

Deep Learning Approaches to Predict Future Frames in Videos

BY

Abdullah Al Mukul

ID: 181-15-1931

AND

Mahtab Mashuq Tonmoy

ID: 181-15-2079

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Amit Chakraborty Chhoton

Sr. Lecturer

Department of CSE

Daffodil International University

Co-Supervised By

Md. Sabab Zulfiker

Sr. Lecturer

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2023

APPROVAL

This Project titled “Predict Future Frames in Videos”, submitted by Abdullah Al Mukul (ID:181-15-1831) and Mahtab Mashuq Tonmoy (ID:181-15-2079) to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 06 February 2023.

BOARD OF EXAMINERS

Chairman

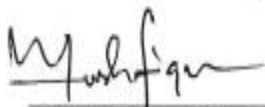
Dr. Tuhid Bhuiyan
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Arif Mahmud
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Mr. Mushfiqur Rahman
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

External Examiner



Dr. Mohammad Shorif Uddin
Professor
Department of Computer Science and Engineering
Jahangirnagar University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Amit Chakraborty Chhoton, Sr. Lecturer, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



05-02-2023

Amit Chakraborty Chhoton
Sr. Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:

Md. Sabab Zulfiker
Sr. Lecturer
Department of CSE
Daffodil International University

Submitted by:



Abdullah Al Mukul
ID: 181-15-1931
Department of CSE
Daffodil International University



Mahtab Mashuq Tonmoy
ID: 181-15-2079
Department of CSE
Daffodil International University

©Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We are really grateful and wish our profound indebtedness to **Amit Chakraborty Chhoton, Sr. Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of Image Processing to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Md. Sabab Zulfiker, Lecturer** and Head, Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Across several fields of computer vision, deep learning methods are gaining importance. While there have been several research on the categorization of pictures and movies, prediction of the coming frames of an input sequence in pixel-space has received very little attention, despite the fact that many applications may benefit from such information. Examples incorporate generated content and robotic agents that must function in natural situations and are autonomous. In actuality, learning how to predict the future of a picture sequence necessitates that the system comprehend and effectively store the content and characteristics for a certain amount of time. Since labelled data video data is rare and difficult to get, it is considered as a viable path that might even aid supervised jobs. Consequently, this paper provides a summary of scientific advancements pertaining to future frame predicting and offers a repeated network model that employs new approaches from research in deep learning. The suggested architecture is founded on the recurrent process responsible with multilayer cells, which enables spatial-temporal data similarities to be maintained. Powered by perceptually driven decision variables and a contemporary recurrent working towards achieving, it outperforms previous methods for future frame creation in multiple video content genres. All of this may be accomplished with fewer training cycles and model parameters.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
CHAPTER	
CHAPTER 1: Introduction	1-5
1.2 Motivation	4
1.3 Rationale of the Study	4
1.4 Research Questions	5
1.5 Expected Output	5
CHAPTER 2: Related Work	6-14
2.1 Neural Network Approach	6
2.2 recurrent Network Approach	7
2.2.1 LSTM Encoder-Decoder Predictor Model	7
2.2.2 Convolutional LSTM Encoding-Forecasting Model	10
2.3 Adversarial Network Approach	12
CHAPTER 3: Datasets	15-17
3.1 Characteristics and Data Generation	15
3.2 Data Processing	17
3.3 Data Augmentation	17
CHAPTER 4: UCF-101	18-21
4.1 Characteristics	18
4.2 Data processing	19
4.3 Data Augmentation	21
CHAPTER 5: Experiment on Moving MINEST	22-26
5.1 Schedule Sampling	22
5.2 Experiment on Moving UCF-101	24

5.3 Hyperparameter Tuning	24
5.4 Test Results	24
5.5 Quantitative results	25
Chapter 6: Contribution	27-30
6.1 Architecture	27
6.2 Discussion	29
6.3 Future Work	30
REFERENCES	32-33

LIST OF FIGURES

FIGURES	PAGE NO
Figure 1.1: Example of an image sequence with an unknown future frame. The sequence is starting from the left and is taken from UCF-101	3
2.1: Single frame predictions using an ANN model with two hidden layers.	7
Figure 2.2: The composite LSTM autoencoder model. The top branch reconstructs the input sequence backwards in time, while the bottom branch performs frame predictions forward in time.	7
Figure 2.3: The ConvLSTM encoding-forecasting model that was used in the paper in the context of frame prediction and precipitation nowcasting.	11
Figure 2.4: A very basic convolutional network that maps a fixed number of input frames X to predict one or multiple future frames $Y = G(X)$. The feature maps exhibit the same height and width at each layer, but different depth.	12
Figure 2.5: Comparison of different loss function combinations using a simple CNN to predict one frames given four inputs. The second future frame is predicted recursively.	13
Figure 3.1: Randomly chosen samples of generated image sequences of size 64 x 64 from the Moving MNIST dataset.	16
Figure 4.1: Sequence examples from UCF-101. These frames have	20

been randomly selected from the different splits, cropped to 32 x 32 and filtered to ensure they contain at least a small proportion of motion.	
Figure 5.1: Influences of scheduled sampling regarding the training and validation error in context of recurrent networks and future frames prediction on Moving MNIST.	23
Figure 5.2: Comparison of validation results based on a model with either using the scheduled sampling or the always sampling training technique.	23
Figure 5.3: Computed image metrics computed on the UCF-101 dataset. All four networks use the same hyperparameter settings, but differ in the number of recurrent layers, as well as the used objective function.	25
Figure 6.1: Architecture diagram of the TensorLight framework and its modules. The user program can take advantage of the provided abstraction layer, but also use the library and utility functions standalone.	28

LIST OF TABLES

TABLES	PAGE NO
<p>Table 1: Metric results of predicted patch sequences using our approach on the UCF-101 test split. The achieved similarity and sharpness results by using different loss layers are given for both the first frame only and the mean of the generated sequence. All our models are trained for the same amount of time, in detail the 2-layer networks for 100,000 and the 2-layer network for 68,000 iterations. The outcomes of other approaches are not comparable to the proposed model in rows 5-9, because the metric results of the given other solution takes only moving areas of the images into account.</p>	26

Chapter 1

Introduction

People have dreamt about creating robots that can think and behave like humans since antiquity. This ushered in the discipline of artificial intelligence (AI), that remains an active area of study and is used in a variety of practical applications. Early AI was primarily concerned with solving issues that were difficult for humans to solve, such as determining the fastest route to an unspecified endpoint that used the well-known Dijkstra method [1]. Ironically, it turns out that problems that people can answer with pure intuition are extraordinarily difficult for computers to perform. It is difficult or impossible, for instance, to develop a software from scratch that can identify objects in images, pronounce the words in spoken text, or explain events in a video clip. In contrast, conventional computer programs must be written algorithmically as a series of instructions or a set of mathematical theorems [6]. However, it is rather difficult to apply this to multidimensional data, such as images or movies, which consist of an incoherent collection of pixels with several color channels, noise, and an infinite number of possibilities. Humans approach this kind of information differently. They know items by practice and implicitly construct mental connection structures. This fundamental notion created the area of machine learning (ML). It describes a process in which information is obtained by dynamic and ever changing from raw data and, as a result, enables sensible decision-making [6]. However, in order to solve many previously intractable issues, it is necessary to know which aspects to study, for instance by constructing a decision tree. Returning to our earlier example, this continues to be difficult to apply to photos or video data in which we know the characteristics we are searching for but cannot specify explicitly how they are represented. This problem is addressed by the discipline of representation learning, which attempts to create the representation automatically. Having just high-level representation may not be sufficient. In order to solve the issue, artificial neural networks (ANN) have indeed been developed. They are physiologically motivated by the structures of the brain and are capable of learning representational hierarchies [8]. For visual object identification, one may consider edges that are recognized at a very basic level and then constructed into curves or forms. In addition, these basic components may be combined in a certain manner such that the neural network can recognize separate

complex things within the data. The increase in computer capacity enables the development of ever-more-complex representation structures and deeper networks. This approach gave rise to the current term deep learning. In recent years, the discipline of deep learning has gained significant success, and according to its fundamental concept, "we are near to solving the issue if we have a suitable end-to-end model and adequate data for training it." [12]. Nevertheless, although there have been many studies and practical uses of object identification on static pictures or voice recognition, the use of these techniques to video data is only beginning to be investigated. Early attempts to deep learning using video data or basic picture sequences target challenges such as human action identification [7], [14], [4], and video categorization [8]. Another approach is optical flow detection [5], which detects the visual flow between frames. To train a network, the majority of these methods need a large quantity of labeled data. The laborious labeling procedure and thus limited availability of certain data may be the primary reason why this issue has not been well explored to yet.

This study investigates if deep learning methods may be effectively used to films to discover a precise interpretation in an unsupervised manner. It is determined if such a presentation is suitable for continuing a video after it has ended. Therefore, to get an understanding of the temporally and spatial development of a series of photographs as well as the movements and kinetics of a scenario. Such a high-level comprehension would be useful for autonomous intelligent beings that must act and, as a result, must comprehend our environment, along with its both physical and temporal constraints [14]. Other possible application areas include generated content [1], field of vision for autonomous vehicles, and optical flow replacement in causal video processing [2]. Other supervised learning tasks, such as human action recognition, might also benefit from such a which was before network in order to enhance overall performance or minimize training time. Obviously, additional types of action recognition are as readily imaginable.



Figure 1.1: Example of an image sequence with an unknown future frame. The sequence is starting from the left and is taken from UCF-101.

This job may seem simple for humans because we have developed an intuition about motion and our surroundings, as was the case with the previous example of object identification in static images. Taking a glance at the images in Figure 1.1, we can form a very good guess as to where this series is going. At the very least for a few ticks of the clock. Even though the ball remains to descend owing to gravity, the foreground youngster will likely move his left foot closer to it. Contrarily, the scenery will remain mostly untouched. Because of the complexity of modeling spatial and temporal data simultaneously and the exponential growth of the search space in multi-step forecasting, creating a deep learning technique to address this problem is no easy job. There are other concerns that must be addressed, such as the development of an efficient training procedure and the measurement of the perceptual picture consistency between expected and ground truth frames. In addition, the current state-of-the-art systems that tackle frame prediction must be evaluated and examined thoroughly to understand from their strengths and flaws.

There are several contributions to this theory. To begin, it gives a comprehensive review of the current deep learning methods that tackle the issue of video frame prediction in the future. Second, it introduces an architecture for neural networks that utilizes unique multilayered LSTM implementations and planned sampling to facilitate better training of recurrent models, building on the foundation of contemporary techniques such as batch normalization. As a result, its prediction error on the Moving MNIST dataset is almost half that of other state-of-the-art models. Finally, the research community has unrestricted access to all TensorFlow implementations, including the convolutional recurrent cell with planned sampling and batch normalization, and the various metrics and loss functions used to evaluate perceptual picture similarity during training. Also donated is a lightweight, high-level, open-source architecture for TensorFlow that may

significantly cut down on repetitive coding tasks often associated with deep learning projects. This is made easier by presenting an abstractions for several typical or difficult challenges encountered during the construction and training of neural network models.

1.1 Motivation

Bangladesh experiences serious road accidents and traffic congestion, with an average of 24945 traffic fatalities per year (2012-2019), which has been increased significantly since then. Inadequate safety at manufacturing and building sites leads to an increase in accidents and fires breakout causing massive loss of human lives and properties. Monitoring crime detection through surveillance to improve safety as crime rate has increased to 70% all time high (2019-2022) and lastly to reduce operative cost and fatal error implementation of automation in surgical procedures.

1.2 Rationale of the Study

The study's goal is to create a model that can anticipate future video frames. Video prediction models have the potential to improve a wide variety of applications, including video compression, video summarization, and autonomous navigation. The algorithm will be trained on big video datasets and tested for its ability to anticipate future frames. The study will also look into the effect of other aspects on the model's effectiveness, such as the length of the prediction horizon and the type of video. The study's overarching goal is to advance the state-of-the-art in video prediction and contribute to the creation of more intelligent and efficient video processing systems.

1.3 Research Questions

1. What is the most efficient method for predicting future frames in videos?
2. How does the length of the prediction horizon influence the model's performance?
3. What effect does the type of video (sports, nature, etc.) have on the model's performance?
4. What effect does the amount of the training dataset have on the model's accuracy?
5. Is it possible to utilize the model to improve video compression and summarization?
6. How does the model's performance stack up against other video prediction methods?
7. What are the model's limitations, and how might they be addressed in future research?
8. How can the model be used in real-world scenarios like autonomous navigation?
9. Is the model suitable for real-time video surveillance or video conferencing?
10. How does the model's performance alter when used to forecast future frames of varying resolutions or frame rates?

1.4 Expected Outcome

The construction of a model or algorithm that can accurately predict future frames in a video based on past frames is the expected conclusion of a "Predict Future Frames in Videos" thesis. The model should be capable of generating believable future frames and demonstrating this skill on a number of different types of films and scenarios. A full review of the model's performance, as well as a discussion of its limitations and potential future enhancements, should also be included in the thesis.

CHAPTER 2

Related work

This chapter presents existing deep learning approaches that have addressed the issue of future frame prediction. These are grouped into three sections depending on the model implementation, namely neural networks, recurrent networks and adversarial networks. The strengths, weaknesses and design decisions of these models are briefly discussed together with a short analysis of the achieved outcomes. Further, it is highlighted how these approaches have influenced the architecture of our final model that is used throughout the evaluation in Chapter 6. Aside from that, their results from the baseline in the assessment of our model.

2.1 Neural Network Approach

The first attempts to forecast future frame in a picture series were made in, followed by a paper in [16] that expanded on these ideas. They attempted to achieve single frame prediction using an artificial neural network, most likely as a result of the limited computer power available at the time and the lackluster development of CNNs at the time. In order to build a model using picture data, they undertook a number of preprocessing processes first. In the beginning, the data from the picture was split into three sections based on the RGB color channels: red, green, and blue. After that, using methods such as principal component analysis, the dimension of the data was decreased first from level of 104 to 100 within every individual section (PCA). After that, the ultimate learning and inference were carried out on three distinct neural networks with the same architecture, each one for each color image. Following each prediction, the principal component analysis (PCA) method was turned upside down to determine the starting dimensionality, and all triple results were merged to produce the ultimate picture.

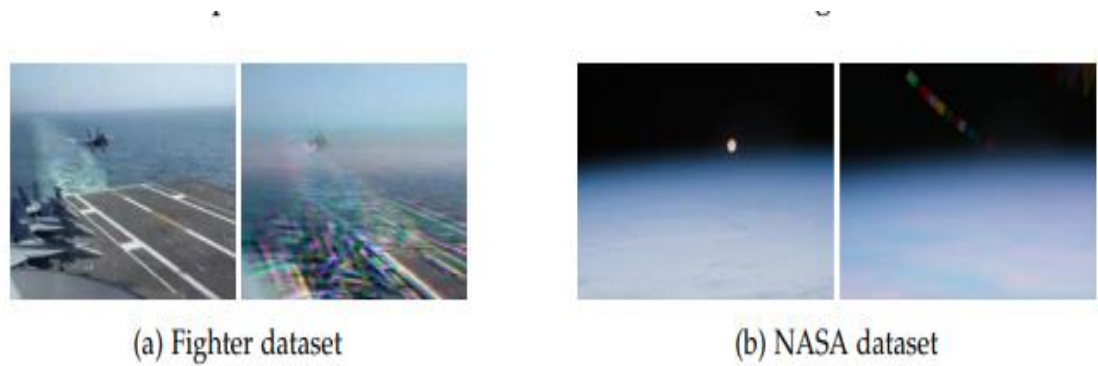


Figure 2.1: Single frame predictions using an ANN model with two hidden layers. Left: ground truth target frame. Right: generated prediction. (From [33])

In order to adjust the network to maintain the image's brightness, contrast, and structure, the loss incurred during in the training phase is expressed in terms on the MS-SSIM index. This index measures the magnitude of the loss. Because the data from the utilized Fighter and NASA datasets have a big picture size, using the multi-scale edition of SSIM is an option that is fair to make.

When we take a closer look at the outcomes of the predictions that have been displayed in Figure 3.1, we discover that the networking more or less uses an average across the sequence that it was given as input. This effect is seen rather clearly in Figure 3.1b, where it is expected that the motion of the moon from either the upper left more toward the earth's threshold is formed of prior moon locations. As a direct consequence of this, the straightforward design does not adequately capture the predict future performance that are present in the input data.

2.2 Recurrent Network Approaches

In order to leverage the sequential structure and temporal correlations of video data, several works performed frame prediction based on recurrent network models. The following models have inspired our final model architecture the most.

2.2.1 LSTM Encoder, Decoder, Predictor Model

Whenever the recurring encoder-decoder architecture introduced in Section 2.4.2 was employed in (SMSIS) to conduct machine learning technique of video representations,

a significant advance was realized. The notion that the exact same operation must be executed at each time step in order to create the very next state was the key reason behind their decision to employ this structure in that scenario.

They demonstrated in an LSTM auto - encoder system that is taught to recreate a whole input sequence of around 10 picture frames, similar to Figure 2.14. This model was then slightly modified in a second step to predict the future sequence of frames. In a last step, both models were combined to a single model that contains only one encoder to learn the dynamics of the video, but two separate decoder networks. Initialized by a copy of the learned representation, one decoder tries to reconstruct the inputs backward in time, while the other decoder predicts the future frames forward in time. Consequently, the decoder has to come up with a representation that can be handled by both decoders. In this way, they tried to compensate the shortcomings of each model, such as the potential tendency of the reconstruction decoder to learn the trivial function, or to counteract that the future predictor considers the last frames of the input sequence only. This combined model delivers the best results and is shown in Figure 3.2.

Within their work, they also explored if the decoder should condition on the previously generated output or not, as earlier discussed in Section 2.4.2. The final choice has fallen on the conditioned variant because it delivered slightly sharper frame prediction results

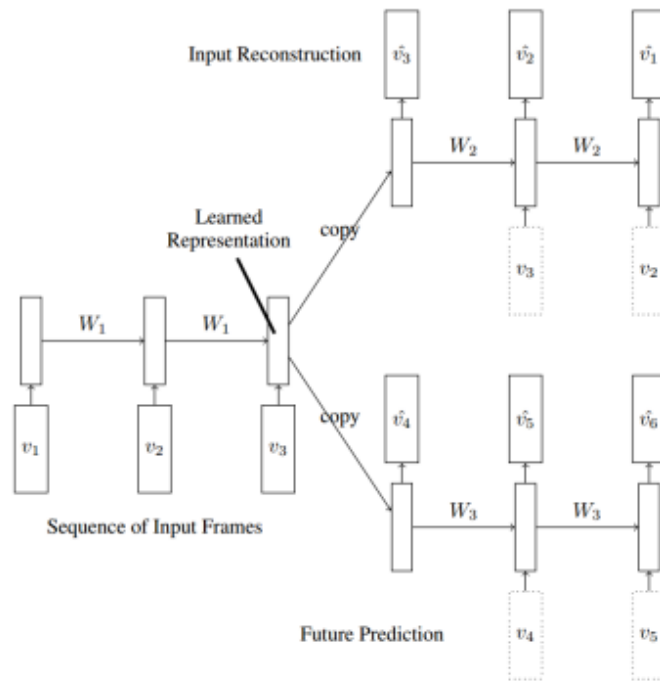


Figure 2.2: The composite LSTM autoencoder model. The top branch reconstructs the input sequence backwards in time, while the bottom branch performs frame predictions forward in time. (From [14])

in the qualitative evaluation. Besides that, they also varied the number of recurrent layers with the clear result that deeper LSTMs yields best performance.

Another contribution of this work was the introduction of a simple dataset that can be generated on-the-fly in order to explore the architecture of the model, as well as the effects of hyperparameter changes. It uses handwritten numbers that bounce around in a short sequence of images. Since this dataset is used in several other subsequent works as well, it offers the ability to be used as a basic benchmark to compare the performance of various models. The dataset will be presented in detail in Section 5.1. Sequences from this and another dataset were then used as input to the LSTM encoder to train the model. It must be highlight that they utilize the full image patch for this purpose. They have also mentioned to use convolutional percepts of the image sequence as inputs, but actually used this approach in the second part of their paper only, where the pre-trained encoder was transferred to improve the performance of supervised human action classification in videos.

The authors also pointed out that the choice of the loss function is fundamental with

respect to quality of results. Nevertheless, they decided to rely on standard error functions and kept the use of more advanced objective functions for further research. To be more precise, they trained their network using binary cross-entropy when being applied to Moving MNIST, and squared error for real world tests on UCF-101. Details about the latter dataset can be found in Section 5.3.

All in all, the strength of this model regarding future frame prediction is that it is able to infer a variable number of frames by taking the temporal correlations of the entire input sequence into account. But as a downside, the use of FC-LSTM cells with such high dimensional inputs implies a huge model complexity in the order of 10^9 in case of a two-layer LSTM with 2048 hidden units each. Consequently, such a model takes a very long time to learn useful patterns. Further, it does not consider spatial properties of each single input due to the use of fully-connected state transitions.

2.2.2 Convolutional LSTM Encoding-Forecasting Model

It was [47]'s overarching objective to build a deep learning strategy for precipitation nowcasting, therefore that's how they expanded on the model stated above. In contrast, they developed a variant of LSTM with multilayer architecture with both insight and state-to-state transfers to reduce the excessive duplication of determine the likelihood in conventional FC-LSTM cells. In Section 4.1.1, we will go further into its construction and internal structure. In a nutshell, they replaced all of the matrix multiplications with convolutions, such that the internal states are now tensors in three dimensions. This allowed them to save the spatial information. Once trained, these convolutional LSTM (ConvLSTM) cells may be employed in the identical decoder-encoder system as previously, as shown in Figure 3.3. The bottom line is that these units have been demonstrated to outperform FC-LSTM cells while also including far less hyperparameters, all while being better at capturing spatio-temporal features of the data.

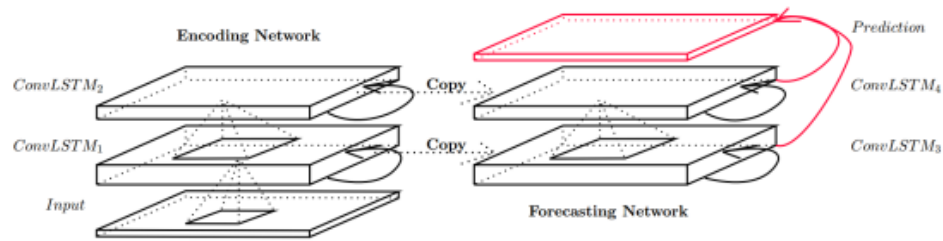


Figure 2.3: The ConvLSTM encoding-forecasting model that was used in the paper on context of frame prediction and precipitation nowcasting. (From [12])

Among other datasets, the authors trained this model on Moving MNIST in the course of their work and therefore it is another good candidate to compare our model with. The model was thereby fed with reshaped tensors of size $16 \times 16 \times 16$ by splitting the original frames using a 4×4 grid [12]. The reason for this reshaping was not argued in the paper, despite the fact that this unnecessarily increases the spatial redundancies. But since it divides the size of the image by a factor of 16, one simple reason might be to reduce the computational complexity because the depth is increased by the ConvLSTM's convolutions nevertheless. To generate the final prediction, the state of each ConvLSTM layer per time step is concatenated and led into a 1×1 convolutional layer for the purpose of reducing their depth to match with the ground truth target [12]. Also at this point, it was not explained why the concatenated hidden state of all layers is used to generate the prediction, instead of the more intuitive choice like relying on the output of the final layer only.

To condense the three most important findings of their evaluations, it was shown that the kernel size of the state-to-state transitions has to be at least bigger than 1×1 to capture spatio-temporal motion patterns. The windows size of this kernel can be interpreted as the maximum motion that the model is able to detect from one time step to the next. The second outcome is that deeper models can produce better results even when each layer contains fewer parameters. And last but not least, as already stated earlier in this section, the use of convolutional LSTM cells instead of FC-LSTM cells enables to reach better performance with less training examples, requires less iterations to converge and is less likely to overfit.

2.3 Adversarial Network Approach

In the final stage of this thesis, we came across a novel approach to train neural networks to perform frame prediction without using recurrent cells. The authors of [10] used a very simple overcomplete convolutional generator network $G(X)$ in order to generate a single or multiple frames from an input sequence X . This simple network is displayed in Figure 3.6 and consists of convolutional layers only with a constant height and width but variable number of feature maps. Training a model of such a simple architecture has several weaknesses, such as that it could only capture short-range dependencies across the entire input sequence with the size of the kernel due to the fixed size feature maps. Also, default loss functions such as L_1 or L_2 during the training experientially lead to blurry results, as previous studies have already shown.

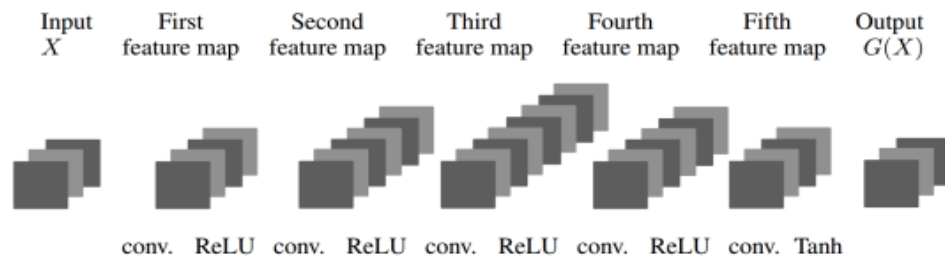


Figure 2.4: A very basic convolutional network that maps a fixed number of input frames X to predict one or multiple future frames $Y = G(X)$. The feature maps exhibit the same height and width at each layer, but different depth. (From [38])

To overcome these issues, they proposed three different but complementary learning strategies. Firstly, they used a multi-scale approach where multiple generator networks are iteratively trained on different scales of the input patch, starting from the lowest scale. The prediction of the next larger scale then used the upscaled prediction of the previous scale as a starting point. This technique enabled the network to consider motion patterns of longer range. Secondly, they extended their loss function with an additional GDL term in order to penalize blurry outputs in image space. This gradient-based loss function has already been introduced in Section 2.6.3. And lastly, they plugged this simple convolutional network into the adversarial training framework. The described network therefore defines a generator network C to predict the next frame,

while a second discriminator network D is consulted in order to assess whether the output of the generator network is the real target frame of the future or just a generated fake. Using an alternating training procedure, both networks learn to perfect the system. In other words, the discriminator network of this adversarial training process can be seen as an adaptive loss layer that assesses the generated output in feature space. Also other works highlight the usefulness of considering the error in feature space in addition to the image space error, such as in [3]. Summarizing the last two proposed learning strategies, a triplet loss is used where a standard loss function in image space is combined with a perceptual motivated loss function to preserve sharpness, as well as an adversarial error function which quantifies the realness of the generated frames.



Figure 2.5: Comparison of different loss function combinations using a simple CNN to predict one frame given four inputs. The second future frame is predicted recursively. (From [9])

The performance of different loss function combinations was also compared in a qualitative evaluation. Several output samples are shown in Figure 3.7. As it can be seen, the combination of multiple loss functions with different objectives enables to end up with predictions of higher quality and realism. Besides that, a detailed comparison to other LSTM based models on the UCF-101 video dataset was given. Thus, it allows us to compare our outcomes to all these results as well.

But even when this network using all the proposed learning strategies is able to produce outstanding frame prediction results, it comes with some weaknesses nevertheless. For instance, the temporal correlations of the input sequence are not explicitly modelled in the generator network. Hence, it has to explore the sequential structure of the data by its own from scratch. Furthermore, adversarial networks are said to be hard to train, because the oscillating loss values of the generator and discriminator networks are tough to interpret. Also experience is from advantage since the learning rates of both networks have to be kept in balance.

CHAPTER 3

Datasets

This chapter presents all datasets that are going to be used in the following evaluation. Three different video datasets were chosen that are used in related works in order to be able to compare the results and analyze the strength and weaknesses of different network models. The selected datasets will be introduced one after another, ordered by the content complexity with respect to the possible variations in color, motion and physical environment. Additionally, random samples from each dataset are shown to get a better idea of how the data looks like that is fed to the network.

Moving MNIST

To develop the algorithm, we use a simulated dataset of black-and-white pictures including soaring handwritten numbers. To better forecast the next frame in a video, [13] introduces the Moving MNIST. But since, it has been referenced several times in subsequent publications like [11] and [12].

3.1 Characteristics and Data Generation

Each series in the suggested configuration consists of 20 64×64 picture frames, each of which contains two randomly moving numerals from the MNIST collection. This straightforward dataset has the primary benefit of having a practically limitless size due to its ability to be produced instantly. Consequently, randomly generated digits are randomly selected from the beginning 55,000 digits of such training dataset and placed on any point of the initial picture patch while training a model. Each digit is given a velocity for the creation of succeeding frames, with the orientation of the velocity being evenly selected from a number line. Furthermore, when any digit with a size of 28×28 meets the wall, the straightforward physical rule that the incident angle equals the angle of reflecting is applied. Due to the need to anticipate the correct trajectory when a ball bounces off a wall, this also makes the dataset more dynamic and allows for numerous occlusion effects from overlapping digits. As a result, despite the dataset's ease of creation, it is challenging for models to provide reliable predictions for the test set

without first establishing a representation that captures the system's internal dynamics [Shi+ 15, p. 6]. Last but not least, we may start understanding the performance of the model with regard to its hyperparameters by using a smaller dataset. Particularly in light of the extensive training period required when using more complicated or even natural video.

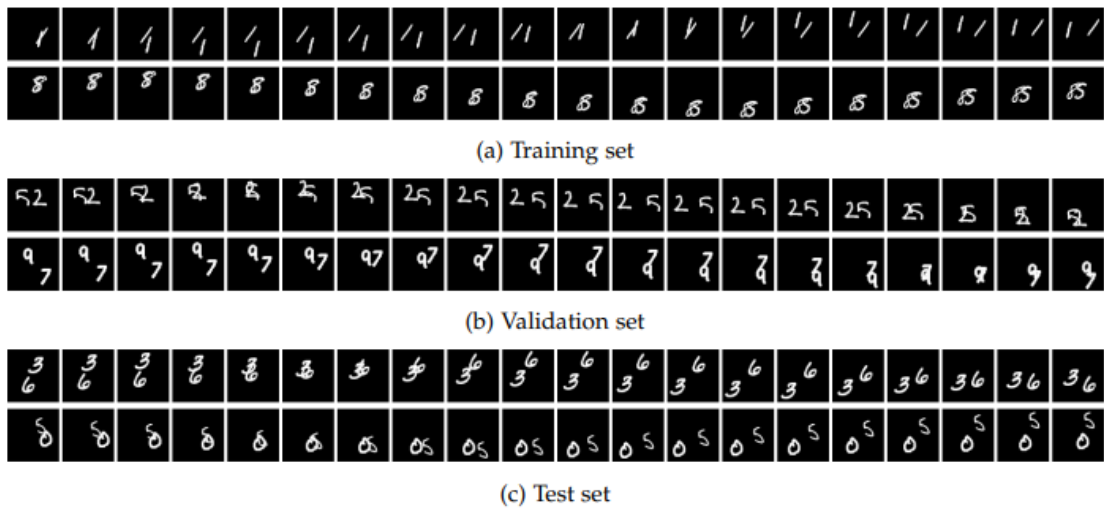


Figure 3.1: Randomly chosen samples of generated image sequences of size 64 x 64 from the Moving MNIST dataset.

With the exception of utilizing the final 5000 digits of both the previous MNIST training split, the production technique for the validation data is the same as the approach for producing the training data that was previously discussed. On the other hand, the MNIST sample split is not used to create the test set. Alternatively, the which was before test set from [11] was used, which consists of 10,000 instances of patterns that are precisely 20 frames long. More similar findings to at minimum one rival model may be obtained in this method. Figure 5.1 displays a representative sample from each of the above divides.

Even though some other works have used only a fixed number of pre-generated frame sequences, the on-the-fly generation process of the initial paper was kept for at least three reasons. First, it limits the amount of data and therefore increases the chance of overfitting. Seconds, loading the pre-generated frames from disk takes more time than generating them on the fly; hence it could have a slightly negative impact on the overall

training time. And third, it reduces the total memory requirements in case the whole data would otherwise be pre-loaded into memory in order to eliminate the second mentioned issue.

3.2 Data Preprocessing

To balance the data, as the actual input image of both the MNIST dataset are in the range [55], a simple way to go in order to [0, 1] is done. In addition, the subtraction of the mean adjacent pixels has been examined, since grayscale pictures possess the serial correlation condition. In view of the fact that it's seldom implemented in operation once the MNIST database is utilized [ING+13] and that it did not result in any observable benefits, average subtracting is not used when pretreatment the data. In addition, one may conclude that because the majority of an image's pixels are black (and hence zero), further data processing is not necessary.

In addition, instead of giving the model continuous floating values within the normalized range, only binary pixel values are employed. This selection is the consequence of using binary bridge as the primary error function for all of this dataset, as it has been proved to be the best option for picture producing models using MNIST [13]. Therefore, each pixel p is given a value of zero if $p < 0.5$, and a value of one to all other pixels. In addition, the output layer uses the sigmoid activation method to facilitate the washout of all pixels either as zero or one.

3.3 Data Augmentation

In regards to data augmentation, the luminance or brightness of the picture samples are not randomly altered, since it would create no sense within the framework of this video game, which utilizes a fixed color palette. However, since the game environment is replicated horizontally, unexpected horizontal flipping occurs if the selected crop does not reveal anything of the technical performance at the bottom. In addition, it iterates through all sequencing 256 times every epoch because to the premise that now the dataset contains only a small number of extremely lengthy sequences. A second reason is that just a brief clip and a modest random cut are taken from these frame sequences.

CHAPTER 4

UCF-101

Thirdly, a dataset consisting of complicated, natural films is employed to see whether the model can handle these as well. For this reason, we use the UCF-101 dataset [15]. It is one of the most extensive labeled datasets for video sequences identification, with 13,320 clips and over 27 hours worth of video data. User-uploaded films with chaotic backgrounds and shaky cameras make up the dataset. Each of the 101 categories may be further broken down into 25 broad groups, with videos in the same group all sharing characteristics like an about equal perspective. Interaction, bodily functions just, interpersonal communication, human social interaction with stringed instruments, and sports are the five action kinds that may be distinguished from one another. Although initially created for human action recognition, this dataset may also be utilized for frame prediction using just the raw video data. It's worth quickly mentioning that there's even bigger video dataset that can be used for feature representation preceding diving into the details of the features and preprocessing methods that were put into play. Over 1.1 million Video on youtube links belonging to 478 classes have indeed been automatically categorized in this massive dataset termed Sports-1M [17]. However, it is not used in this thesis because to infrastructure concerns with such a large dataset and a substantial time commitment for data pretreatment.

4.1 Characteristics

Videos in this dataset differ in length from around 1 second to such a peak of 71 seconds, with a mean duration of 6.2 seconds. However, there are a few films with somewhat different resolutions than the stated 320 x 240 (30 fps) and 25 fps (25 frames per second) in the original study. To make all films seem the same, these frames are either padded using zeros or cut in the middle. Figure A.1b shows one such padding film along with several additional segments as examples. Whether you're interested in action recognition or movement detection, this dataset has you covered with its three predefined train/test splits. In this study, we choose the 3rd standards split for activity recognition since it provides the greatest number of training films and the easiest test

split. Because it is assumed that a large training set is crucial for the network to thoroughly investigate the underlying dynamics, all validation data is drawn first from testing split rather than the training split. Others have utilized UCF-101 for framing prediction before, however they either didn't remark on the data partitions they used for validation and testing, or they only utilized 10% of the testing dataset. [10] In the end, the training set, the validation set, and the test set include 9624, 1232, and 2464 movies, respectively.

4.2 Data Processing

UCF-101's data preparation is fairly similar to the method outlined in Section 5.2.2, with both the following exceptions. Before using linear interpolation to create a smaller frame size, we first halve the width and height of each picture, resulting in 160 x 120 pixel films. This is carried out to make up for the films' loud, pixelated artifacts. A randomized crop of 32 by 32 pixels offers a better probability of having genuine motion in it rather than merely flickering due to noise. Second, while choosing the crop zone from the randomly chosen clip, the movement filtering limitation is relaxed significantly. A sequence with no motion at all in the last frame is not automatically rejected only because the motion was absent in the preceding frames. Numerous sequential samples plucked from the various dataset splits used to train our model are shown in Figure 5.3.



Figure 4.1: Sequence examples from UCF-101. These frames have been randomly selected from the different splits, cropped to 32 x 32 and filtered to ensure they contain at least a small proportion of motion.

In addition, because it is inefficient to load the entire video file into memory, particularly when considering that only a relatively small amount of the information is definitely used to create one example for the subsequent group, a supplementary interactive pre - processing step of the data is performed prior to the beginning of the actual training process. This occurs before the beginning of the official training process because it's incredibly inefficient to pack the entire movie document into memory. As a consequence of this, it cycles through all of the video footage files and creates semi binary scenes that are each 30 frames long. After carrying out this procedure only once, the files that are produced may, thankfully, be utilized again. In the end, it comes down to having 14,451 footage for the test set, 55,150 clips for the test dataset, and 7183 clips for the test dataset.

4.3 data Augmentation

In terms of the enhancement of the data, the sharpness and luminance levels of the whole picture sequence are randomized to be changed by a differential of 20%. In addition to that, the training data go through a random process of horizontal flipping. Even if there is no random fluctuation in the intensity or brightness of the verification or test data, their size is increased by employing both of the normal as well as the flipped examples. In order to strike a compromise between the need for more uniform assessments and the want to maintain an acceptable level of processing speed, each review iteration makes use of four crops taken from each video. The processes necessary for this enhancement are carried out in real time. An enhanced double-buffered entry queue is used in order to make this process easier. The very first dynamic Scene is then arbitrarily populated containing reference to the binary sequential files that were produced before. After that, sixteen different CPU threads simultaneously similar material a reference from any of this queue, load it into memory, and then proceed to complete all of the preprocessing stages in parallel. This is done with the intention of producing a single example for training purposes. The last step is to add this example to the shuffled group queue, which is the location from where the model pulls the batches that it uses for each iteration. As a direct result of this, there is no downtime during the various stages of training.

CHAPTER 5

Experiments on Moving MNIST

To test how well our model performs, we begin with the fabricated Moving MNIST dataset. The network must decode the motion of an input pattern of 10 frames in order to forecast the following ten frames in the future. Because this dataset has been utilized in a number of earlier publications, we compare our findings to those of other network models here, both qualitatively and numerically. In this subsection, we use a loss layer that initially establishes $ssim = 0$.

5.1 Scheduled Sampling

In this first part, we take a look at the aftereffects of the planned sampling method. We use a modeling approach, and Figure 6.1 shows the binary cross-entropy during training and validation using both scheduled sampling (SS) and always sampling (AS). It is analogous to SS with a continuous selection probability of $p = 0$ that the latter technique employs, meaning that it always samples from the previously created frame. In the first stage, when the SS-approach primarily educated on input samples retrieved from the ground truth, a large gap can be detected between the generalization error and the validation error. Since no cell in the recurrent network needs to make up for the mistakes of its predecessors, training loss converges much more quickly when ground-truth samples are used at every time step. On the other hand, the validation loss is quite high at this stage since the results are the mean of 10 predictions as well as the spatio-temporal decoder unexpectedly uses its own deep feature projections to make a prediction on the following frame, which is not how it was operated during training. However, the key takeaway here is that, in comparison to the strategy of always taking samples from previously generated frames at the very beginning, the accomplished prediction accuracy is consistently better once the scheduled sample selection component has completely switched the input behaviour patterns to inference mode.

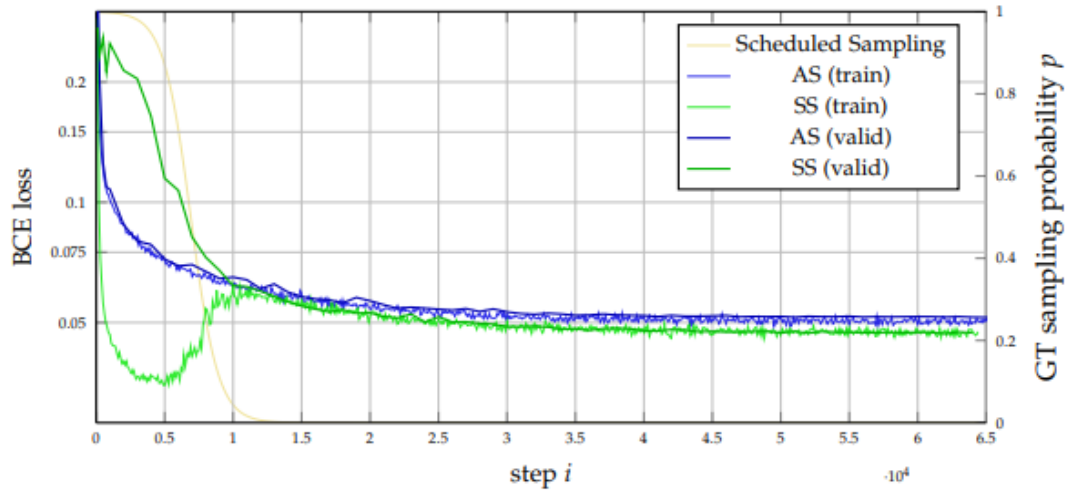


Figure 5.1: Influences of scheduled sampling regarding the training and validation error in context of recurrent networks and future frames prediction on Moving MNIST.

This could happen because the SS approach has a kind of pre-training phase, during which the network could indeed gain knowledge to anticipate the next frame whenever the current frame is ideal. So, it doesn't have to deal with mistakes made in the last time step. By slowly switching this actions to the method it uses during implication, it starts to learn how to deal with input frames that aren't perfect. When you look at Figure 6.2b and compare the PSNR results for AA and SS, you can see that they behave in a similar way. But then when people look just at edge enhancement differential measure in Figure 6.2a, we can see that scheduled sampling makes the predictions more accurate over time.

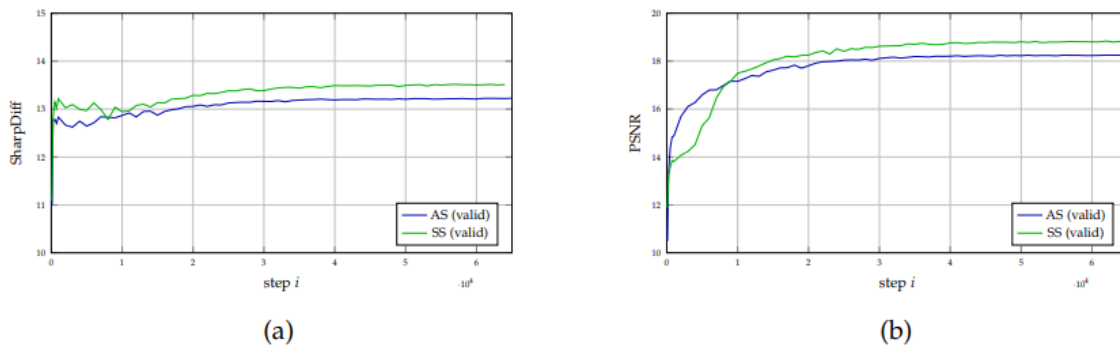


Figure 5.2: Comparison of validation results based on a model with either using the scheduled sampling or the always sampling training technique.

5.2 Experiment on UCF-101

In the last test, we look at how well the suggested neural network model works with genuine footage. In Section 5.3, we go further into the specifics of how we train our networks: using 32x32 UCF-101 patches. This dataset's clips depict real situations with such a lot of motion and noise, thus it's reasonable to assume that the addition of perceptually driven bias factors inside the error function may have its full impact. With the goal of finding even more substantial gains, we redo the analysis of various gradient descent configurations from of the experiment 2 on MsPacman and apply it to this database.

5.3 Hyperparameter Tuning

Finding optimal hyper - parameter values and subsequent parameters is quite similar to the method described in Section 6.2.1. A number of model examples are trained with different rates of learning, but time is saved by restricting the search space for the regularization coefficients to, for example, $= 1e 6, 1e 5$. In addition, the lessons from both prior trials suggest that no single-layer ConvLSTM arrangement has been evaluated. In particular, most experiments are run on two-layer ConvLSTM configurations, and then the same setup is tested with a three-layer ConvLSTM. While it has been proved that using MSE as the primary objective element of the triplet loss function yields the greatest results on MsPacman, the error percentage is still employed as the primary loss term. That's because, as discussed in Section 2.6.1 in Chapter 3, it works very well in the context of natural imagery. The grid search converges on the identical model setup as the preceding test. However, at this stage, the benefits of utilizing lower learning levels become more apparent than they were before. Researches below assume a mass decay of $= 1e 5$ and a learning rate of $= 0.0005$ to simplify matters.

5.4 Test Results

In the following, the test results of different network configurations are presented in a quantitative and qualitative analysis. In analogy to the previous experiments on

MsPacman, it is worth mentioning that the video patches from the test set are not filtered in case they show very less movement.

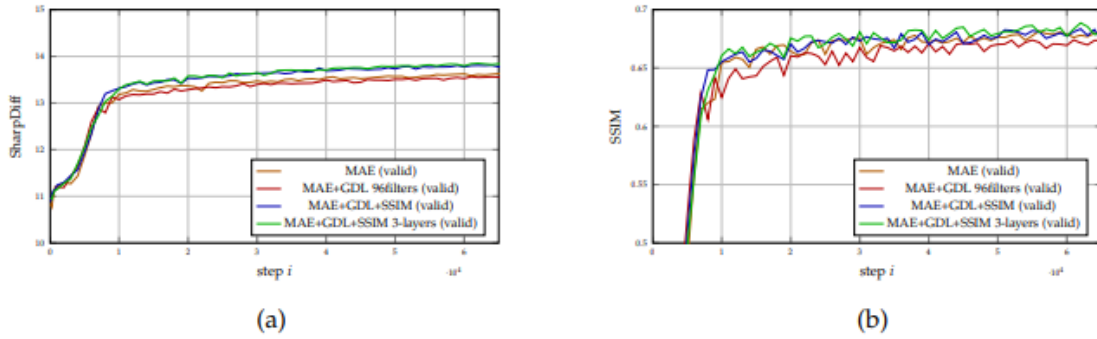


Figure 5.3: Computed image metrics computed on the UCF-101 dataset. All four networks use the same hyperparameter settings, but differ in the number of recurrent layers, as well as the used objective function.

5.5 Quantitative Results

Starting with the qualitative test results, Table 6.4 compares several configurations of our model with outcomes from related works on various image similarity or sharpness metrics. It demonstrates test results of the first predicted frame, as well as the average metric values of the forecasted 10 frames long sequence. However, the results of our model are not really comparable with [10], since they calculate the metrics only on specific regions of the image, where the optical flow exceeds a specified threshold.

Loss function	1 st frame			mean of 10 frames		
	Similarity		Sharpness	Similarity		Sharpness
	PSNR	SSIM		PSNR	SSIM	
single-scale ℓ_2 [MCL16]	26.5	0.84	24.7	-	-	-
multi-scale ℓ_1 [MCL16]	28.7	0.88	24.8	-	-	-
multi-scale GDL+ ℓ_1 [MCL16]	29.4	0.90	25.0	-	-	-
multi-scale Adv+GDL [MCL16]	31.5	0.91	25.4	-	-	-
MSE+GDL	15.9652	0.7112	13.6947	22.5218	0.7016	14.5219
MAE	30.8241	0.9043	17.0282	25.9873	0.7653	15.4952
MAE+GDL ₉₆	29.4092	0.9047	16.7509	25.8577	0.7686	15.4089
MAE+GDL+SSIM	31.3144	0.7668	17.3785	26.6385	0.7668	15.7363
MAE+GDL+SSIM 3layers	29.4000	0.9031	17.2316	26.2130	0.7679	15.7283

Table 1: Metric results of predicted patch sequences using our approach on the UCF-101 test split. The achieved similarity and sharpness results by using different loss layers are given for both the first frame only and the mean of the generated sequence. All our models are trained for the same amount of time, in detail the 2-layer networks for 100,000 and the 2-layer network for 68,000 iterations. The outcomes of other approaches are not comparable to the proposed model in rows 5-9, because the metric results of the given other solution takes only moving areas of the images into account.

CHAPTER 6

Contribution

Before concluding this thesis, we would like to present a side contribution that arised in parallel to this thesis. When the implementation of this final project has started, the TensorFlow library for machine intelligence had just published its second public release with version 0.7. Thus, there has not been that much experience and best practices around with TensorFlow, as well as its API is very low-level for several use cases even today. As a result, there has been the desire to create a reusable library to reduce boilerplate code of TensorFlow based projects, as well as to retain best practices of existing examples and also the lessons learned from thesis. A second idea has been that future theses or other deep Learning projects of the Computer Vision Group' at TUM might benelit from such a library. However, this project has grown larger and larger over time and ended up in a powerful high-level framework, that has been developed independently from other high-level APIs for TensorFlow like TF-Sfiin² or Reras³. Ultimately, about 99% of the overall code of this thesis has been transferred into this framework, consistently with having abstraction and reusability in mind.

6.1 Architecture

The framework may be broken down into three primary components when looking at it from an architectural vantage point. To begin, a set of utility maximization that have nothing to do with machine learning is presented here. A few examples of such functions are those that process photos and videos, download and extract databases, process pictures and videos, and make animations and movies from data arrays, just to mention a few. The second component is the high-leiieel library, which is built on TensorFlow. It encompasses various modules that either give a simple access to functions that it regularly requires while constructing deep learning applications or features that are not yet available in TensorFlow. Either way, it was necessary when creating deep learning applications. For example, it takes care of the construction of bias and weight parameters on its own, provides a number of ready-to-use methods for

loss and activation, and has some sophisticated visualization tools that can show feature maps or output pictures direct in an IPython Notebook. Third, an encapsulation layer to simplify the entire lifecycle, to extend the description of model graphs, and to allow reusable and uniform access to datasets. This layer will also help generalize the concept of a model graph. TensorLight's general architecture is shown in Figure 5.1 for your reference.

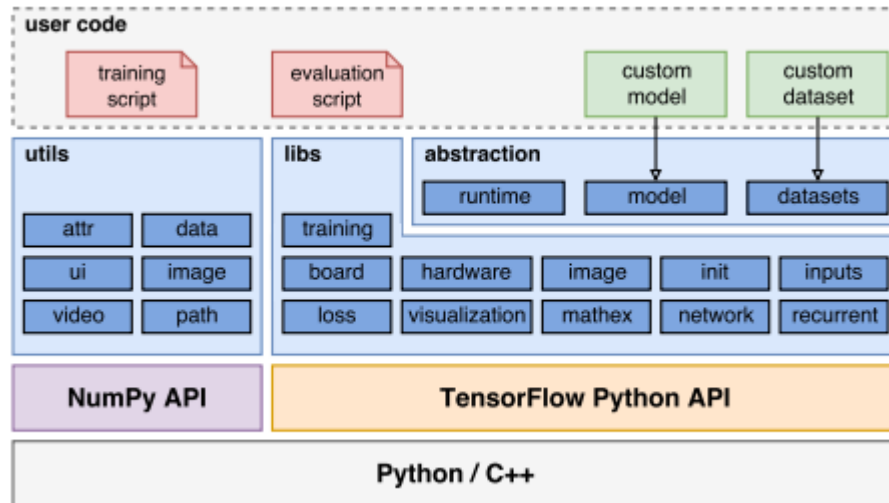


Figure 6.1: Architecture diagram of the TensorLight framework and its modules. The user program can take advantage of the provided abstraction layer, but also use the library and utility functions standalone.

The user program has the option of either using the high-level framework and the supplied functionalities for his already established projects or making use of TensorLight's encapsulation layers while developing new deep learning projects. Because of the latter, it is possible to significantly cut down on the amount of programming that needs to be written in order to train or evaluate the model. This is accomplished by enclosing inside a runtime module not only the full training or assessment loop but also the lifespan of TensorFlow's session, structure, summary-writer, or checkpoint-saver. This makes it possible to attain this goal.

6.2 Discussion

After applying batch normalization and the scheduled sampling learning strategy, we were honestly surprised that our model was able to outperform related models by such a large margin. Nevertheless, we believe that there is still space to improve the proposed architecture, the chosen hyperparameter configuration and particularly the data preprocessing. Regarding the latter, only simple rescaling of the data has been performed to be roughly zero mean, but dedicated the scaling of the values completely to batch normalization layers and the entire network.

We could also figure out that the choice of the appropriate loss function has a huge impact regarding the generated future frame predictions, if not even the most tremendous effects. However, as the evaluation in chapter 6 clearly shows, there is no perfect solution for this purpose. But it must be emphasized that the detailed properties of the used image or video data have to be analyzed in detail, in order to be able to achieve good results. Having said that, the fine-tuning of neural networks in context of image processing tasks remains to be very difficult even with a good loss function at hand. This can be attributed to the discrepancy between the mathematical and perceptual similarity of two images. Also the use of perceptual motivated metrics presented in Section 2.6.2 is not always very helpful, because an increase in one metric can lead to a decline in others. It is easy to get lost when multiple metrics are used.

Finally, it can be assumed that the proposed model could be currently trained more effectively using a different deep learning framework than TensorFlow, at least at the time of this writing. This can be argued with the fact that the current batch normalization layer in this framework currently depends on some operations where no GPU kernel is implemented yet. Such a bottleneck might be the root cause why the training process of our model is so slow that it requires up to four days to train the network for only 100,000 steps. But this will certainly change in one of its next releases.

6.3 Future Work

At least the five suggestions listed below have been made for potential future work:

To begin, the suggested model for the network should be scrutinized and its parameters adjusted in more depth. We have a strong opinion that this architecture is capable of achieving even better results by undertaking a more in-depth hyper - parameter discovery, training it for a much greater number of iterations, or using a bigger dataset such as Sports-1M. Regrettably, this is outside the scope of the timescale for this thesis; hence, certain assessments were carried out on networks that still had additional potential in the event that further training iterations were carried out.

Secondly, because the application of the scheduled sampling learning strategy for recurrent networks has improved our results in such an extent, it would be worthwhile experimenting with new variants of this approach. For instance, the recurrent network could dynamically grow in the course of the training process. Thereby, it could start to predict a single frame only until the validation loss reaches a specified threshold. Afterwards, the decoder RNN can be extended at runtime to predict more and more future frames per training iteration. As a result, such a network should be able to predict longer sequences with a higher stability regarding the quality of generated frames.

Thirdly, since the GAN approach described in Section 3.3 yields such promising results, one has to imagine what might be possible when the proposed model is plugged into this adversarial framework. Despite the fact that they use the probably simplest generator network one can think of, our proposed model as a replacement for their generator network would explicitly take advantage of the spatio-temporal properties of the data. Especially without the need that the model would have to learn these correlations from scratch. As a consequence, the additional adversarial network would introduce an additional objective function in feature space, whose benefits have already been mentioned in the end of section.

Next, a trained network instance of our model can be examined to serve as a pre-training for supervised learning tasks like human action recognition, which can be very helpful according to [18]. Similar efforts have already been taken in [13] with positive results. In detail, it should be possible to detach the encoder components of our trained

model, including its ability to generate a useful feature space representation given a sequence of frames, and plug it into a different network architecture specialized for classification. Unfortunately, performing such experiments with labeled video data is beyond the scope of this thesis.

Lastly, the proposed network architecture itself can be further extended to cope with different unsupervised tasks. To name just one, the recurrent components can be updated to bidirectional RNNs in order to solve tasks like slow motion video generation or video compression more effectively.

Reference:

- [1] J. Ascenso, C. Brites, and F. Pereira. “Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding.” In: *5th EtrFASiP Conference on Speech and Image Processing, Multimedia Communications and Services*. 2005, pp. 1—6.
- [2] C. Couprie, C. Farabet, Y. LeCun, and L. Najman. “Causal Graph-Based Video Segmentation.” In: *ICIP*. 2013.
- [3] A. Dosovitskiy and T. Brox. “Generating Images with Perceptual Similarity Metrics based on Deep Networks.” In: Feb. 2016.
- [4] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. “Long-term Recurrent Convolutional Networks for Visual Recognition and Description.” In: 2014.
- [5] P. Fischer, A. Dosovitskiy, P. Hausser, C. Hazirbas, and V. Golkov. “FlowNet: Learning Optical Flow with Convolutional Networks.” In: 2015.
- [6] I. Goodfellow, Y. Bengio, and A. Courville. “Deep Learning.” Book in preparation for MIT Press. 2016
- [7] S. Hochreiter and J. Schmidhuber. “Long short-term memory.” In: *Neural Computing*. 1997, pp. 1735—1780.
- [8] S. Ioffe and C. Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” In: *Proc. of IEEE*. 2015.
- [9] D. P. Kingma and J. L. Ba. “Adam: A Method for Stochastic Optimization.” In: *CCLI*. 2015.
- [10] A. Kar. “Future Image Prediction using Artificial Neural Networks.” In: 2012. A. Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks. May 2015. cxi: <http://karpathy.github.io/2015/05/21/rnn-effectiveness> (visited on 09/25/2016).
- [11] M. Mathieu, C. Couprie, and Y. LeCun. “Deep Multi-Scale Video Prediction Beyond Mean Square Error.” In: *iCLS*. Feb. 2016.
- [12] V. Pătraucean, A. Handa, and R. Cipolla. “Spatio-temporal video autoencoder with differentiable memory.” In: *ICLs*. 2016. K. Ridgeway, J. Snell, B. D. Roads, R. S. Zemel, and M. C. Mozer. “Learning to Generate Images with Perceptual Similarity Metrics.” In: Mar. 2016.
- [13] X. Shi, Z. Chen, H. Wang, and D.-Y. Yeung. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting.” In: 2015.
- [14] N. Srivastava, E. Mansimov, and R. Salakhutdinov. “Unsupervised Learning of Video Representations using LSTMs.” In: *MCML*. 2015.

- [15] K. Simonyan and A. Zisserman. "Two-Stream Convolutional Networks for Action Recognition in Videos." In: Proc. NIPS. 2014, pp. 56&-576.
- [16] N. K. Verma. "Future Image Frame Generation using Artificial Neural Network with Selected Features." In: PIPE. Oct. 2012.
- [17] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng. "Large Scale Distributed Deep Networks." In: NIPS. 2012.
- [18] X. Glorot, A. Bordes, and Y. Bengio. "Deep Sparse Rectifier Neural Networks." In: *AISTATS*. 2011.

Deep Learning Approaches to Predict Future Frames in Videos

ORIGINALITY REPORT



PRIMARY SOURCES

1	online-journals.org Internet Source	9%
2	Submitted to Daffodil International University Student Paper	1%
3	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
4	Tariqul Islam, Md. Hafizul Imran, Md. Ramim Hossain, Md. Tamjeed Monshi et al. "Deep Learning Approaches to Predict Future Frames in Videos", International Journal of Recent Contributions from Engineering, Science & IT (ijES), 2022 Publication	<1%
5	open.metu.edu.tr Internet Source	<1%
6	Chen, C H. "LATEST DEVELOPMENTS OF LSTM NEURAL NETWORKS WITH APPLICATIONS OF DOCUMENT IMAGE ANALYSIS", Handbook of Pattern Recognition and Computer Vision, 2016. Publication	<1%