Final Master Thesis
Master on Artificial Intelligence

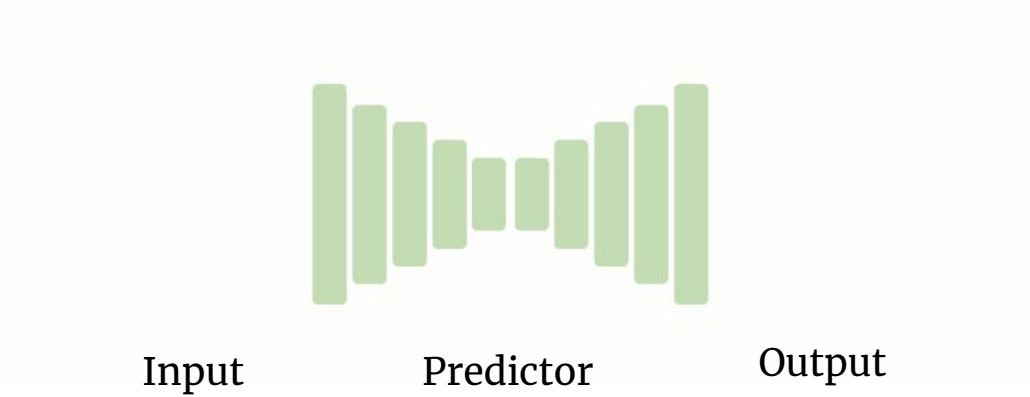
A Comprehensive Survey on Deep Future Frame Video Prediction

by
Javier Selva Castelló

Supervised by **Sergio Escalera Guerrero** and **Marc Oliu Simón**

Future Frame Prediction

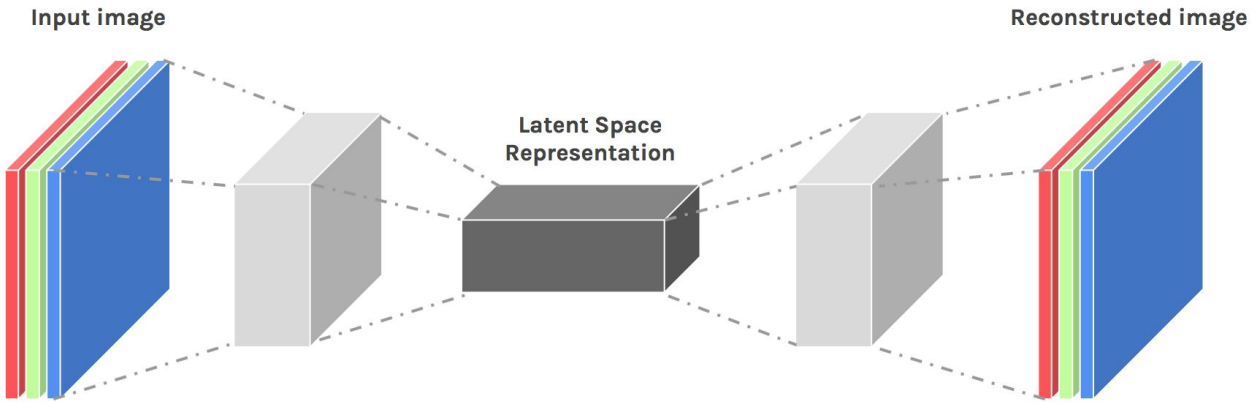
Given a video sequence, generate the next frames.



Background

Unsupervised learning based on **autoencoders**:

- **Generative** models.



Source: Understanding Autoencoders

Learning Unsupervised Features

- Using Temporal Information:
  - Movement dynamics → Relative features.
  - Better learn visual features.
  - Invariance to light, rotation, occlusion.

- Predictive Coding:
  - Neuroscience Theory of the Brain
  - Always generating predictions.
  - Compares predictions against sensory input.
  - Use difference to learn better models of the world.

Applications

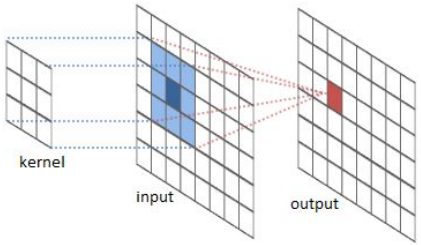
- Unsupervised learning:
  - **Early behaviour detection** & understanding:
    - Falls in elderly people.
    - Robbery or aggression.
  - **Planning for agents**:
    - Interaction with environment.
    - Autonomous cars.
- Video processing:
  - Compression.
  - Slow motion.
  - Inpainting.

Structure of the Presentation

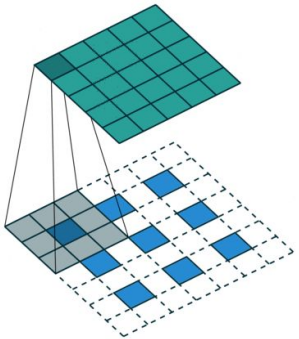
- Fundamentals.
- Training techniques.
- Loss functions.
- Measuring prediction error.
- Models and main trends.
- Experiments.
- Results.
- Discussion.
- Conclusions and future work.

Fundamentals (I)

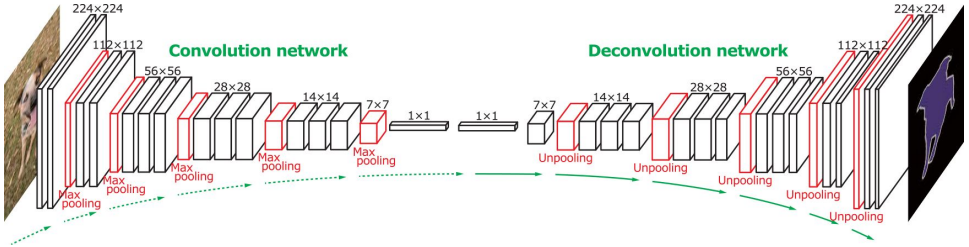
Convolutional Neural Networks (CNN)



Convolution [1]



Deconvolution [2]

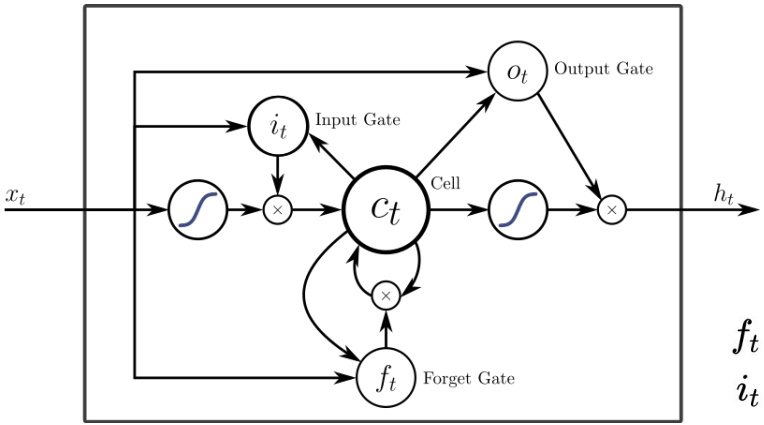


Convolutional Autoencoder with Pooling layers [3]

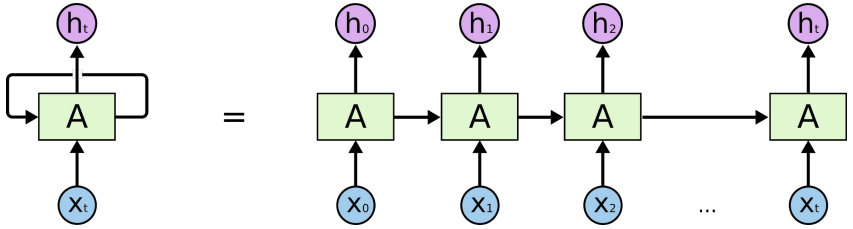
[1] Intel Labs, *Bringing Parallelism to the Web with River Trail,* http://intellabs.github.io/RiverTrail/tutorial/
[2] Vincent Dumoulin, *Convolutional Arithmetics*: https://github.com/vdumoulin/conv_arithmetic
[3] H. **Noh**, S. Hong, and Bohyung Han. *Learning deconvolution network for semantic segmentation.* ICCV (2015).

Fundamentals (II)

Long Short-Term Memory (LSTM)



An LSTM cell [1]



Unrolled LSTM Network [2]

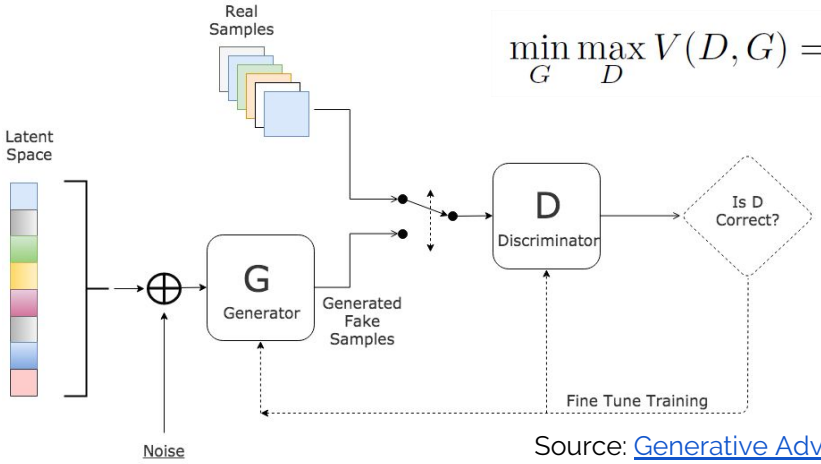
$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f c_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i c_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o c_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

[1] A. **Graves**, A. R. Mohamed, and G. Hinton. Wikimedia commons: *Peephole long short-term memory*, 2017.
[2] C. **Olah**. Understanding LSTM networks, 2015.

Fundamentals (III)

Generative Adversarial Networks (GAN)[1]

- Generative network produces samples. (G)
- Discriminative network classifies real from generated samples. (D)



$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Source: Generative Adversarial Networks

Source: HeuriTech Blog

[1] I. **Goodfellow**, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. *Generative adversarial nets*. In NIPS, 2014.

Improved Training

- Curriculum learning:
  - Model learns to generate short sequences first.
  - Then it is progressively fine-tuned for longer predictions.
- Pretrain for reconstruction:
  - First train the model for sequence reconstruction.
  - Then fine-tune for future frame prediction.
- Feedback Predictions:
  - Many models use past predictions as input during **test** time.
  - **Train** the model to predict based on previously generated frames.
  - Model more robust to own errors. Avoids propagating mistakes.

Loss Functions

Distance Losses (Blurry)

$$L_2(y_i, t_i) = \sum_{i=0}^{n} (y_i - t_i)^2$$

$$MSE(y_i, t_i) = \frac{1}{n} \sum_{i=0}^{n} (y_i - t_i)^2$$

$$ED(y_i, t_i) = \sqrt{\sum_{i=0}^{n} (y_i - t_i)^2}$$

Adversarial to ensure sharp predictions.

Other Common Losses

$$CE(y_i, t_i) = \frac{1}{n} \sum_{i=0}^{n} t_i \log y_i + (1 - t_i) \log (1 - y_i)$$

Gradient Difference Loss (GDL)

$$\mathcal{L}_{gdl}(\mathbf{y}, \mathbf{z}) = \sum_{i,j}^{h,w} | \left(|\mathbf{y}_{i,j} - \mathbf{y}_{i-1,j}| - |\mathbf{z}_{i,j} - \mathbf{z}_{i-1,j}|\right) |^{\lambda} + | \left(|\mathbf{y}_{i,j-1} - \mathbf{y}_{i,j}| - |\mathbf{z}_{i,j-1} - \mathbf{z}_{i,j}|\right) |^{\lambda}.$$

Measuring Results

From a given sequence and correct movement dynamics, **multiple futures are possible**.

Compare against Ground Truth

- Mean Squared Error
- Peak Signal to Noise Ratio
- Structural Similarity
- Structural Dissimilarity

Realistic looking sequences

Inception Metric

- Train a traditional classifier.
- Measure accuracy with predicted sequences.

Human Evaluation

- “Which sequence do you prefer?”

Application for other tasks

- Weather prediction.
- Improved planning for a system playing Atari Games. [1]
- Emulate video-game. [1]
- Fine-tune the model for:
  - Action Classification.
  - Optical flow estimation.

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

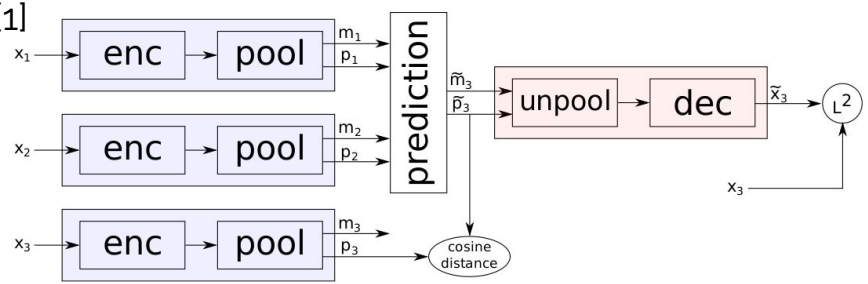
Models

- Simple non-recurrent proposals.
- Use input to generate prediction filters:
  - Non-recurrent.
  - Recurrent.
- Predict using basic element other than frames.
- Explicit separation of content and motion.
- Models for the experiments.
- Others.

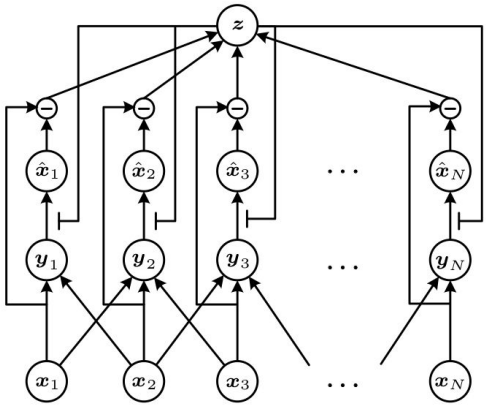
Models (I)

Simple non-recurrent proposals

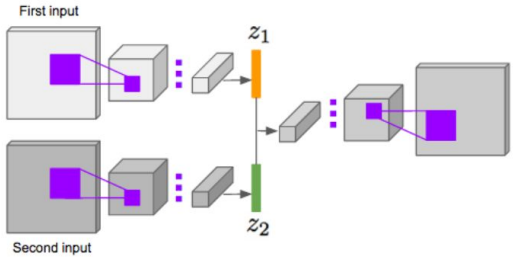
[1]



[2]



[3]

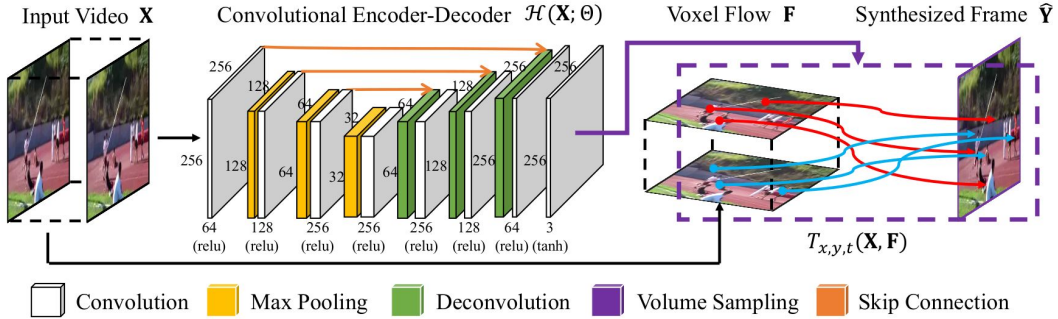


[1] R. **Goroshin**, M. **Mathieu**, and Y. LeCun. *Learning to linearize under uncertainty.* NIPS 2015.
[2] M. **Zhao**, C. Zhuang, Y. Wang, and T. Sing Lee. *Predictive encoding of contextual relationships for perceptual inference, interpolation and prediction.* In ICLR'15, 2014.
[3] Y. **Zhou** and T. L. Berg. *Learning temporal transformations from time-lapse videos.* In ECCV, 2016.

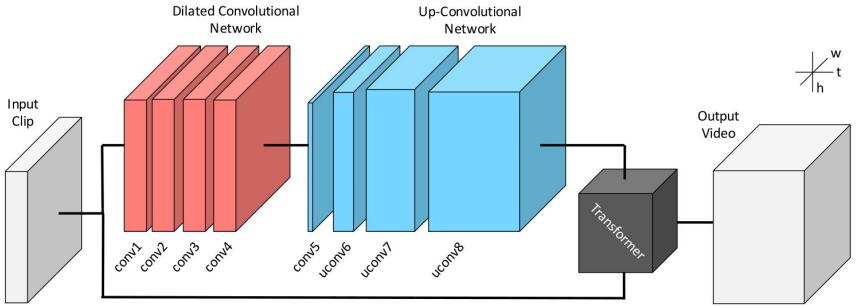
Models (II)

Predict filter which is applied to last input frame(s)

[1]



[2]

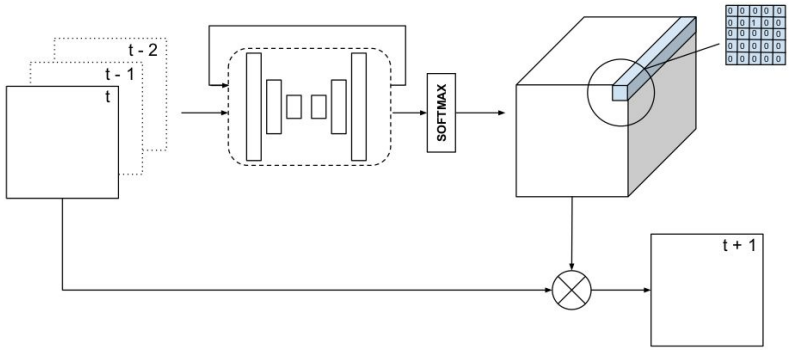


[1] Z. **Liu**, R. Yeh, X. Tang, Y. Liu, and A. Agarwala. *Video frame synthesis using deep voxel flow.* In ICCV, 2017.
[1] C. **Vondrick** and A. Torralba. *Generating the future with adversarial transformers*, 2017.

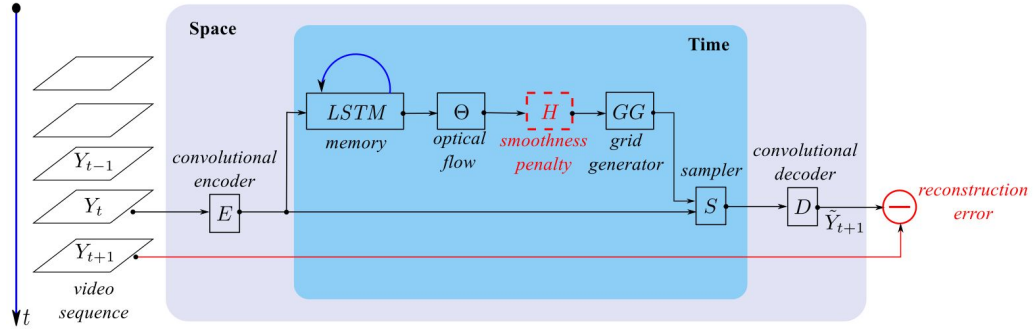
Models (III)

Predict filter which is applied to last input frame(s) (recurrent)

[1]



[2]

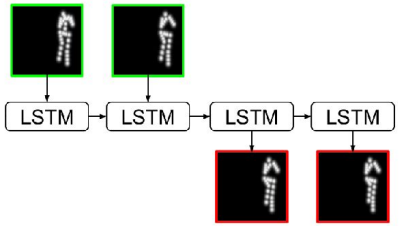


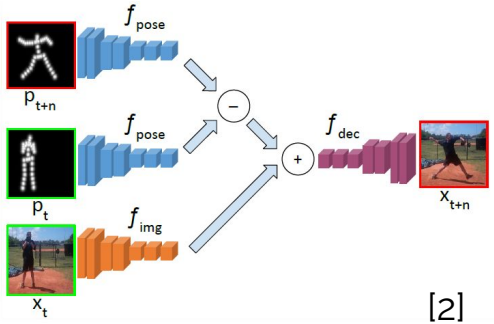
[1] B. De **Brabandere**, X. **Jia**, T. Tuytelaars, and L. Van Gool. *Dynamic filter networks.* In NIPS, 2016.
[2] V. **Pătrăucean**, A. Handa, and R. Cipolla. *Spatio-temporal video autoencoder with differentiable memory.* ICLR Workshop, 2016.

Models (IV)

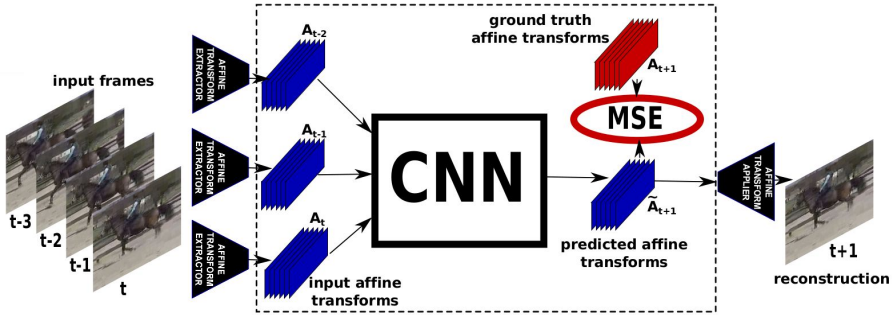
Predict at some feature level, then generate future frame.

[1]





[2]

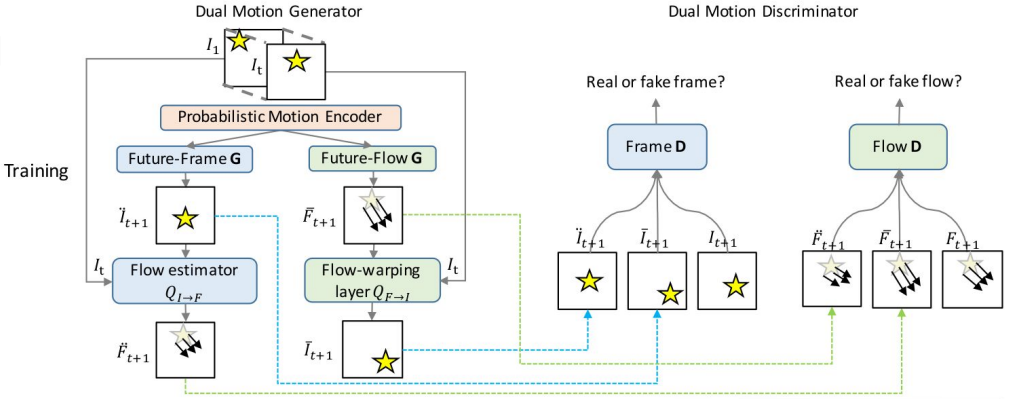


[1] R. **Villegas**, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee. *Learning to generate long-term future via hierarchical prediction.* 2017.
[2] J. R. van **Amersfoort**, A. Kannan, M.'A. Ranzato, A. Szlam, D. Tran, and S. Chintala. *Transformation-based models of video sequences.* CoRR, 2017.

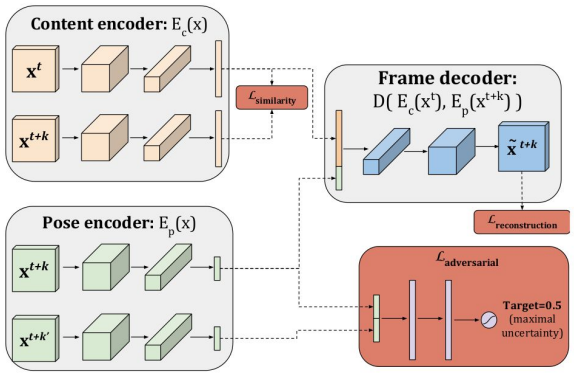
Models (V)

Explicit separation of content and motion.

[1]



[2]

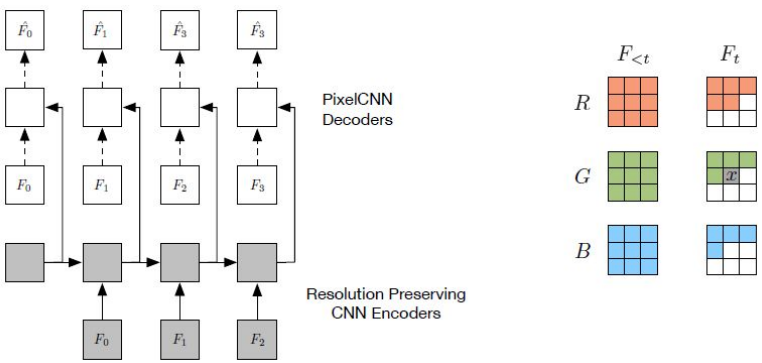


[1] X. **Liang**, L. Lee, W. Dai, and E. P. Xing. *Dual motion gan for future-flow embedded video prediction.* In ICCV, 2017.
[2] E. L. **Denton** and V. Birodkar. *Unsupervised learning of disentangled representations from video.* In NIPS, 2017.

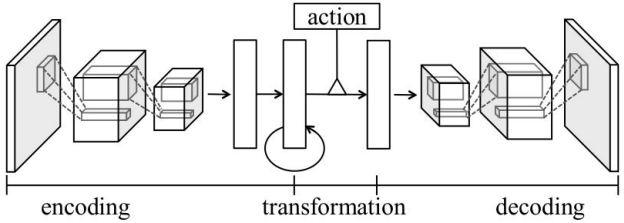
Models (VI)

Others.

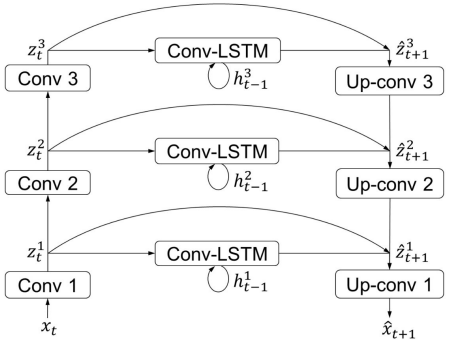
[1]



[2]



[3]



[1] N. **Kalchbrenner**, A. van den Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. *Video pixel networks.* CoRR, 2016.
[2] J. **Oh**, X. Guo, H. Lee, R. L. Lewis, and S. Singh. *Action-Conditional Video Prediction using Deep Networks in Atari Games.* In NIPS, 2015.
[3] F. **Cricri**, X. **Ni**, M. Honkala, E. Aksu, and M. Gabbouj. *Video ladder networks.* CoRR, 2016.

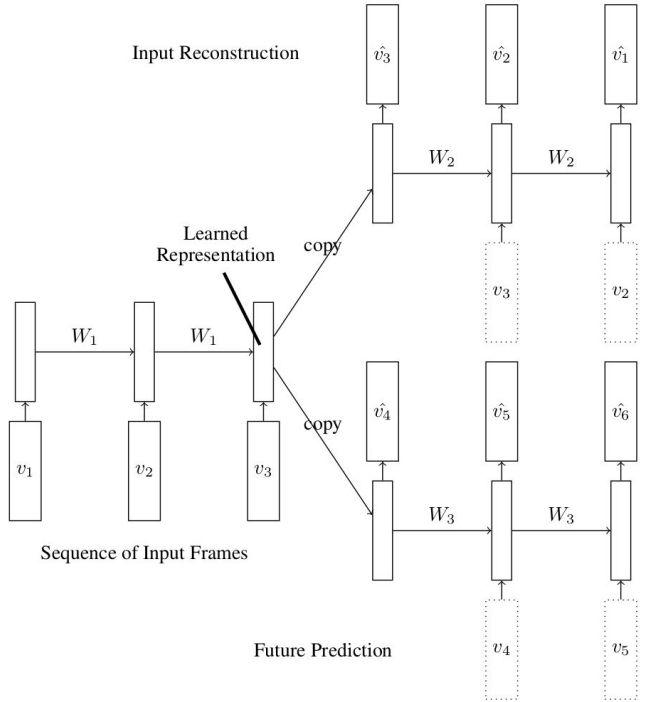
Tested Models

- Deep architectures.
- Ability to work with varying number of frames.
- Complexity of design enough to handle the proposed datasets.
- Code available online.
- Implementation adaptable to the experiments.

Tested Model (I)

Srivastava

- Recurrent Model.
- Fully Connected LSTM AE.
- Independent encoder-decoder:
  - Unroll encoder on whole input.
  - Unroll decoder to generate predictions.
- L2 reconstruction loss.

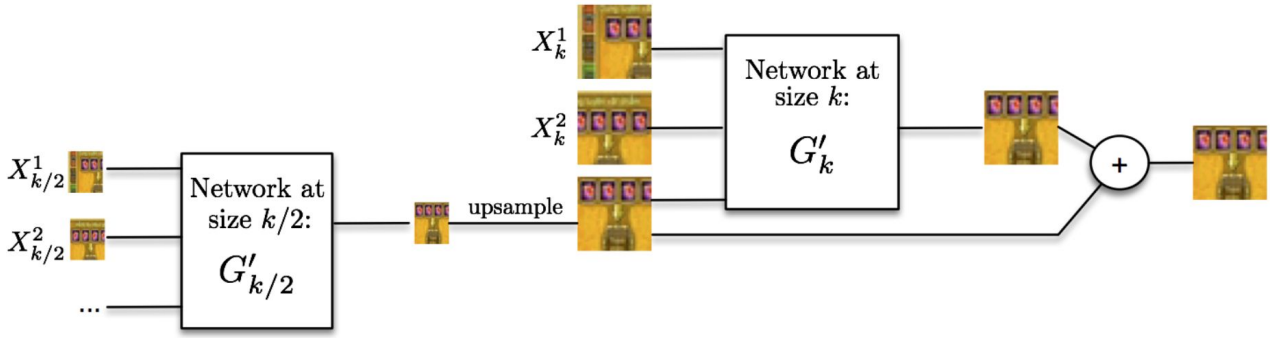


[1] N. **Srivastava**, E. Mansimov, and R. Salakhudinov. *Unsupervised learning of video representations using LSTMs.* In ICML, 2015.

Tested Model (II)

Mathieu

- Non-Recurrent Model.
- Multi scale CNN.
- Inputs and outputs volumes of frames.
- L2, Adversarial and GDL.

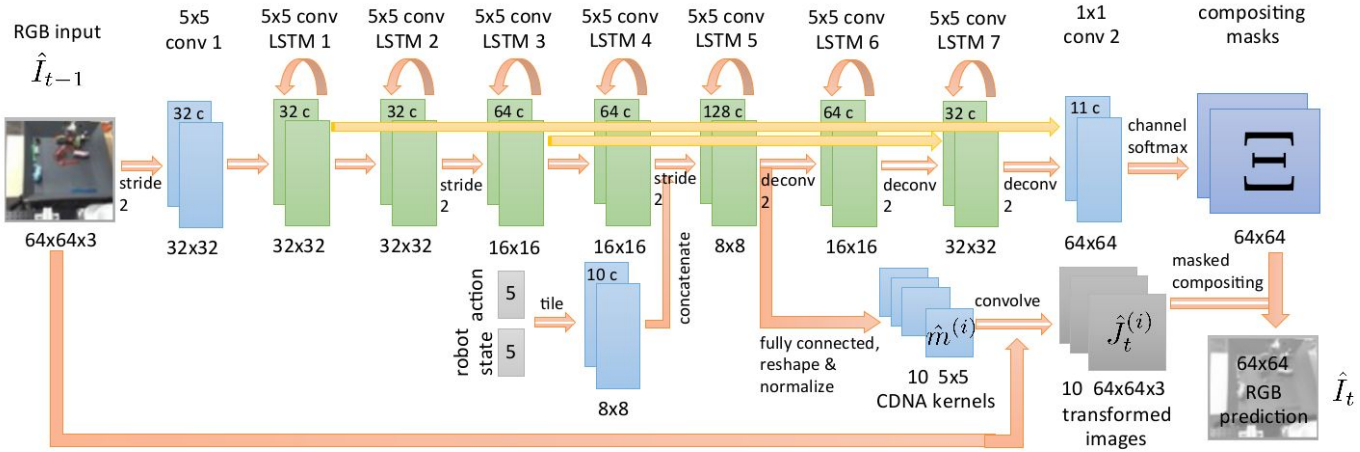


[1] M. **Mathieu**, C. Couprie, and Y. LeCun. *Deep multi-scale video prediction beyond Mean Square Error.* 2015.

Tested Model (III)

Finn

- Recurrent Model.
- Convolutional LSTM AE.
- Predicts patch transformations.
- Dynamic masks for applying transforms at pixel level.
- Explicit foreground/background separation.
- Allows for hallucinating new pixels.
- Pixel distance and GDL.

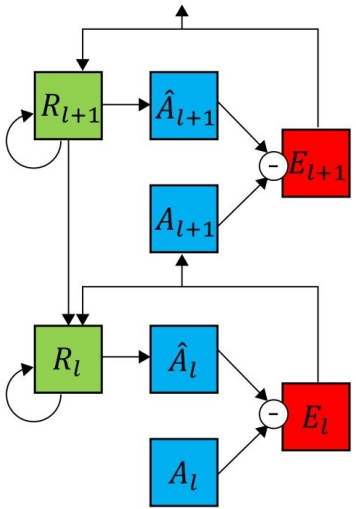


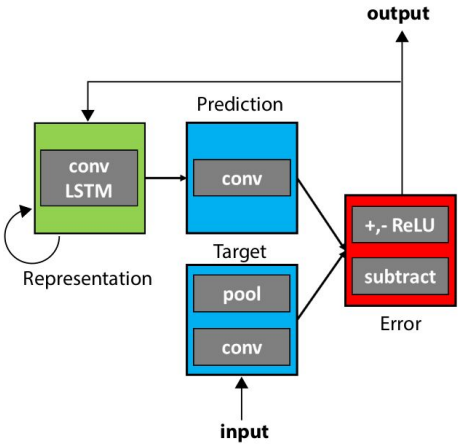
[1] C. **Finn**, I. Goodfellow, and S. Levine. *Unsupervised learning for physical interaction through video prediction.* In NIPS, 2016.

Tested Model (IV)

Lotter

- Recurrent Model.
- Convolutional LSTM.
- Each layer tries to fix previous layer mistakes.
- Two step execution:
  - Top-down pass to update predictor state.
  - Bottom-up pass to update predictions, errors and targets.



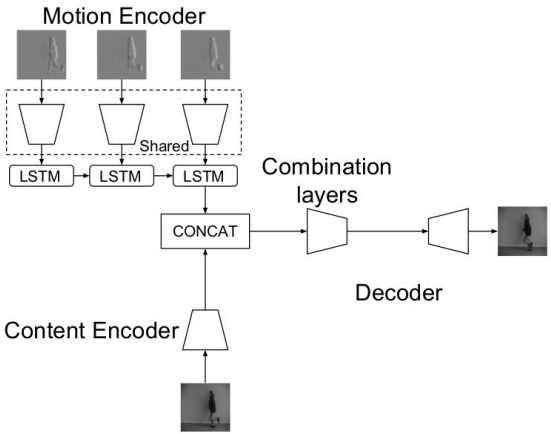


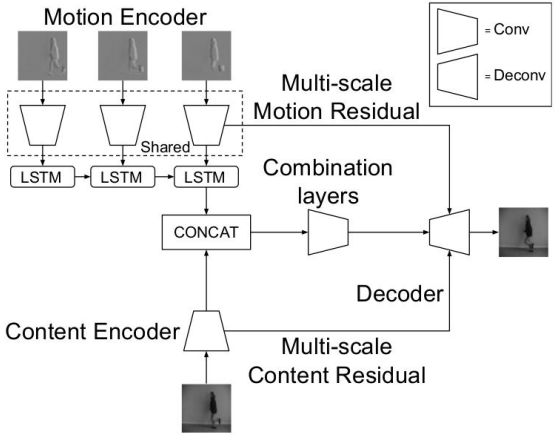
[1] W. **Lotter**, G. Kreiman, and D. Cox. *Deep predictive coding networks for video prediction and unsupervised learning*. ICLR, 2016.

Tested Model (V)

Villegas

- Recurrent model.
- Autoencoder with residual connections.





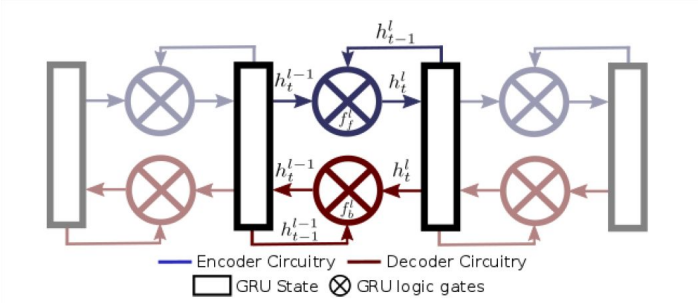
- Separate input:
  - Difference images through CNN + LSTM (Motion).
  - Single static frame through CNN (Content).
- They used a fused loss with L2, Adversarial and GDL.

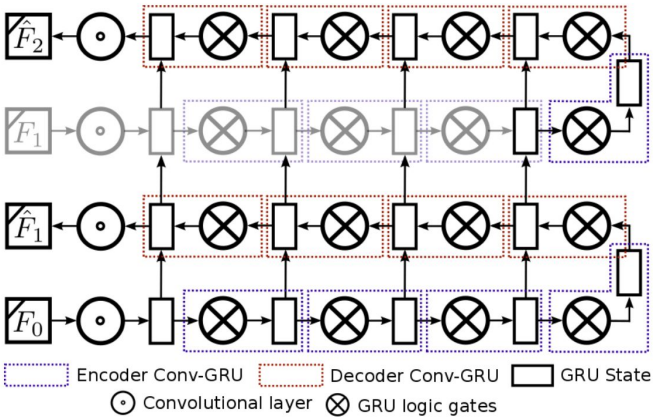
[1] R. **Villegas**, J. Yang, S. Hong, X. Lin, and H. Lee. *Decomposing motion and content for natural video sequence prediction*. 2017.

Tested Model (VI)

Oliu

- Recurrent model.
- Conv. GRU AE-like architecture with shared weights:
  - Unroll encoder to take all input sequence.
  - Unroll decoder to generate whole predicted sequence.
- They used a simple L1 loss.





[1] M. **Oliu**, J. Selva, and S. Escalera. *Folded recurrent neural networks for future video prediction,* 2017.

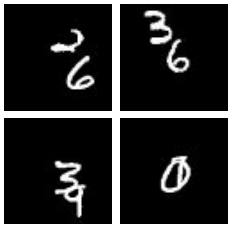
Experimental Setting

- Use **10** frames as input **to** predict future **10** frames.
- Used implementations adapted to use specific sampling:
  - Take random subsequence during train.
  - Slide over all possible sequences for testing.
- Three datasets with increasing complexity.
- Measure results quantitatively with **MSE**, **PSNR** and **DSSIM**.

Datasets (I)

Moving MNIST[1]

- **64 x 64** (grayscale)
- Generated randomly.
- **Train**: 1M seq. **Test**: 10K seq.
- Simple motion dynamics, occlusion, separate objects.

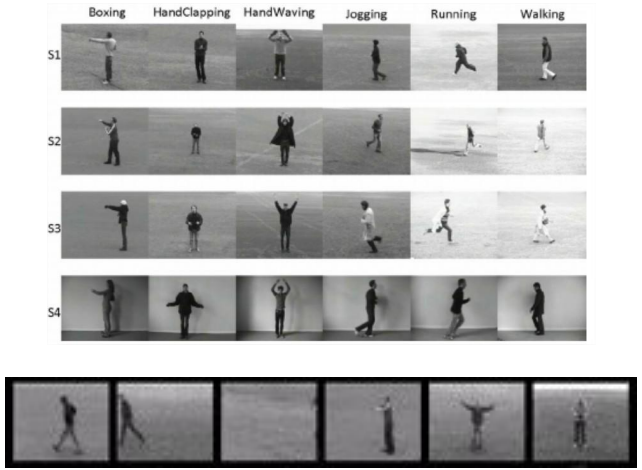


[1] N. **Srivastava**, E. Mansimov, and R. Salakhudinov. *Unsupervised learning of video representations using LSTMs.* In ICML, 2015.

Datasets (II)

KTH [1]

- 25 subjects performing **6 actions** in 4 different settings.
- 120 x 160 (cropped and resized to **64 x 80**) and grayscale.
- **Train**: 383 seq. **Test**: 216 seq.
- Complex human motions, static background.

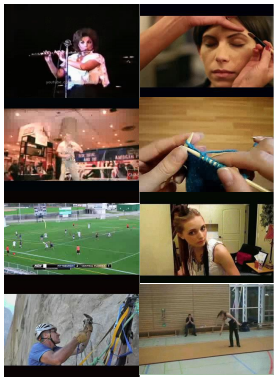


[1] C. **Schuldt**, I. Laptev, and B. Caputo. *Recognizing human actions: A local SVM approach.* In ICPR, 2004.

Datasets (III)

UCF101[1]

- Videos of humans performing **101 different actions**.
- Objects and humans interacting in different ways.
- 240 x 320 x 3 (cropped and resized to **64 x 85 x 3**).
- Frame rate halved to increase motion between frames.
- Most complex case with varying background, objects and camera motion.
- **Train**: 9950 seq.
  Test: 3361 seq.



[1] K. **Soomro**, A. R. Zamir, and M. Shah. *UCF101: Action Recognition dataset*, 2011.

Quantitative Results (I)

Model	Moving MNIST			KTH			UCF101		
	MSE	**PSNR**	**DSSIM**	**MSE**	**PSNR**	**DSSIM**	**MSE**	**PSNR**	**DSSIM**
Srivastava	0.01737	18.183	0.08164	0.00995	21.220	0.19860	0.14866	10.021	0.42555
Mathieu	0.03071	15.361	0.32770	0.00194	29.097	0.10018	0.01287	20.492	0.20730
Lotter	0.04137	14.017	0.14201	0.00807	24.635	0.13588	0.02124	20.398	0.19013
Finn	**0.00561**	**22.986**	0.04985	**0.00065**	**36.101**	**0.02790**	0.00940	23.601	0.15196
Villegas	0.04254	13.857	0.13896	0.00165	30.946	0.07657	0.00940	23.457	0.14150
Oliu	0.00947	21.386	**0.04376**	0.00175	29.299	0.07251	**0.00908**	**23.872**	**0.13055**

- DSSIM is more related to qualitative results. MSE and PSNR regard blurry predictions as good.
- **Finn** seems to perform better for static backgrounds. Only worked for square videos.
- **Lotter** and **Villegas** were not able to learn an initial representation for Moving MNIST.
- The fully connected model by **Srivastava** needed too many parameters for KTH and UCF101.
- **Oliu** and **Villegas** present more balanced results over the different datasets.

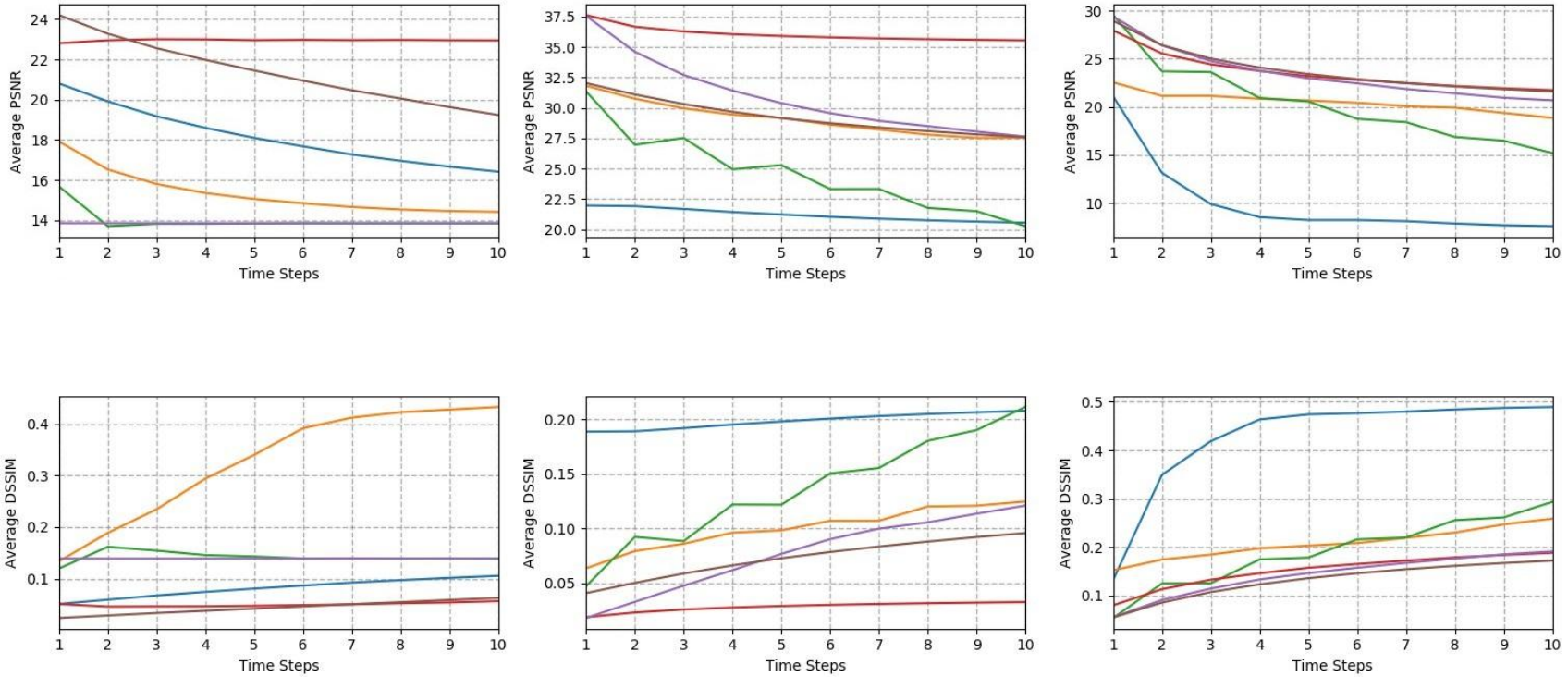
Moving MNIST

KTH

UCF101

PSNR

DSSIM



Qualitative MMNIST Results (I)

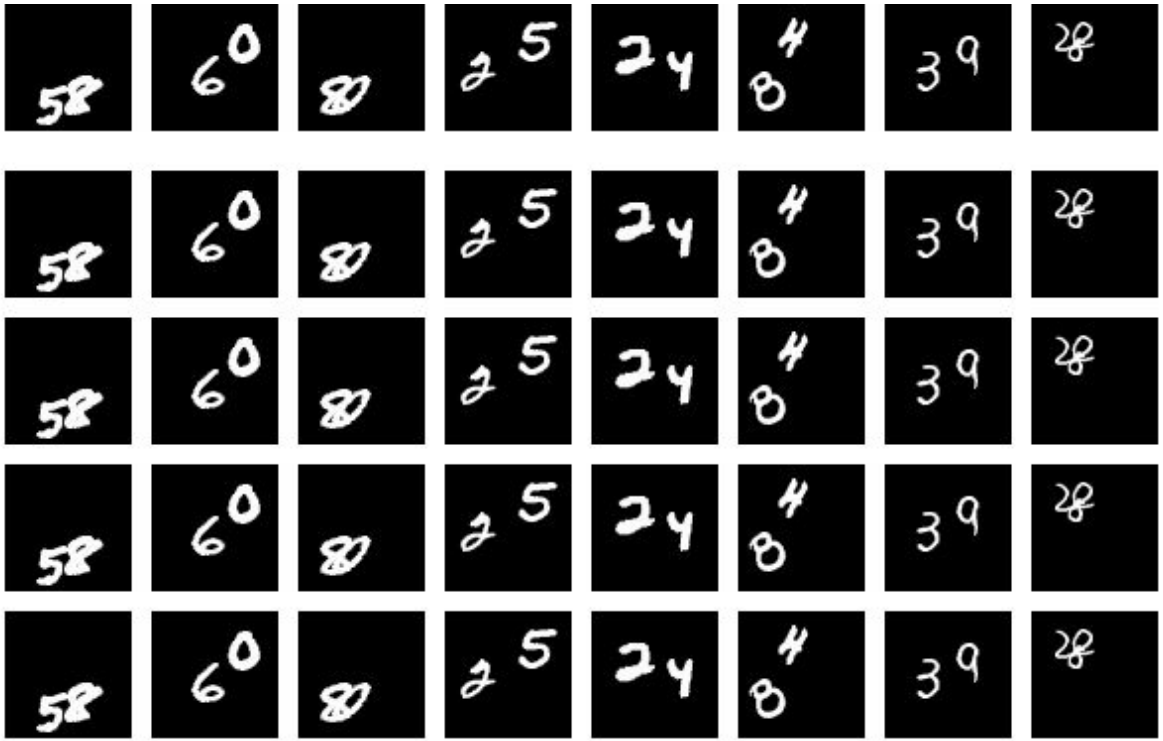
5 frames input
10 Ground Truth

Srivastava

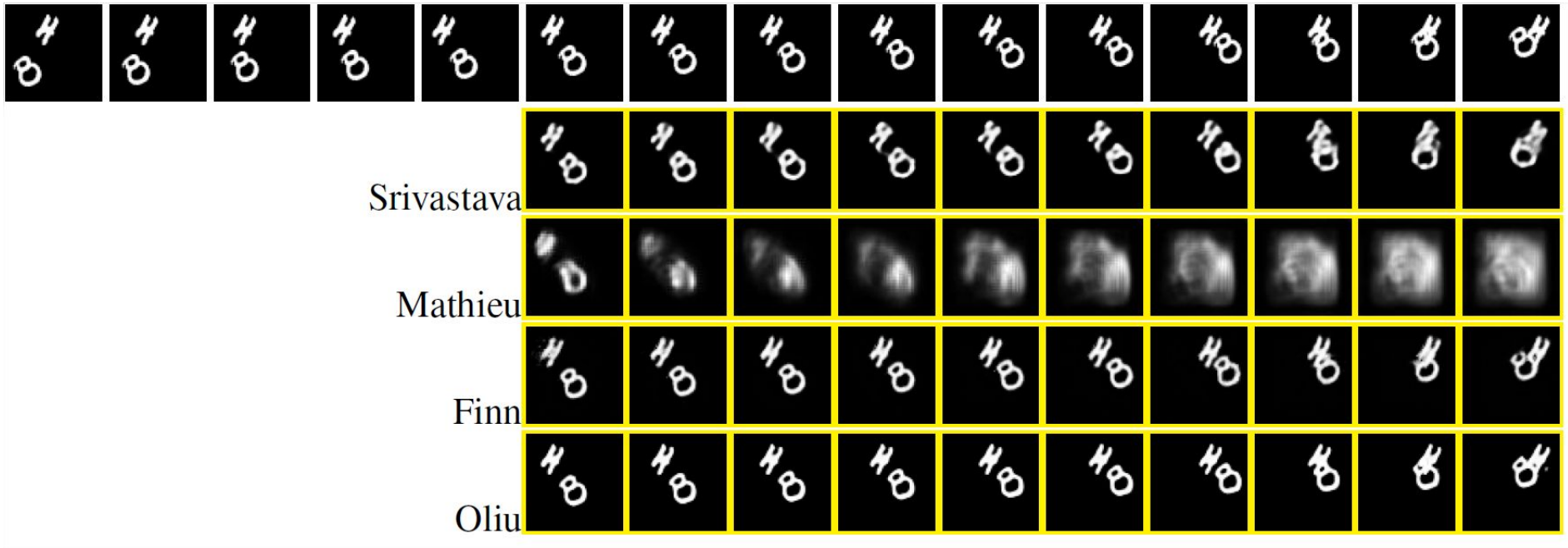
Mathieu

Finn

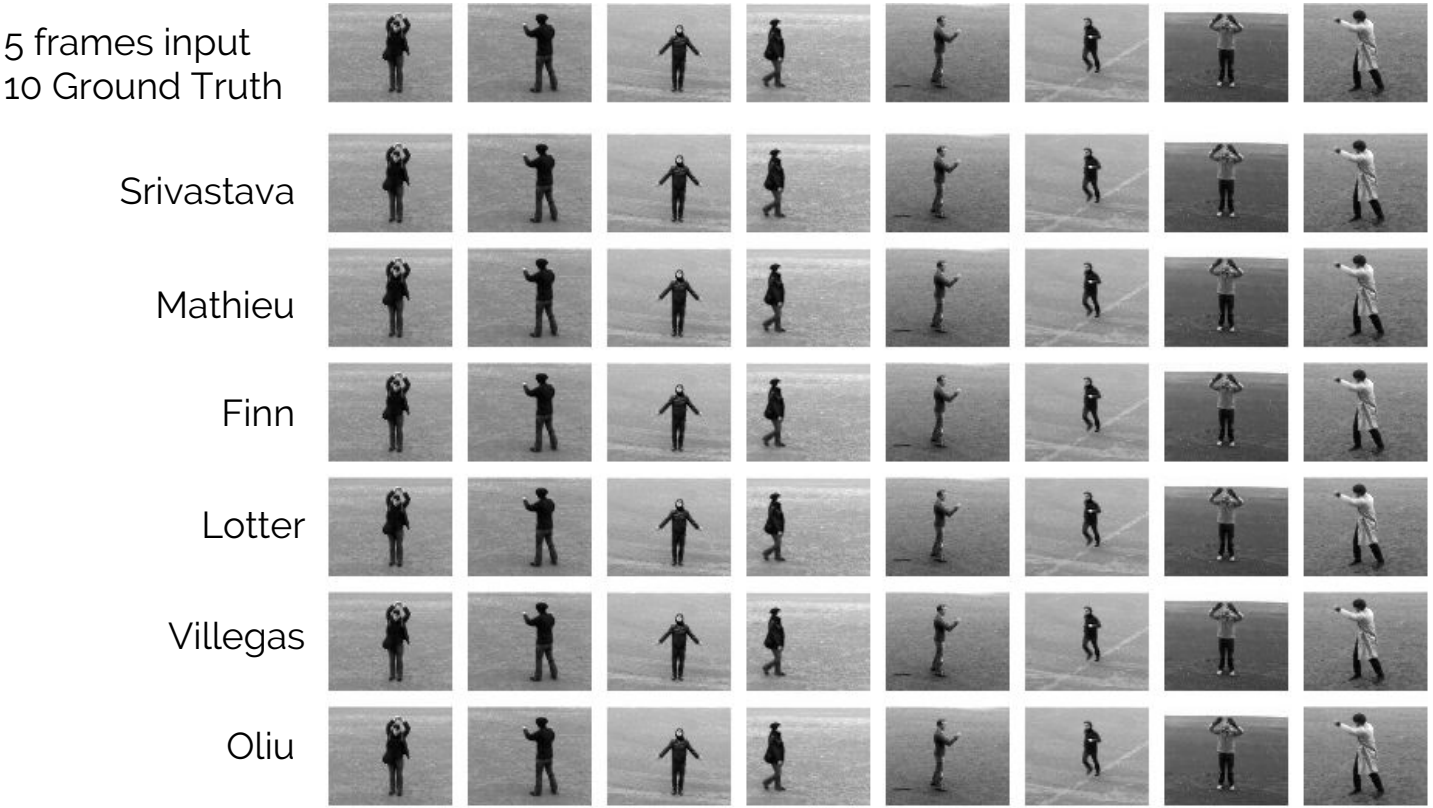
Oliu



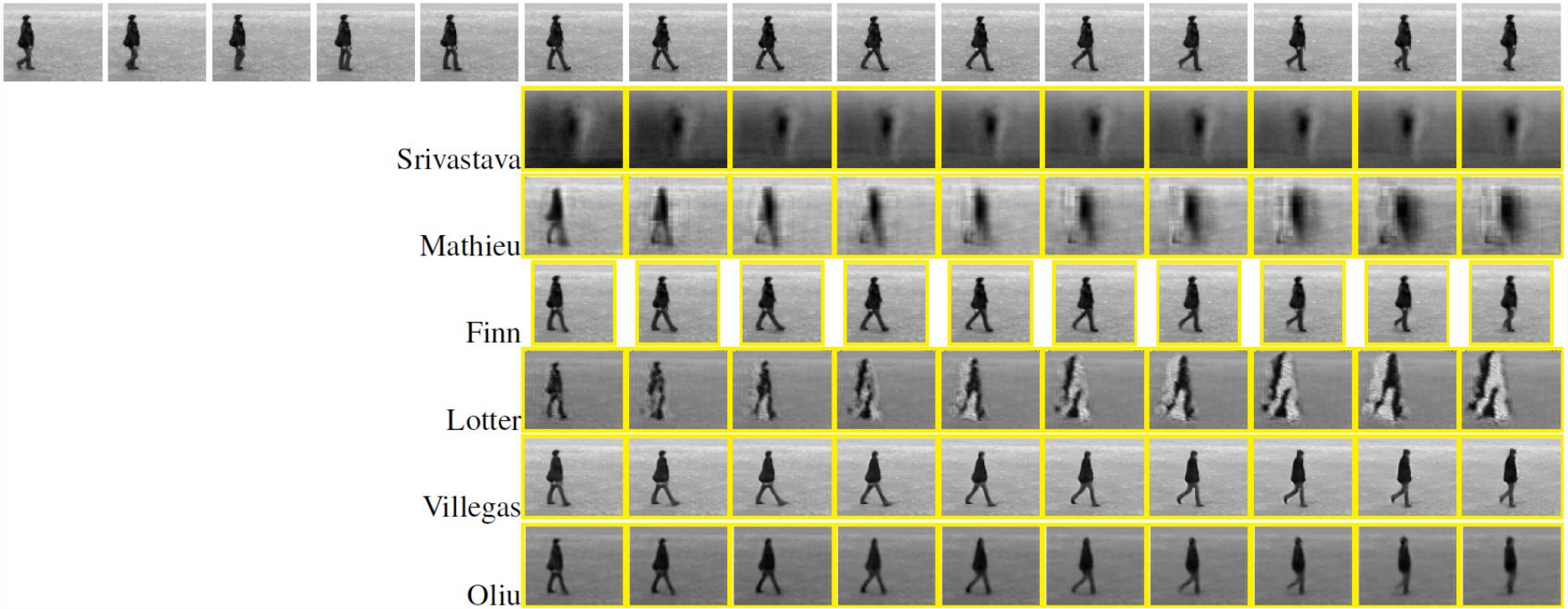
Qualitative MMNIST Results (II)



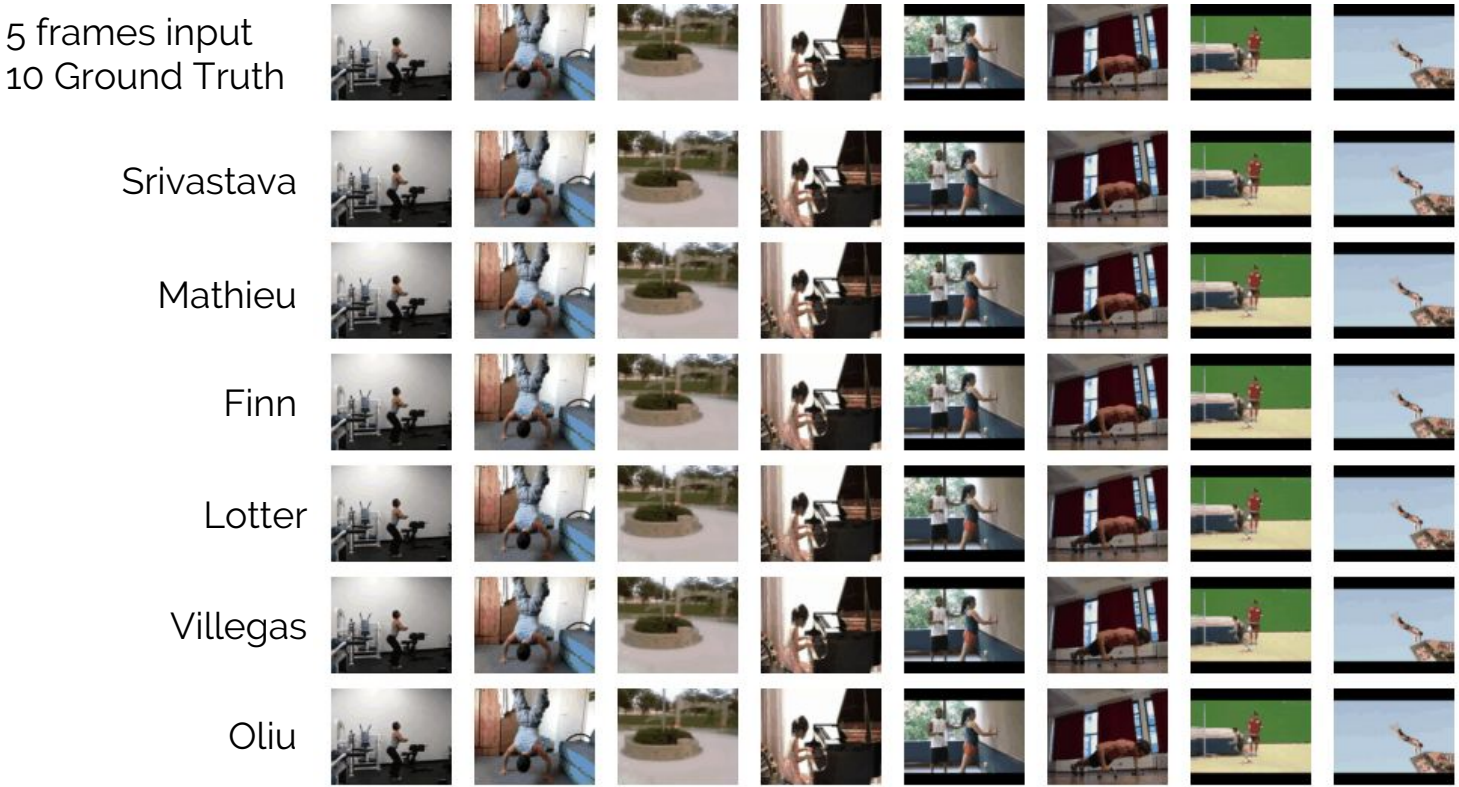
Qualitative KTH Results (I)



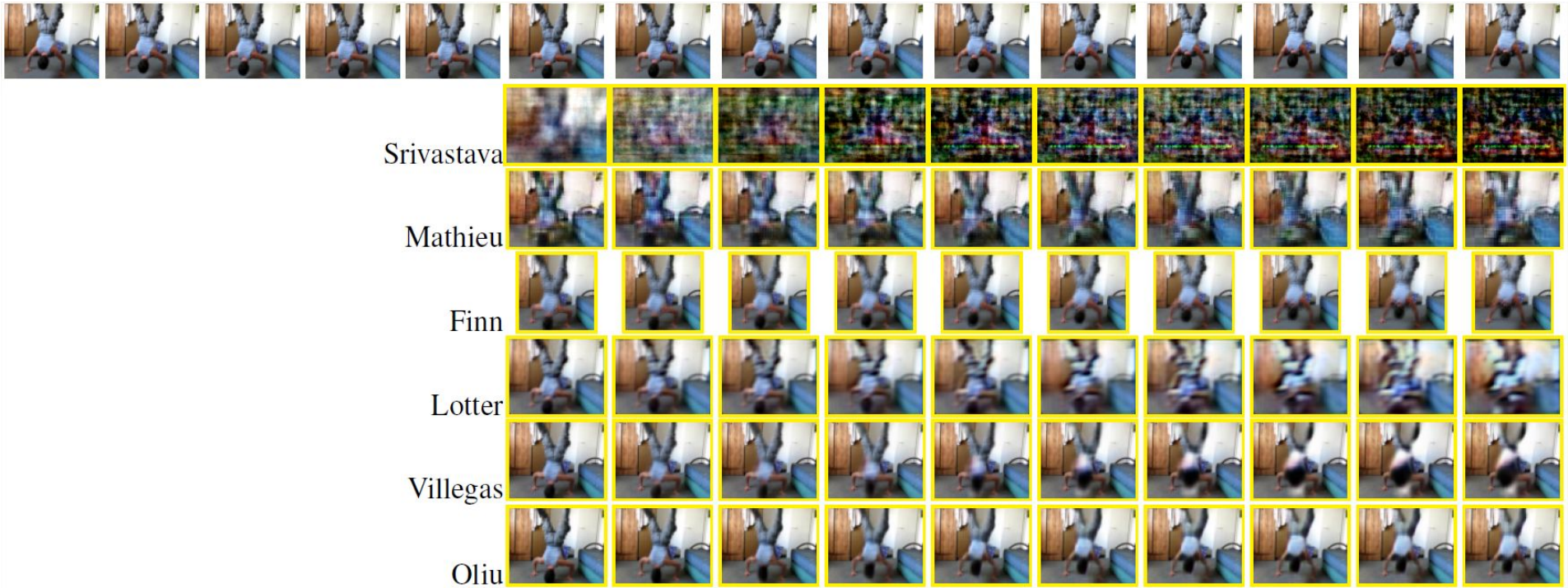
Qualitative II Results (II)



Qualitative UCF101 Results (I)



Qualitative UCF101 Results (II)



Discussion

- Use a **metric** that **regards the structure** of the image.
- **Residual connections** → **Feature hierarchy**.
- **Feedback** during train → Models **robust** to errors.
- **Predict further without** them → Consistent sequences.
- **Separate content and motion** → Focus learning efforts.
- **Incremental learning** → Improves learning.
- **Multi scale** → Not aparent impact.
- Pixel difference losses are not enough:
  - **Adversarial** produces sharp results, but not better predictions.
  - **GDL** reduces artifacts.

Conclusion

- Task of future frame prediction has been **presented**.
- Different **trends** for solving the problem.
- Specific models have been **tested and compared**.
- Results of the experiments **analysed**.
- **Discussion** for the different approaches.
- Related **publications**:
  - **CVPR'18** submission. [1]
  - Springer Book Chapter

Future work:

- Need for a **proper evaluation** metric.
- **Design** and build predictive model.
- Separately **test different variables**.
- Change **hyperparameters** of tested models.

[1] M. **Oliu**, J. Selva, and S. Escalera. *Folded recurrent neural networks for future video prediction,* 2017.

Thank you!