases the other.

42.3.1 Sampling Module

42.3.2 Selection Module

42.3.3 Design Considerations and Implementation Details

- Sub-networks are designed for different dedicated objectives.
  - The sampler is required to produce high quality proposals without requirement of motion accuracy.
  - Selector captures the motion information of previous inputs and select out proposals with high motion accuracy.

  These objectives are complementary, which essentially encourages cooperation between different sub-networks.

42.4 Experiments

42.4.1 Datasets and Evaluation Setup

- Datasets:
  - Moving MNIST: Resolution - 64x64
  - Robot Push: Resolution - 64x64
  - Human 3.6M: Resolution - 64x64; Human subject only takes a small portion of current frame, whose motion could easily be ignored only with the regression loss.
- Baselines and other models:
  - Dynamic Filter Networks (DFN)
  - CDNA
  - DrNet
  - MCNet
  - Stochastic Variational Video Prediction (SV2P)
  - Stochastic Video Generation with Learned Prior (SVG-LP)

42.4.2 Quantitative Evaluation

- PSNR
- SSIM
- Inception Score

42.4.3 Qualitative Evaluation

- 1270 prediction results of each dataset.
- 40 subjects.
- 3 aspects:
  1. Regarding the real video samples as baseline, which one is more realistic (Not based on previous inputs)?
  2. Considering image quality and previous inputs, which one is more similar to the Ground Truth?
  3. Considering motion accuracy and previous inputs, which one is more similar to the Ground Truth?

42.4.4 Discussion

42.5 Conclusion

43 FutureGAN: Anticipating the Future Frames of Video Sequences using Spatio-Temporal 3d Convolutions in Progressively Growing GANs

Paper Details

- Authors: Sandra Aigner, Marco Korner
- Institutions: Technical University of Munich
- Project website:
- Code: https://github.com/TUM-LMF/FutureGAN
- Published in: 2018
- Citations: 5 (As of 22-10-2019)

Abstract

43.1 Introduction

- Motivation: Robotics, autonomous driving, learning video representations.
- Contributions:
  - GAN based model for Video Prediction, which predicts multiple future frames at once.
- Uses Progressively Growing GANs (PG-GANs)

43.2 Related Work

43.3 FutureGAN Model

- Loss Functions: WGAN Loss, gradient-penalty, epsilon-penalty

43.4 Experiments

- Datasets: Moving MNIST, KTH, Cityscapes
- Baselines: Copying previous frame, FRNN, MCnet
- Evaluation Metrics: MSE, PSNR, SSIM
- On KTH dataset, prediction is done upto 120 steps.

44 Reduced-Gate Convolutional LSTM Using Predictive Coding for Spatiotemporal Prediction

Paper Details

- Authors: Nelly Elsayed
- Institutions:
- Project website:
- Code: https://github.com/NellyElsayed/rgcLSTM
- Published in: arXiv 2018, ICLR 2019 Reject
- Citations: 3 (As of 21-10-2019)

Abstract

44.1 Introduction

- Motivation:
- Contributions:
  - –

44.2 Related Work

44.3 Method

- Loss Functions:

44.4 Experiments

- Datasets: Moving MNIST, KITTI
- Baselines: PredNet, FC-LSTM, CDNA, DFN, VPN, ConvLSTM, ConvGRU, TrajGRU, PredRNN
- Evaluation Metrics: MSE, MAE, SSIM

45 Stochastic Adversarial Video Prediction (SAVP)

Abstract

- Learning to predict raw future observations, such as frames in a video, is exceedingly challenging—the ambiguous nature of the problem can cause a naively designed model to average together possible futures into a single, blurry prediction.
- Two recent approaches:
  - Latent variational variable models that explicitly model underlying stochasticity (VAEs). Doesn't produce realistic results.
  - Adversarially-trained models that aim to produce naturalistic images (GANs). Doesn't produce diverse predictions.
- Combine these 2 complementary approaches.
- Code:
  - https://alexlee-gk.github.io/video_prediction
  - https://github.com/alexlee-gk/video_prediction

45.1 Introduction

- **Motivation**: The ability to imagine future outcomes provides an appealing avenue for learning about the world. Unlabeled video sequences can be gathered autonomously with minimal human intervention, and a machine that learns to accurately predict future events will have gained an in-depth and functional understanding of its physical environment.
- Once trained, such a video prediction model could be used to determine which actions can bring about desired outcomes.
- Ambiguity in future makes the problem multimodel.
- Using MSE as loss function and a deterministic model leads to averaging of possible futures and hence produce blurry predictions.
- VAEs optimize a variational lower bound on the likelihood of the data in a latent variable model. However, the posterior is still a pixel-wise MSE loss, corresponding to the log-likelihood under a fully factorized Gaussian distribution. This makes training tractable, but causes them to still make blurry and unrealistic predictions when the latent variables alone do not adequately capture the uncertainty.
- GANs are notoriously susceptible to mode collapse, where latent random variables are often ignored by the model, especially in the conditional setting. Hence, diversity is lost.
- Our model consists of a video prediction network that can sample multiple plausible futures by sampling time-varying stochastic latent variables and decoding them into multiple frames.
- At training time, an inference network estimates the distribution of these latent variables, and video discriminator networks classify generated videos from real.
- The full training objective is the variational lower bound used in VAEs combined with the adversarial loss used in GANs. This enables to capture stochastic posterior distributions of videos while also modeling the spatiotemporal joint distribution of pixels.
- SSIM was not designed to handle spatial ambiguities.

45.2 Related Work

- 3 types of video prediction: Deterministic, low stochastic (stochasticity introduced only by noise), fully stochastic (latent vector)

45.3 Video Prediction with Stochastic Adversarial Models

- Our model consists of a recurrent generator network G, which is a deterministic video prediction model that maps an initial image x_0 and a sequence of latent random codes $z_{0:T-1}$ to the predicted sequence of future images $\hat{x}_{1:T}$
- Intuitively, the latent codes encapsulate any ambiguous or stochastic events that might affect the future.
- At test time, we sample videos by first sampling the latent codes from a prior distribution $p(z_t)$, and then passing them to the generator. We use a fixed unit Gaussian prior, $\mathcal{N}(0; 1)$.
- 64x64 frames. About 28–30 frames are predicted.
- Loss equations:

$$\mathcal{L}_1(G, E) = \mathbb{E}_{\mathbf{x}_{0:T}, \mathbf{z}_t \sim E(\mathbf{x}_{t:t+1})|_{t=0}^{T-1}} \left[\sum_{t=1}^{T} \|\mathbf{x}_t - G(\mathbf{x}_0, \mathbf{z}_{0:t-1})\|_1 \right]$$

$$\mathcal{L}_{\text{KL}}(E) = \mathbb{E}_{\mathbf{x}_{0:T}} \left[\sum_{t=1}^{T} \mathcal{D}_{\text{KL}}(E(\mathbf{x}_{t-1:t}) \| p(\mathbf{z}_{t-1})) \right]$$

$$G^*, E^* = \arg\min_{G,E} \lambda_1 \mathcal{L}_1(G, E) + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}(E)$$

$$\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{x_{1:T}} [\log D(x_{0:T-1})] + \mathbb{E}_{x_{1:T}, z_t \sim p(z_t)|_{t=0}^{T-1}} [\log(1 - D(G(x_0, z_{0:T-1})))]$$

$$G^* = \arg\min_{G} \max_{D} \mathcal{L}_{\text{GAN}}(G, D)$$

$$G^*, E^* = \arg\min_{G,E} \max_{D, D^{\text{VAE}}} \lambda_1 \mathcal{L}_1(G, E) + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}(E) + \mathcal{L}_{\text{GAN}}(G, D) + \mathcal{L}_{\text{GAN}}^{\text{VAE}}(G, E, D^{\text{VAE}})$$

45.4 Experiments

45.4.1 Evaluation Metrics

Study of realism, diversity, and accuracy of the generated videos.

- Realism: 2AFC (Two Alternative Forced Choice) test: Humans used as discriminators. Given a real and fake video, humans have to identify the real video.
- Diversity: Average distance between randomly sampled video predictions. Distance is measured in VGG space, (pretrained VGG model for ImageNet classification) averaged across five layers. **Todo: Check which layers from the code**
- Accuracy: The model is sampled a finite number of times (100 samples). The similarity between the best sample and ground truth is evaluated. Along with PSNR and SSIM, cosine similarity in pretrained VGG feature space is also used.
  - Why this metric?: One short-coming of diversity is that, even if predictions are diverse, we may not cover the entire feasible output space. Since we don't know the true distribution, we have no way of verifying if the predictions have convered the entire feasible output space. But since we have a reference video, we can check if one of the predictions is the reference video or not. So, we sample a finite video predictions and compare the closest resembling one with the reference video. If none of the predictions are close to the reference video, then this implies that we have not covered the entire feasible output space.
  - Cosine Similarity: Angle between two vectors: $\cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}||\ ||\vec{b}||}$.
    Todo: Check how it is done in code
  - Generalization ability is tested by running the model for more time steps than it was trained for.

45.4.2 Datasets

- BAIR action-free robot pushing dataset and KTH human actions dataset are used.
- BAIR:
  - Randomly moving robotic arm.
  - Resolution: 64x64
  - Conditioned on first 2 frames, next frames are predicted.
  - 10 frames are predicted for 2AFC test.
  - 28 frames are predicted for other experiments.
- KTH:
  - Frame resolution: 120x160. Frame rate: 25fps.
  - Preprocessing: Center-crop each frame to a 120x120 square and then resize to a spatial resolution of 64x64.
  - Conditioned on first 10 frames, next frames are predicted.
  - 10 frames are predicted for 2AFC test.
  - 30 frames are predicted for other experiments.
  - 6 activities: walking, jogging, running, boxing, hand-waving, hand-clapping
  - Stochasiticity is obtained only when all initial frames are empty. Stochasticity lies in what time human enters the frame.

45.4.3 Methods: Ablations and Comparisons

- Ablations:
  - GAN-only
  - VAE-only
  - deterministic
- Comparisons:
  - Stochastic Variational Video Prediction (SV2P)
  - Stochastic Video Generation with a Learned Prior (SVG-LP)
  - MoCoGAN

45.4.4 Experimental Results

- GAN based variants get high realism score but diversity is low. VAE based variants get high diversity but realism score is low.
- SV2P was not evaluated on KTH dataset. So, hyperparameters are not optimal.
- VGG similarity has been shown to better match human perceptual judgments.
- The general trend is that models trained with $\mathcal{L}_2$, which favors blurry predictions, are better on PSNR and SSIM, but models trained with $\mathcal{L}_1$ are better on VGG cosine similarity.
- We expect for our GAN-based variants to underperform on PSNR and SSIM since GANs prioritize matching joint distributions of pixels over per-pixel reconstruction accuracy.
- VGG similarity is a held-out metric, meaning that it was not used as a loss function during training.
- On stochastic environments (BAIR action free), there is some correlation between diversity and accuracy - a model with diverse prediction is more likely to sample a video that is close to ground truth.

- On less stochastic environments (KTH conditioned on 10 frames), the above correlation doesn't hold.
- KTH dataset conditioned on 10 frames is less stochastic because there is not much difference between deterministic and savp models.
- On stochastic datasets, adding GAN to VAE model doesn't reduce diversity, but improves realism.

46 Stochastic Dynamics for Video Infilling (SDVI)

Abstract

- Stochastic generation framework to infill long intervals of video sequences.
- Video interpolation aims to produce transitional frames for a short interval between every two frames. Video Infilling, however, aims to complete long intervals in a video sequence.

46.1 Introduction

- Video interpolation is used for videos with frame rate above 20fps. This paper focuses on long-term interval filling for videos with frame rate less than 3fps.
- Application: Low frame rate camera with limited memory.
- Uncertainties and randomness in long-term interval is greater compared to short-term intervals.
- Utilizing long-term information can benefit the dynamics inference by eliminating uncertainties.
- In this paper, motion dynamics is modelled same as video prediction. It has 2 major challenges:
  - Unlike the video prediction, the temporal layout of our input is bi-directional with temporal gaps.
  - Interpolation requires coherency between the generated sequence and reference frames from both directions.

46.2 Related Work

46.3 Methods

46.4 Experiments

- 4 datasets: Stochastic Moving MNIST (SM-MNIST), KTH action database, BAIR robot pushing dataset and UCF101.
- LMS: Last Momentum Similarity: Mean Squared distance between optical flow from X_{T-1} to X_T and the optical flow from $\tilde{X}_{T-1}^{infr}$ to X_t
- Ablation studies are conducted by removing the spatial sampling or the extended reference frames in $\mathbf{X_{WR}}$.
- **Code not yet released. Code will be released after the paper review process.**

47 TGANv2: Efficient Training of Large Models for Video Generation with Multiple Subsampling Layers

Abstract

- Datasets: UCF101, FaceForensics
- Resolution: 192x192 and 256x256.
- 16 frames generated.

48 Quality Assessment of In-the-Wild Videos

Paper Details

- Authors: Dingquan Li, Tingting Jiang, Ming Jiang
- Institutions: Peking University
- Project website:
- Code: https://github.com/lidq92/VSFA
- Published in: ACM Conference on Multi-Media 2019
- Citations: 0 (As of 24/12/2019)

Abstract

48.1 Introduction

- Motivation:
- Contributions:
  - –

48.2 Related Work

48.3 The Proposed Method

- Main components
  - Content Aware feature extraction using ResNet
  - Mean and Standard Deviation pooling
  - Dimensionality Reduction using Fully Connected Layers
  - GRU for long-term dependencies
  - Subjectively inspired temporal pooling

48.4 Experiments

49 Order Matters: Shuffling Sequence Generation for Video Prediction (SEE Net)

Paper Details

- Authors: Junyan Wang
- Institutions: Newcastle University, UK.
- Project website:
- Code: https://github.com/andrewjywang/SEENet
- Published in: BMVC 2019
- Citations: 0 (As of 20-10-2019)

Abstract

49.1 Introduction

- Motivation: Robotics and healthcare
- Contributions:
  - SEE Net for long-term future frame prediction.
  - Shuffle discriminator to explicitly control the extraction of sequential information.

49.2 Related Work

49.3 Problem Statement

49.4 SEE-Net

- Loss Functions: Adversarial Loss, Shuffle Loss, Consistency Loss.
- Optical Flow between frames is fed to the network. PWCNet is used to obtain Optical Flow.

49.5 Experiments

- Datasets: Moving MNIST, KTH, MSR
- Baselines: MCnet, DrNet
- Evaluation Metrics: PSNR, SSIM

50 Predicting Future Frames using Retrospective Cycle GAN

Paper details

- Authors: Yong-Hoon Kwon, Min-Gyu Park
- Institutions: LG Electronics
- Project website:
- Code:
- Published in: CVPR 2019
- Citations: 1 (As of 20-10-2019)

Abstract

- The key idea is to train a single generator that can predict both future and past frames while enforcing the consistency of bi-directional prediction using the retrospective cycle constraints.

50.1 Introduction

- Motivation: Abnormal event detection, video coding, video completion, robotics, and autonomous driving.
- Contributions:
  -

50.2 Related Work

50.3 Proposed Method

- Loss Functions: L1 loss between frames, L1 loss between LoG of frames, frame adversarial loss, sequence adversarial loss.

50.4 Experimental Results

- Datasets: KITTI, Caltech Pedestrian, UCF101, Surveillance Videos
- Baselines: Copying previous frame, PredNet, DM-GAN, BeyondMSE, ContextVP, MCnet+RES, EpicFlow, DVF
- Evaluation Metrics: MSE, PSNR, SSIM

50.4.1 Datasets

50.4.2 Training Details

50.4.3 Quantitative and qualitative evaluation

50.4.4 Multi-step prediction evaluation

- Trained to predict 1 frame. Can predict upto 15 frames.

50.5 Conclusion

51 Disentangling Propagation and Generation for Video Prediction

Paper Details

- Authors: Hang Gao, Fisher Yu, Trevor Darrell
- Institutions: UC Berkeley
- Project website:
- Code: https://github.com/Fangyh09/Disentangling-Propagation-and-Generation-for-Video-Prediction
- Published in: ICCV 2019
- Citations: 10 (As of 11-05-2020)

Abstract

51.1 Introduction

- Motivation:
- Contributions:
  - –

51.2 Related Work

51.3 Method

- Loss Functions:

51.4 Experiments

- Datasets:
- Baselines:
- Evaluation Metrics:

51.5 My Summary

Predict optical flow and warp the last frame to get a coarse next frame. From this, generate a confidence map, which gives low confidence in disoccluded areas. Use an Generator (inpainting module) to fill in disoccluded areas.

52 Spatio-Temporal Measures Of Naturalness

Paper Details

- Authors: Zeina Sinno, Alan Bovik
- Institutions: UT Austin
- Project website:
- Code:
- Published in: ICIP 2019
- Citations: 0 (As of 10-02-2020)

Abstract

52.1 Introduction

- Early visual system is shaped to properties of the natural environment, including important statistical regularities.
- Many VQA models leverage the fact that undistorted videos present statistical regularities that are systematically and predictably degraded by distortions.
- Microsaccades are not studied yet.
- Bandpass processed natural images have highly regular first and second-order statistics, which are perturbed by distortions.
- This paper modeled statistics of frame differences of natural videos that are oriented in (spatially displaced) space-time.
- This paper devised space-time directional natural video statistics (NVS)

52.1.1 Space-Time Directional Models

- Four directional temporal differences:

$$D_H(i,j)_t = I_t(i,j) - I_{t+1}(i,j-1) \tag{1}$$
$$D_V(i,j)_t = I_t(i,j) - I_{t+1}(i-1,j) \tag{2}$$
$$D_{D_1}(i,j)_t = I_t(i,j) - I_{t+1}(i-1,j-1) \tag{3}$$
$$D_{D_2}(i,j)_t = I_t(i,j) - I_{t+1}(i-1,j+1) \tag{4}$$

- MSCN coefficients are computed with a 7 x 7 window, and Gaussian weights (set to 3σ)
- 96×96 patches are used
- The empirical distribution (histogram) of the directional MSCN coefficients from each patch are fit to a zero-mean GGD by Method of Moments (MoM). This is done for each of the space-time directions.
- For a given video, 4 directional NVS models are obtained. Each vector NVS model is of size $p \times 2$, where p is the total number of patches in the video.
- Each model is obtained on the Luminance frame of the video.
- This is done on a collection of high quality videos, selected by a small Subjective Study.
- Since most of the videos are of resolution 1920×1080 and consumer devices can produce wide range of video resolutions, randomly selected videos were downscaled to one of 1280×720, 960×640 and 640×360.
- There are about 50 videos for each resolution.

- Any video longer than 10 seconds is trimmed to 10s.
- For higher resolution videos (1920×1080 and 1280×720), only 5% of the patches are selected based on sharpness. For lower resolution videos (960×640 and 640×360), 25% of the patches are selected.
- The best fitting parameters (α,β) were collected and horizontally aggregated over space and time across all videos for each of the four orientations, yielding 4 pristine models P_H, P_V , P_{D_1} and P_{D_2}.
- Mahalanobis distance is computed between the each directional pristine model and the corresponding directional input video models. Higher the distance, lower the quality.

52.2 Is Directional Naturalness a Good Predictive Measure of Quality?

- PLCC, SROCC and RMSE are computed between negative values of the distances and MOS. For PLCC and RMSE, non-linear mapping is applied.
- A scatter-plot of MOS v/s these distances is also plotted.
- Results:
  - PLCC ≈ 0.6
  - SROCC ≈ 0.6
  - RMSE ≈ 13.58

53 Bounce and Learn - Modeling Scene Dynamics with Real-World Bounces (ICLR 2019, CMU)

Abstract

53.1 Introduction

- **Motivation - Robotics and Augmented Reality**: The ability for a system to make such predictions will allow applications in augmented reality and robotics, such as compositing a dynamic virtual object into a video or allowing an agent to react to real-world bounces in everyday environments.
- From videos, physical properties of the surfaces are approximated. This is then used as supervision to learn an appearance-based estimator.
- **Goal**:
  - Predict post-bounce trajectories
  - Estimate surface-varying coefficients of restitution (COR) and effective collision normals.
- To model collision events, often rigid-body physics are used. But during collision real-world objects deform and hence violate rigid-body assumptions.
- **Contributions**:
  - Predicting post bounce trajectories
  - Inferring physical properties (COR and collision normal) from single still image
  - Bounce Dataset: large-scale dataset of real-world bounces in a variety of everyday scenes

53.2 Related Work

53.3 Bounce and Learn Model and Bounce Dataset

53.3.1 Physics Inference Module (PIM)

- Physics Inference Module (PIM) is pretrained on simulation data
- Input: 10 frames; Output: 10 frames
- Encode past 10 frames into a vector. Using physical properties, predict the output vector. Decode the output vector into frames.
- Training Loss for PIM: Distance between encodings of predicted and actual trajectory (after collision). Note: There are other terms as well.

53.3.2 Visual Inference Module (VIM)

- Visual Inference Module (VIM) takes input as as the image of a scene and outputs the physical parameters for each location in the scene.
- VIM is trained end-to-end along with PIM.
- Online Learning: Estimates of physical parameters in a scene are updated online upon observing bounces. This is useful in robotics where an agent can interact with environment to infer the physical properties.

53.3.3 Bounce Dataset

- Real-world Bounce Dataset:
  - 5172 stereo videos
  - On average, each video contains 172 frames with the ball.
  - Each sample in the dataset consists of the RGB frames, depth maps, point clouds for each frame, and estimated surface normal maps.
- Simulation Data:
  - Simulate a set of sphere-to-plane collisions with the PyBullet Physics Engine

53.4 Evaluation

53.4.1 Visual Forward Prediction

- Split of dataset: No common scenes across the splits.
  - Training: 4503
  - Validation: 196
  - Test: 473
- Mix of 3:1 synthetic-to-real data in the mini-batches.
- Quantitative Evaluation: $\mathcal{L}_2$ distance between world coordinates of ball center of the predicted and ground truth at time step 0.1 seconds post-bounce.
- Baselines:
  1. Replace PointNet encoder with 'center-encoding' model
  2. Parabola encoding: Fit a parabola to ball centre positions in a Least-Squares sense. Parameters of this parabola are input to a neural net.

54 FVD - A new Metric for Video Generation

Paper Details

- Authors: Thomas Unterthiner, ...
- Institutions: Google Brain
- Project website:
- Code:
- Published in: ICLR 2019 (Workshop paper)
- Citations: 0 (As of 13-02-2020)

Abstract

54.1 Introduction

- Generative models of video will enable many applications, including missing-frame prediction, improved instance segmentation, or complex (relational) reasoning tasks by conducting inference.
- FVD builds on principles of FID.

54.2 Frechet Video Distance

- Code: https://git.io/fpuEH
- It is difficult to solve Frechet Distance for general distributions, but for multivariate Gaussians, it has a closed for solution.

$$d(P_R, P_G) = |\mu_R - \mu_G|^2 + \mathrm{Tr}\left(\Sigma_R + \Sigma_G - 2(\Sigma_R\Sigma_G)^{\frac{1}{2}}\right)$$

- The multivariate Gaussian is a reasonable approximation in a suitable feature space.
- Inflated 3D ConvNet (I3D) is used to extract features. Frechet distance on these features gives FVD.
- I3D network generalizes Inception architecture to sequential data and is trained to perform action recognition on Kinetics dataset.
- Kernel Video Distance (KVD): Uses Maximum Mean Discrepancy (instead of Frechet distance). Polynomial kernel is used: $\kappa(a, b) := \left(a^T b + 1\right)^3$

54.3 Experiments

54.3.1 Noise Study

- Static and temporal noise are added at 6 different levels.
- Videos from BAIR, Kinetics-400 and HMDB51

54.3.2 Human Evaluation

- Where other metrics like PSNR and SSIM cannot distinguish between good models, FVD can reliably distinguish.
- No other metric can reliably distinguish between good models that are equal in terms of FVD.
- No other metric can improve upon the ranking induced by FVD.

54.4 Conclusion

54.5 Appendix

54.5.1 Appendix A: Noise Study

- Static noise:
  - Black rectangle
  - Gaussian blur
  - Gaussian noise
  - Salt and pepper noise
- Temporal noise:
  - Locally swapping randomly chosen frames with its neighbor
  - Globally swapping random frames
  - Interleaving frames
  - Switching to different video in middle.

54.5.2 Appendix B: Effect of Sample size on FVD

54.5.3 Appendix C: Human Evaluation

- Models:
  - CDNA
  - SV2P
  - SVG-FP
  - SAVP
- BAIR data set.
- 3000 models by varying hyperparameters.
- 2 context frames, 14 predicted frames.
- PSNR and SSIM are computed on 100 videos and the best score is chosen.
- 256 videos to compute FVD
- **One Metric Equal**: For a given metric, 10 models are chosen which cannot be distinguished by this metric, which are at 75% good. Videos of these models are evaluated on other metrics and also with subjective evaluation.
- **One Metric Spread**: For a given metric, 10 models are chosen with scores between 10% to 90% percentile. Videos of these models are evaluated on other metrics and also with subjective evaluation.
- Human raters are showed 2 videos and then asked to identify which looks better or are they same. 3 raters for each pair.

55 Reasoning about Physical Interactions with Object-Oriented Prediction and Planning (O2P2)

Paper Details

- Paper: arXiv, OpenReview
- Authors: Michael Janner, Sergey Levine, William T Freeman, Joshua B Tenenbaum, Chelsea Finn, Jiajun Wu
- Institutions: UC Berkeley, MIT
- Project website: https://people.eecs.berkeley.edu/~janner/o2p2/
- Code: Generating training data
- Published in: ICLR 2019
- Citations: 31 (As of 24-06-2020)

Abstract

- O2P2 model jointly learns
  - A perception function to map from image observations to object representations
  - A pairwise physics interaction function to predict the time evolution of a collection of objects
  - A rendering function to map objects back to pixels

55.1 Introduction

- Given a scene, can you describe how we arrived at the scene? Which objects were moved first and which were moved later?
- O2P2: Object-Oriented Prediction and Planning: we train an object representation suitable for physical interactions without supervision of object attributes.
- We evaluate our learned model not only on the quality of its predictions, but also on its ability to use the learned representations for tasks that demand a sophisticated physical understanding.

55.2 Object-Oriented Prediction and Planning (O2P2)

- O2P2 consists of three components, which are trained jointly
  - A perception module that maps from an image to an object encoding. The perception module is applied to each object segment independently.
  - A physics module to predict the time evolution of a set of objects. We formulate the engine as a sum of binary object interactions plus a unary transition function.
  - A rendering engine that produces an image prediction from a variable number of objects. We first predict an image and single-channel heatmap for each object. We then combine all of the object images according to the weights in their heatmaps at every pixel location to produce a single composite image.

55.2.1 Perception Module

55.2.2 Physics Module

- $\bar{o}_k = f_{trans}(o_k) + \sum_{j \neq k} f_{interact}(o_k, o_j) + o_k$
- **Idea! May be this Physics module can be used to validate laws of physics while assessing a video**

- **Question: Here we consider only effect of one object's movement on another object. What if there is transitive effect? Movement of object A affects object B which in turn affects object C. How is this captured?**
- **They talk about simplifying the problem by considering only action planning instead of physical interactions. Did not understand what that is**

55.2.3 Rendering Engine

- We train such that lower heatmap values are used for nearer objects. No true depth-maps are used while training.

55.2.4 Learning Object Representations

- S_0 – Segmented Image, $\mathbf{O} = f_{percept}(S_0)$, $\bar{\mathbf{O}} = f_{physics}(\mathbf{O})$, $\hat{I}_0 = f_{render}(\mathbf{O})$, $\hat{I}_1 = f_{render}(\bar{\mathbf{O}})$
- We compare each image prediction $\hat{I}_t$ to its ground-truth counterpart using both $\mathcal{L}_2$ distance and a perceptual loss $\mathcal{L}_{\text{VGG}}$.
- Perceptual Loss: $\mathcal{L}_2$ distance is feature space of pretrained VGG network.
- Losses for individual engines:
  - Perception Module: Reconstruction of I_0: $\mathcal{L}_{\text{percept}} = \mathcal{L}_2(\hat{I}_0, I_0) + \mathcal{L}_{\text{VGG}}(\hat{I}_0, I_0)$
  - Physics Engine: Reconstruction of I_1: $\mathcal{L}_{\text{physics}} = \mathcal{L}_2(\hat{I}_1, I_1) + \mathcal{L}_{\text{VGG}}(\hat{I}_1, I_1)$
  - Rendering Engine: Reconstruction of both I_0 and I_1: $\mathcal{L}_{\text{render}} = \mathcal{L}_{\text{percept}} + \mathcal{L}_{\text{physics}}$

55.2.5 Planning with Learned Models

- High level algorithm:
  - Given the segmented goal image, perception module extracts object representations (O^{goal})
  - Sample actions (initial frame)
  - For each sample action (initial frame), extract object representations using perception module. Apply physics engine to predict the end-result. Compare end-goal with end-result using $\mathcal{L}_2$ distance between object representations (not $\mathcal{L}_2$ in pixel domain).
  - Choose the sampled action with least $\mathcal{L}_2$ distance. Execute the action in MoJoCo.
  - Repeat as many actions as the number of objects in the given goal image.

55.3 Experimental Evaluation

Goal:

- Can O2P2 reason physical interactions in an actionable way
- Does object factorization helps in video prediction i.e. does it perform better than black-box video prediction?
- Is object factorization useful, even without supervision?

55.3.1 Image Reconstruction and Prediction

- **MuJoCo** simulator is used to generate training data.
- 60000 training images: Only initial and final frames.

55.3.2 Building Towers

- Compared with No Physics Ablation, SAVP, Oracle (pixels) and Oracle(objects) models.
- Evaluation metric:
  - Simulator: Object error is computed in terms of position, identity and color and then thresholded to get a binary value. Then accuracy is computed as fraction of goals achieved. Only relative ordering matters. Metric values differ based on threshold.
  - Robotic arm: Percentage of goal configurations built.
- SAVP model failed in stacking objects to build towers.

55.3.3 The importance of understanding physics

- Although the model is never trained to produce valid action decisions, the planning procedure selects a physically stable sequence of actions.

55.3.4 Transfer to Robotic Arm

- Given a goal image, they make a Sawyer robotic arm place the appropriate objects at appropriate places in appropriate sequence
- Two layer MLP to map actions to object representations: $o_m = f_{\text{embedder}}(a_m)$

55.4 Related Work

- This item is in between two kinds of works
  - Regid notion of object representation via supervision.
  - No scene factorization into objects.
- We show that a model capable of predicting and reasoning about physical phenomena can also be employed for decision making.
- This model achieves substantially better results at tasks that require building structures out of blocks.

55.5 Conclusion

55.6 My summary

Given a goal image made up of blocks, O2P2 predicts the action sequences to be performed to achieve the goal image.

For every action, multiple initial frames are sampled and fed to physics engine. It outputs the steady state result of that action. This result is compared to end-goal image in object representation space using $\mathcal{L}_2$ distance and the action that has least distance is picked. This action is given to MoJoCo simulator and the result is obtained. This is repeated for as many number of actions as the number of objects in the end-goal image.

The claim is that physics engine takes care of stability and in turn leads to appropriate sequence of actions. They conclude that for building towers O2P2 (applying physics on object representations) is better than object agnostic models (applying physics directly in pixel space).

56 Time Agnostic Prediction (Conference Paper - ICLR 2019)

57 Deep Compressed Sensing

Abstract

57.1 Introduction

- Compressed Sensing (CS) provides a framework that separates encoding and decoding into independent measurement and reconstruction processes.
- Autoencoders are trained end-to-end. CS reconstructs via online optimization.
- Neural Networks are trained from scratch for both measuring and online reconstruction.
- Contributions:
  - How to train Deep Nets within CS framework
  - Meta-learned reconstruction process is more accurate and faster
  - New GAN based on latent optimzation
  - Extension of this framework to semi-supervised GANs. Gives meaningful latent spaces.

57.2 Background

57.2.1 Compressed Sensing

- $\mathbf{m} = \mathbf{F}\mathbf{x} + \eta$
  where
  x is the signal
  F is CxD measurement matrix. Typically wide/fat (C <<D)
  η is measurement noise

- Restricted Isometry Property (RIP)

$$(1-\delta)\|x_1 - x_2\|_2^2 \leq \|F(x_1 - x_2)\|_2^2 \leq (1+\delta)\|x_1 - x_2\|_2^2$$

  $\delta \in (0, 1)$ is a small constant.
  It states that projection from $\mathbf{F}$ preserves the distance between 2 signals, bounded by factors of (1-δ) and (1+δ)
  This property holds with high probability for various random matrices $\mathbf{F}$ and sparse signals x

- Compressed Sensing guarantees that minimizing the mesurement error under the constraint that x is sparse, leads to accurate reconstruction $\hat{x} \approx x$ with high probability.

$$\hat{x} = \arg\min_x \|\mathbf{m} \text{ - } \mathbf{F}\mathbf{x}\|_2^2$$

- The contraint that x being sparse can be replaced by sparsity in a set of basis ϕ i.e. z is sparse, but $x = \phi z$ need not be. Measurement function is applied on $x = \phi z$. In other words, the signal should be sparse in transform domain.

57.2.2 Compressed Sensing using Generative Models (CSGM)

- Sparse basis like Fourier or Wavelet restricts to very few data distributions. To relax this, a Generator (neural network) is used map a latent representation z to signal space x:

$$x = G_\theta(z)$$

- This implicity constrains x in a low-dimensional manifold. This constraint provides Set-Restricted Eigenvalue Condition (S-REC) with random matrices.

- With this condition, low reconstruction error with high probability can be achieved similar to CS.

$$\begin{aligned} \hat{z} &= \arg\min_{z} E_\theta(m, z) \\ E_\theta(m, z) &= \|m - FG_\theta(z)\|_2^2 \\ \hat{x} &= G_\theta(z) \qquad\qquad \text{is the reconstructed signal} \end{aligned}$$

- Compared to previous one, where the optimization is directly in signal space x, here the optimization is in latent space z

- E_θ is intractable. Hence Gradient Descent is used.

$$\hat{z} \sim p_z(z)$$
$$\hat{z} \leftarrow \hat{z} - \alpha \frac{\partial E_\theta(m, z)}{\partial z}\bigg|_{z=\hat{z}}$$

- T steps of Gradient Descent are used. Typically thousands of gradient descent steps are required with several restarts to get a sufficiently good $\hat{z}$
- Disadvantages of CSGM
  - Optimization is slow.
  - Random measurement matrices are sub-optimal.

57.2.3 Model Agnostic Meta Learning (MAML)

- Train a generic model, which can be adapted to any new task with minimal training.
- Even if the loss function is highly non-convex, by back-propagating thru gradient-descent process, only few gradient steps are enough to adapt to new tasks.

57.2.4 Generative Adversarial Networks

- In most GAN models, discriminators become useless after training. Here, discriminators are used to move latent representations to areas more likely to generate realistic images.

57.3 Deep Compressed Sensing

Two stages:

- Combine Meta Learning with CSGM
- Measurement matrices are replaced by functions (DeepNets)

57.3.1 Compressed Sensing with Meta Learning

- Refer to Algorithm 1
- Meta Learning is used in Latent Optimization
- Model parameters and latent variables are trained to minimize measurement error

$$\min_\theta \mathcal{L}_G = \mathbb{E}_{x_i \sim p_{data}(x)} \left[E_\theta(m_i, \hat{z}_i) \right]$$

- Gradient of E_θ is not stochastic gradient since there is only one sample of z
- Online optimization is over latent variables rather than parameters. Since latent variables are much fewer, update is quicker.
- Enforcing RIP is equivalent to minimizing the below

$$\mathcal{L}_F = \mathbb{E}_{x_1, x_2} \left[(\|F(x_1 - x_2)\|_2 - \|x_1 - x_2\|_2)^2 \right]$$

57.3.2 Deep Compressed Sensing with Learned Measurement Function

57.3.2.1 Learning Measurement Function

- Refer to Algorithm 2
- Measurement Matrix **F** is made trainable $F_\phi(x)$
- All equations are similar

$$
\begin{aligned}
E_\theta(m, z) &= \|m - F_\phi(G_\theta(z))\|_2^2 \\
\mathcal{L}_G &= \mathbb{E}_{x_i \sim p_{data}(x)} \left[E_\theta(m_i, \hat{z}_i)\right] \\
\mathcal{L}_F &= \mathbb{E}_{x_1, x_2} \left[(\|F_\phi(x_1 - x_2)\|_2 - \|x_1 - x_2\|_2)^2\right] \\
&\min_{\theta,\phi}(\mathcal{L}_G + \mathcal{L}_F)
\end{aligned}
$$

57.3.2.2 Generalised CS 1: CS-GAN

- 1D measurement i.e. Measurement Function encodes how likely the input data is real or fake
- One way to formulate this is to train measurement function F_ϕ using the below loss

$$
\mathcal{L}_F = \begin{cases} \|F_\phi(x) - 1\|_2^2 & x \sim p_{\text{data}}(x) \\ \|F_\phi(\hat{x})\|_2^2 & \hat{x} \sim G_\theta(\hat{z}), \forall \hat{z} \end{cases}
$$

  This becomes similar to Least-Squares GAN

- To get original GAN, set T=0 (in algorithm 2) and use a binary classifier D_ϕ as measurement function which gives the probability that x comes from the dataset. This measurement function is equivalent to Discriminator. We need to use cross-entropy loss rather than squared loss

$$
\mathcal{L}_F = t(x)\ \ln[D_\phi(x)] + (1 - t(x))\ \ln[1 - D_\phi(x)]
$$

  where

$$
t(x) = \begin{cases} 1 & x \sim p_{data}(x) \\ 0 & x \sim G_\theta(z),\ \forall z \end{cases}
$$

58 ChainQueen: A Real-Time Differentiable Physical Simulator for Soft Robotics

Paper Details

- Paper: arXiv, IEEE Xplore
- Authors: Joshua Tenenbaum, William Freeman, Jiajun Wu
- Institutions: MIT
- Project website:
- Code: https://github.com/yuanming-hu/ChainQueen
- Published in: ICRA 2019
- Presentation: video
- Citations: 22 (As of 24-06-2020)

Abstract

58.1 Introduction

- Motivation: Differentiable Physical Simulators enable the use of gradient-based optimizers for inverse-physics problems.
- Contributions:
  - Differentiable Physical Simulator for deformable objects.

58.2 Related Work

58.3 Forward Simulation and Back Propagation

- Loss Functions:

58.4 Evaluation

- Efficiency: Time taken per frame for a given number of particles.
- Accuracy: Relative error in forward simulation and gradient.
- Evaluation Metrics:

58.5 My Summary

-

59 High Fidelity Video Prediction with Large Stochastic Recurrent Neural Networks

Paper Details

- Authors: Ruben Villegas
- Institutions: University of Michigan, Google Research, Adobe Research
- Project website: https://sites.google.com/view/videopredictioncapacity
- Code:
- Published in: NIPS 2019
- Citations: 1 (As of 09-04-2020)

Abstract

59.1 Introduction

- Motivation:
- Contributions:
  - –

59.2 Related Work

59.3 Method

- Loss Functions:

59.4 Experiments

- Datasets: Towel pick, Human 3.6M, KITTI
- Baselines:
- Evaluation Metrics:

60 Large-Scale Study of Perceptual Video Quality (LIVE VQC Database)

Paper Details

- Authors: Zeina Sinno, Alan Bovik
- Institutions: UT Austin
- Project website: http://live.ece.utexas.edu/research/LIVEVQC/index.html
- Code:
- Published in: TIP 2019 Feb
- Citations: 10 (As of 11-02-2020)

Abstract

60.1 Introduction

- Large database of 585 videos, 80 videographers, 43 different models, 101 unique devices, 205000 opinion scores.
- Has real world distortions rather than synthesized distortions.
- Most existing video quality assessment databases offer only very limited varieties of video content, shot by only a few users, thereby constraining the ability of learned VQA models trained on them to generalize to diverse contents, levels of videographic expertise, and shooting styles.
- Previous study [20] did not take care of video stalling or frame freezing during subjective study.
- In KoNViD-1k database, subject was asked to rate any number of videos within range 10–550. This leads to subject fatigue.
- Contributions:
  - A new robust framework for conducting crowd-sourced video quality studies and for collecting video quality scores in AMT.
  - New LIVE VQC database.

60.2 LIVE Video Quality Challenge (LIVE-VQC) Database

- Features:
  - 585 videos
  - 80videographers
  - 43 different models
  - 101 unique devices
  - Min 10s video
  - No post processing (like Snapchat)
  - No single contributor captured more than 9% of the videos.
  - Contributors span age range of 11–65 years.
  - No distortion labels to videos
- Table of different video resolutions
- All portrait videos of resolutions 1080×1920, 2160×3840 and 720×1080 were downscaled using bicubic interpolation to 404×720. This is done so that all these videos can be displayed at native resolutions for subjective study.

60.3 Testing Methodology

- Single Stimulus study
- 50 videos in a session: 7 training and 43 testing.
- 43 test videos include:
  - 4 distorted videos from LIVE IQA (golden videos).
  - 31 random videos from the new database.
  - 4 repeated videos from above 31
  - 4 videos are rated by all subjects.
- Initial position of cursor was randomized.
- Subject Features:
  - 4776 subjects
  - 56 countries
  - 1 dollar per subject
  - 205000 opinion scores
  - 205 opinion scores per video (after rejection)

60.4 Subjective Data Processing and Results

- Subject Consistency is checked by dividing scores on a video into 2 group and checking correlation between the means of the thwo groups.

60.5 Performance of Video Quality Predictors

- For models that require training, 5 fold cross validation technique was used. The scores from each fold are aggregated. Scatter plot of NIQE, VIIDEO, BRISQUE and V-BLIINDS against MOS scores. VIIDEO correlated poorly.
- For evaluating models, 100 splits were used. Median PLCC, SROCC and RMSE values are reported.

61 Predicting the Quality of Images Compressed After Distortion in Two Steps

Paper Details

- Authors: Xiangxu Yu, Praful Gupta, Alan Bovik
- Institutions: UT Austin, Netfilx
- Project website: http://live.ece.utexas.edu/research/twostep/index.html
- Code: https://github.com/xiangxuyu/2stepQA
- Published in: TIP 2019
- Citations: 1 (As of 08-02-2020)

Abstract

61.1 Introduction

- Pictures captured by non profession users are already of lower quality. Quality degrades more on compressing them.
- FR metrics don't consider the already poor quality of reference images. NR metrics don't perform so well. Since there is a reference available, it can be used to get better predictions of quality.
- Both Full Reference (FR) and Reduced Reference (RR) IQA models will be referred to as R IQA models.

61.2 Related Work

61.3 Two Step IQA Model

- R IQA models predict quality of distorted images by making a perceptual comparison of it with a reference image.
- R IQA model is actually a perceptual fidelity measure.
- R IQA models only provide relative image quality scores. If the reference image itself is of lower quality, that is not taken into account. By augmenting a NR IQA score with R IQA score, a better prediction of perceptual distance from the natural image space can be made.
- Generally, a two step model should fulfill three important properties:
  - If compression does not occur, or has an imperceptible effect on quality, then the two-step model should report the innate source (reference) image quality.
  - If the source is pristine, then the two-step model should accurately predict the effect of compression on perceived quality.
  - If the source is already distorted and then compressed with perceptual loss, then the two-step model should yield a better prediction than either the R and NR components applied on the compression image.
- 2 step model:
  - Predict the quality of reference image using a NR IQA model (prior quality) - Q_{NR}.
  - Predict the quality of distorted image w.r.t. reference image using a R IQA model (conditional quality) - Q_{R}.
  - Re-map the scores to same interval - Q'_{NR} and Q'_{R}.
  - Take the product of the two scores - $Q'_{\mathrm{NR}} \cdot Q'_{\mathrm{R}}$

- Example: Using MS-SSIM as R IQA measure and NIQE as NR IQA measure, we get

$$Q_{\text{2stepQA}} = \text{MS-SSIM} \cdot \left(1 - \frac{\text{NIQE}}{\alpha}\right)$$

  α is a scaling constant and is set to 100.
  2stepQA is robust over a wide range of the values of $\alpha \in [50, 150]$.

- Logistic Remapping: To preserve monotonicity, allow for generalizability, and to scale the scores to either [0, 1] or the MOS range, a simple four-parameter logistic mapping is used.

$$Q' = \beta_2 + \frac{\beta_1 - \beta_2}{1 + e^{-(Q - \beta_3/|\beta_4|)}}$$

  The parameters β can be effectively determined by using the subjective data from one or more IQA databases.

- If a model is trained on MOS scores, it need not be remapped.

- Instead of simple product, a general model can have exponentially weighted product of the two scores:

$$Q_{\text{G}} = (Q'_{\text{NR}})^{\gamma} \cdot (Q'_{\text{R}})^{1-\gamma} \qquad \text{where} \quad \gamma \in [0, 1]$$

61.4 A new Distorted-Then-Compressed Image Database

- The new database is called 'LIVE Wild Compressed Picture Quality Database'.
- 80 images are chosen from LIVE In the Wild Challenge IQA Database. These images are subjected to JPEG compression at 4 different levels. Thus a there are a total of 400 images.
- Subjective Study: 6 out of 29 subjects were outliers.
- Subject Consistency Analysis: The subjects are divided into two disjoint equal groups and MOS is computed on each image for both the groups. SROCC is computed between the scores and this is repeated 25 times. SROCC was found to be 0.9805.
- Box plot of MOS at different compression levels is plotted.
- A line graph of MOS scores for each content image at different compression levels is plotted

61.5 Performance Evaluation

- Since DMOS doesn't represent actual quality, MOS is used here.
- LCC and SROCC are used for evaluation.
- Predicted IQA scores were passed through a logistic non-linearity before computing the LCC measure.
- 80% training sets and 20% testing sets (even for non-trainable models). 1000 random such splits.
- Statistical Significance Test:
  - Nonparametric Wilcoxon Rank Sum Test is used.
  - SROCC scores over 1000 splits are used.
  - Null hypothesis is that the median SROCC of the two algorithms is same.
  - 95% significance level is used.
- Box plot of LCC and SROCC over 1000 splits, showing median value and standard deviation, is plotted.
- To test where 2stepQA beats MS-SSIM:

  - Using NIQE, reference images (80) are divided into two equal groups, one with higher quality images and other with lower quality images. The corresponding compressed images are put with their respective reference images.
  - MS-SSIM and 2stepQA both correlated similarly on the subset of high quality reference images.
  - However, on the subset of poor quality reference images, 2stepQA significantly outperformed MS-SSIM. This is because of the contribution of the NR component.

- To highlight the importance of accurate NR algorithms, 2stepQA is also done with using actual MOS of the reference images as the NR score.
- In the general two step model, γ represents the relative contribution of NR component.
- For a fixed NR model, γ is higher for high performing R model and vice-versa.
- The choice of NR algorithm doesn't influence the performance of two step model, as much as the choice of R algorithm.
- Generalized models achieved nearly optimal performance for fixed $\gamma = 0.5$ as well (compared to the case when the model is optimized for γ).
- Performance of two step model as α, γ varies is plotted (two different plots).
- When reference images are of high quality, MS-SSIM performs better than 2stepQA, but 2stepQA is not far behind. But NIQE does not do so well.
- Note that predicted score of 2stepQA model will be close to 1 only when both the reference image and the compressed image are of high quality.
- Different ways of combining R and NR scores have been tried, but multiplication gave best result.
- The problem can be viewed as prediction R quality after compression, given the NR quality measuremtn before compression

62 Event-driven Video Frame Synthesis

Paper Details

- Authors: North Western University

Abstract

- Code and demo: https://github.com/winswang/int-event-fusion/tree/win10

62.1 My Summary

- Synthesizes intermdediate frames to produce high frame rate video that can capture high speed events
- Denoises videos at the end
- Video Prediction: Based on past 2 frames and future events, predicts the next frame.
- Evaluation metrics: PSNR, SSIM

63 From Here to There: Video Inbetweening Using Direct 3D Convolutions

Abstract

- Datasets: BAIR Robot pushing, KTH, UCF101
- Generates 14 in-between frames
- Conditioned on only 1 past and 1 future frame. Total 16 frames.
- Frames downsampled and cropped to 64x64

64 Physics-as-Inverse-Graphics: Joint Unsupervised Learning of Objects and Physics from Video

Abstract

- Datset: 2D circles, digits moving.

65 Scaling Autoregressive Video Models

Paper Details

- Authors: Dirk Weissenborn
- Institutions: Google Research
- Project website: https://sites.google.com/view/video-transformer-samples
- Code:
- Published in: 2019
- Citations: 2 (As of 20-10-2019)

Abstract

65.1 Introduction

- Motivation:
- Contributions:
  - –
- Auto-regressive models generate videos pixel by pixel.

65.2 Related Work

65.3 Video Transformer

- Loss Functions:

65.4 Experiments

- Datasets: BAIR, Kinetics-600, Moving MNIST, PUSH
- Baselines:
- Evaluation Metrics: SSIM, FVD

66 VideoFlow: A Flow-Based Generative Model for Video

Paper Details

- Authors: Manoj Kumar, Mohammad Babaeizadeh, Chelsea Finn, Sergey Levine, Durk Kingma
- Institutions: Google Brain, University of Illinois
- Project website: https://sites.google.com/view/videoflow/home
- Code: https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/video/next_frame_glow.py
- Published in: 2019
- Citations: 15 (As of 22-10-2019)

Abstract

- A model for video prediction based on normalizing flows, which allows for direct optimization of the data likelihood, and produces high-quality stochastic predictions.

66.1 Introduction

- Motivation:
- Contributions:
  - –

66.2 Related Work

66.3 Preliminaries: Flow-Based Generative Models

- Loss Functions: log-likelihood

66.4 Proposed Architecture

66.5 Quantitative Experiments

- Datasets: Stochastic Movement Dataset, BAIR
- Baselines: SAVP-VAE, SV2P
- Evaluation Metrics: PSNR, SSIM, VGG cosine similarity (best out of 100)
- On a temporal patch of 4 frames, the log-likehood of 4th frame given the context of 3 previous frames is maximized.

66.6 Qualitative Experiments

- 100 frames are predicted.

67 Probabilistic Video Prediction from Noisy Data with a Posterior Confidence (BP-Net)

Paper Details

- Authors: Yunbo Wang, Joshua B. Tenenbaum
- Institutions: Stanford, MIT
- Project website:
- Code:
- Published in: CVPR 2020
- Citations: 0 (As of 09-04-2020)

Abstract

67.1 Introduction

- Motivation:
  - Precipitation forecasting
  - Traffic flows prediction
  - Robotics
- Contributions:
  -

67.2 Related Work

67.3 Method

- Loss Functions:

67.4 Experiments

- Datasets: Moving MNIST, KTH
- Baselines: DFN, FRNN, PredRNN++, VideoGAN, MCnet, SVG-LP
- Evaluation Metrics: MSE, SSIM over 100 predictions (best and worst)

68 Learning Human Objectives by Evaluating Hypothetical Behavior

Paper Details

- Paper: arXiv
- Authors: Sergey Levine
- Institutions:
- Project website:
- Code:
- Published in: ICML 2020
- Citations: (As of 21-06-2020)

Abstract

68.1 Introduction

- Motivation:
- Contributions:
  - –

68.2 Related Work

68.3 Method

- Loss Functions:

68.4 Experiments

- Datasets:
- Baselines:
- Evaluation Metrics:

68.5 My Summary

- Setup:
  - There is an RL agent that learns to take actions to solve a task.
  - The rewards to the RL agent is generated by a neural network.
  - The reward generating neural network is trained with data containing (state, action, reward-label) tuples.
  - A generator generates trajectories that are shown to user who labels them and then are used to train the reward generating neural network.
- The goal of this work is to train a better reward generating neural network.
- An acquisition function is used to generate more useful trajectories. The trajectories are predicted which
  - maximize uncertainty in the reward generated by the neural network.
  - maximizes the reward.
  - minimizes the reward.

  - maximizes the novelty of trajectories (diversity).

- Because of this acquisition function, good training data is generated which in turn leads to faster convergence of reward generating neural network.