Final Report - Data Science 209b

Video Generation Project

https://github.com/VincentCa/videogeneration

Prepared by Group #13:

Vincent Casser, vcasser@g.harvard.edu
Camilo Fosco, cfosco@g.harvard.edu
Justin S. Lee, justin_s_lee@g.harvard.edu
Karan R. Motwani, kmotwani@g.harvard.edu

1. Problem Statement and Motivation

In this project, we present a variety of deep learning based setups for text-to-image and text-to-image sequence generation. Image sequence generation is a challenging task and an actively researched branch within computer vision, posing special challenges such as temporal coherency. To address this problem, we describe a variety of partial and complete solutions that we developed in three stages: (1) Text to Image Synthesis: using a traditional GAN, we generate a single image from a textual representation. (2) Text + Image to Video Synthesis: using a fully convolutional network, we generate an image sequence given a single frame and a description of the action taking place. (3) Inspired by the recent success of generative adversarial networks, we then also train this architecture in a truly adversarial setting.

Throughout our work, we make use of different datasets. Primarily, we evaluated our approaches on our own synthetic datasets with increasing difficulty, then moving to natural images from a human action dataset. We also performed text to image synthesis experiments on the T-GIF dataset, but noticed that the high diversity and other issues with this dataset make it rather unsuitable for video generation experiments.

2. Introduction

Generative Adversarial Networks (GANs) have received particular attention in the past few years, mainly for their ability to learn rich, continuous distributions over a complex feature space. They have been used to generate super-resolution images, change pictures from night to day, and generate impressive images of birds from raw text, among many other applications.

A clear next step for GANs is to be able to generate animations, short videos and, perhaps at some point, full-length feature films. However, these applications are still in a stage of infancy - current work applied to videos has shown that it is possible to generate a few new frames based on a particular pose or situation (Tulyakov et al. 2017), but in a very limited resolution and with little correlation to the actual action that is performed in the ground truth data. A possible path to address these issues involves leveraging the current advances in text to image generation (Reed et al. 2016; Zhang et al. 2017). A model could be conceived where a simple text description of the action/performance guides the generation of the following frames given an appropriate starting point (i.e first image of a scene). Efforts have been made towards this kind of solution: Marwah et al. (Marwah et al. 2017) propose an attention-based pipeline that generates new video based on a given caption. The pipeline works in a recursive manner, and uses the frames that were already generated as input to their architecture to generate the next frame of the sequence. Pan et al. (Pan et al. 2017) propose a GAN-based approach to create videos that conditions the generation with a given caption.

We use a slightly similar training strategy to Pan et al., but with an entirely novel architecture, to generate GIFs containing translations and rotations of multiple objects, and finally animating human motion (walking). Following our investigations, we were able to not only reproduce their results, but also propose a successful and very promising architecture towards tackling the general problem.

3. Project Trajectory

In our initial project proposal, we planned to use GANs to generate GIFs conditioned on an input caption and first frame, trained on the T-GIF dataset. However, training a purely text-to-image GAN (i.e. no first frame conditioning) on a simplified subset of T-GIF yielded a model that did not reach photorealistic levels. Combined with the fact that in the literature, models are typically developed for more specialized data, we concluded that the complexity of topics and visual information in T-GIF was too much for a neural network to model. Based on this, we decided to (1) create synthetic data depicting basic events that served our needs to test our new architectures, in the manner of Pan, et al. (2017), and (2) find datasets that depicted a narrower set of events than in T-GIF. For the purposes of (2), following Marwah, et al. (2017), we decided to use the KTH Human Action dataset by separating the videos of one action (walking) into individual frames, and taking advantage of the structure of the videos to generate simple but sufficient captions for sets of frames, thereby creating a usable dataset for our architectures. This simplification of our input data boosted the signal of what we set out to model initially: temporal cohesion in image sequences as perceived by a human observer.

4. Literature Review

The field of video generation and text to image systems is constantly growing. We will focus on the papers most relevant to our work below; other references used in our work follow.

a. Pan, et al. [2017]: To Create What You Tell: Generating Videos from Captions

This paper was the inspiration for a substantial part of our work; researchers developed a novel architecture to generate videos conditioned on an input caption. They used a modified GAN with a generator that outputs video (i.e. sequences of images), for which the typical input noise vector is augmented by an LSTM sentence embedding of the input caption. The discriminator was also modified to account for the definition of a "real" data point in this context: not only must a video appear temporally cohesive to a person, but it must also match its accompanying caption. The generator was trained on a loss that was a uniformly weighted sum of three loss functions that each emphasize different aspects of temporally realistic video that are semantically aligned with an input caption. The researchers also trained their model on a synthetic dataset created by moving MNIST digits paired with appropriate captions; this motivated us to create our own synthetic datasets. While our approach was different from that of the authors, their work and perspective were helpful to us.

b. Reed, et al. [2016]: Generative Adversarial Text to Image Synthesis

In this paper, researchers trained a convolutional GAN to generate high-quality images of birds and flowers from detailed text descriptions, trained on sets of text and accompanying images. An example of a training data point in this work is a detailed text description of an image (e.g. "an all black bird with a thick, rounded bill") and a corresponding set of images that satisfy the given caption. A CNN for learning image properties was used in concert with an RNN for caption analysis. This paper shows that, even without the problem of temporal coherency, generating realistic visual data from textual descriptions is difficult to achieve for all but the simplest categories.

c. Vondrick, et al. [2016]: Generating Videos with Scene Dynamics

In this paper, researchers trained a GAN to generate short, realistic videos on the order of one second at full frame rate. The generative network was given noise as input data, and generated a video in two channels: the first channel was a spatio-temporal convolutional network for foreground generation, whereas the second was solely a spatial convolutional network for background generation. A discriminator was also trained to recognize realistic videos from fakes. This architecture served to address the two observations that realistic videos need temporal coherency, and backgrounds of videos are usually stationary. The researchers were also able to pass an image as input, and have the network generate a short video as output. Unlike our proposed work, all training videos used in this work were unlabeled. This paper shows that realistic video generation conditioned on a first frame is possible; however, we believe that this could be better guided by incorporating textual descriptions.

d. Marwah, et al. [2017]: Attentive Semantic Video Generation Using Captions

This work describes a novel architecture for video generation with temporal and semantic considerations. The authors approach the problem of video generation from a probabilistic perspective, specifically by putting a distribution on Y_i, the i^{th} frame in a video Y, given a caption X and all previously generated frames. The distribution $P(Y_i \mid X)$ is then defined in terms of two components: one distribution that conditions Y_i on X and all preceding generated frames (i.e. the long-term stimulus), and another that conditions Y_i only on X and the frame that immediately preceded Y_i (i.e. short-term stimulus). The researchers also note the importance of attention, as it is often the case that different words in a caption correspond to different "refresh rates." That is, some words denote the background of the video, whereas others denote a subject. The caption attention, long-term stimulus, and short-term stimulus are all modelled using various LSTMs. A Variational autoencoder conditioned on the attention pipeline output is then used to generate the next frame. Like in Pan, et al. (2017), these authors also use moving MNIST digits, in addition to the KTH dataset, and achieve good results. It is interesting to note that here, the previous frames are used to generate the next one (similar to what we do), while in Pan et al., only the caption is used to synthesize the video.

5. Description of Data

Throughout our project, we use a variety of datasets that we outline below. We arranged the dataset table such that there is a progression from less diverse to both more diverse and challenging ones.

Name	**Description**	**Size**	**Reference**
Icons-in-motion I	Sequences of single icons at different scales, moving up-down/left-right. Icons are flags, hearts, clocks, houses, clouds, arrows, keys. Icons vary in size.	10,000 GIFs of 10 frames each	Available in Notebook
MNIST-in-motion I	Sequences of single digits moving up-down/left-right. 10 digits in total, with varying drawing styles.	10,000 GIFs of 10 frames each	Available in Notebook
Icons-in-motion II	Sequences of single icons at different scales, moving up-down/left-right, or rotating around their own axis or the image center. Icons are flags, hearts, clocks, houses, clouds, arrows, keys.	10,000 GIFs of 10 frames each	Available in Notebook
MNIST-in-motion II	Sequences of two digits moving up-down/left-right independently. 10 digits in total, with varying drawing styles.	10,000 GIFs of 10 frames each	Available in Notebook
KTH Human Action Dataset	Contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios, from which we used videos of people walking.	400 sequences of a person walking of varying length. Transformed to 2336 sequences of 10 frames or less for our experiments.	http://www.nada.kth.se/cvap/actions/
T-GIF	GIFs scraped from Tumblr, each with an unstructured sentence	101,413 GIFs of	Li et al.

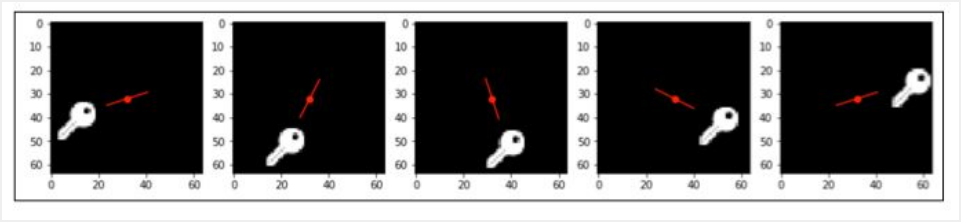
	describing its content.	varying length	[2016]

Table 1: *Summary of the datasets we used, showing name, a description, the respective size and reference where they can be accessed.*

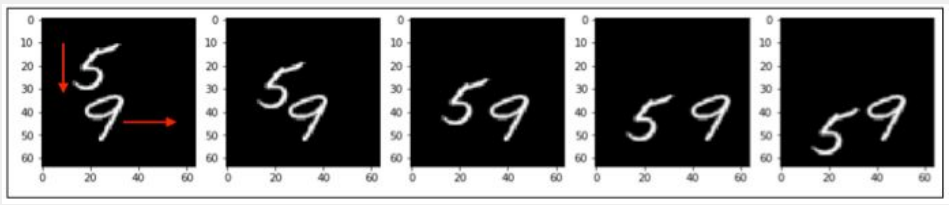
Some examples from the described datasets:

(a) Example from the KTH dataset, caption “a man walks right”

(b) Example from the TGIF dataset, with caption “a man is casting a spell and two school boys looking down”



(c) Example from icons-in-motion I (created by us), caption “the key is rotating counter-clockwise”



(d) Example from MNIST-in-motion II, caption “the digit 5 is moving up and down and the digit 9 is moving left and right”

Figure 1: *Examples from our various datasets.*

Notes:

- The data from Tumblr consists of GIFs depicting inconsistent actions or random movements. Modelling an image sequence generation model for reproducing a GIF of this type will require very comprehensive training data of each action type.
- The Tumblr dataset also consists of captions that are often very high level and short representations of the GIF content. GANs for text-to-image or image sequence require a larger number of more detailed descriptions.
- The KTH action dataset addresses some of the above concerns by maintaining more homogeneous backgrounds and smaller variation in actions.
- KTH frames without action were filtered out using frame-level labels provided with the dataset. Caption generation was done automatically by exploiting the fact that each video featured one person walking left-right and right-left twice in that order.
- There are 200 GIFs each for ‘a man walking left’ and ‘a man walking right’, which is sufficient for modelling the movement.
- The MNIST data on the other hand is synthetic and specifically curated for generation of image sequences. The images are simpler and our generation process allows us to create as much data as needed, as well as control the amount of variability.
- Similarly, the synthetic dataset of icons provides the same advantages as MNIST but is a bit simpler as we have only 7 classes and not much variability between the forms of each icon (the houses are always the same)

6. Modeling Approach and Results

Our main objective is to generate temporally coherent frames to create a video given the first frame and a textual description of the action occurring (caption). To achieve this objective, we first analyze simple text-to-image results on our dataset to assess the validity and difficulty of the problem, and then implement two novel architectures for frame prediction - both inspired by U-Net, with one incorporating an adversarial training situation.

6.1. Initial Experiments: Text to Image Synthesis (GAN)

In recent years, generic and powerful recurrent neural network architectures have been developed to learn discriminative text feature representations to perform text-to-image synthesis. Meanwhile, deep convolutional generative adversarial networks (GANs) have begun to generate highly compelling images of specific categories, such as faces, album covers, and room interiors. In this approach, we test the ability to generate plausible images from the most frequent one-line textual descriptions found in T-GIF, as well as from the human walking annotations of the KTH dataset.

This segment of our research was an initial experiment to observe the practicality of converting a text sequence to an image output. We validate our hypothesis that text-to-image synthesis is a complex problem.

Problem

The main issue with text-to-image synthesis is that the distribution of images conditioned on a text description is highly multimodal, in the sense that there are many plausible configurations of pixels that correctly illustrate the description. The complexity of this conditional multi-modality makes it a well-suited problem for GANs, in which a generator network is optimized to fool an adversarially-trained discriminator into predicting that synthetic images are real.

Data

1) The data used in this stage was a subset of the Tumblr data. The most frequent trigram observed was “a man is”. To reduce the abstractness of the task but retain a large amount of data for training, we filtered the dataset to captions that contain this trigram. The size of the dataset after filtering in this manner was reduced from 120K to 9K. In addition, only the first frame of the GIFs was used since we were interested in generating a single image from text at this stage.

2) However, for a more curated attempt, we used 400 images by choosing two classes from the KTH Human Action Dataset: “a man is walking left” and “a man is walking right”.

Note: Most GANs used for text-to-image synthesis are trained on multiple captions for a single image. Here, the datasets we used have only one caption; this increases the complexity of the task.

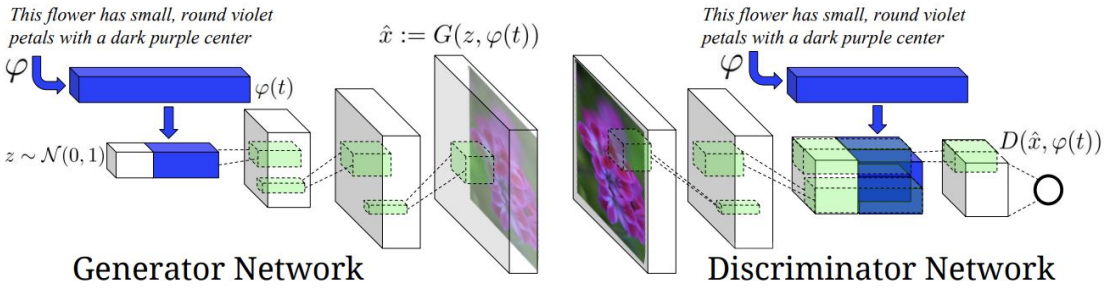


Figure 2: *Illustration of the Text-Conditional Convolutional GAN architecture from Reed et al. [2016]: “Generative Adversarial Text to Image Synthesis”*

In the generator G, first we sample from the noise distribution and encode the text query. The description embedding $\phi(t)$ is compressed to a lower dimension using a fully-connected layer and then concatenated to the noise vector z. Following this, inference proceeds as in a normal deconvolutional network: we feed it through the generator G to produce a synthetic image. Image generation corresponds to feed-forward inference in the generator G conditioned on query text and a noise sample. In the discriminator D, we perform several stride-2 convolutions with spatial batch normalization followed by a leaky ReLU. We again reduce the dimensionality of the description embedding, then perform convolution followed by rectification to compute the final score from the discriminator.

Results

In many cases, the first frame of an image sequence used for training did not contain the subject of interest. This was a problem as we sought to train a model to generate the first frame of an image sequence conditioned on a caption. To account for this, we filtered the 9K subset of first frames through Facebook's Mask R-CNN Object Detection Model[1] to check for the presence of a person. This helped us increase the accuracy of the results.

The aforementioned Object Detection algorithm was trained to identify over 70 objects in the given image. However, we filtered this to output a binary flag which indicated the presence of a human being in the image.

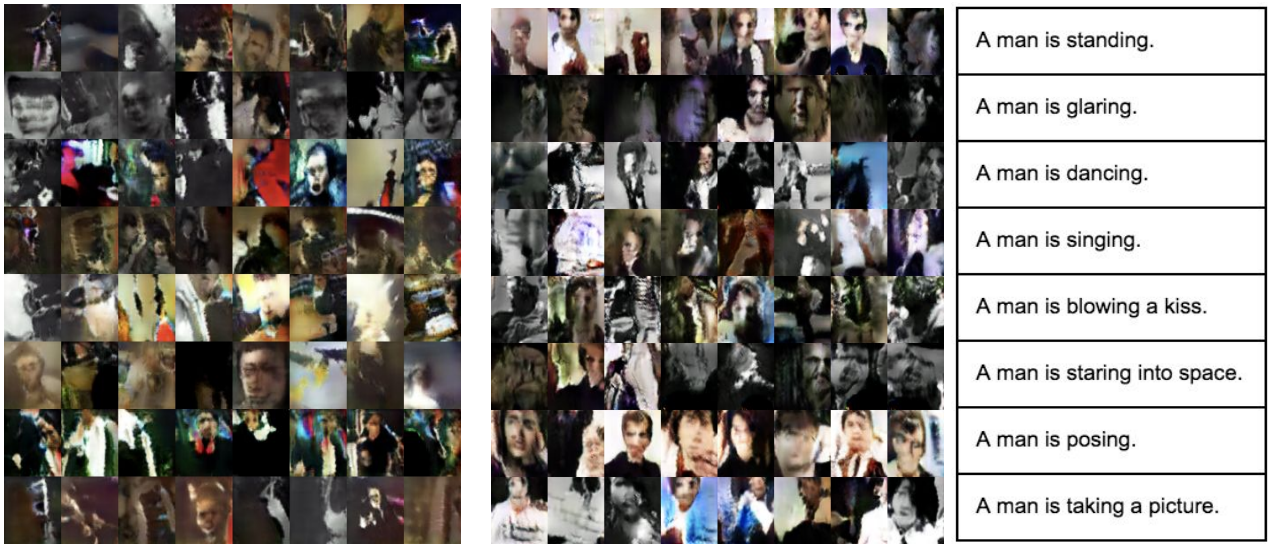


Figure 3: *Example results obtained from our text-to-image generator. Each row corresponds to the input sentence shown on the right. (a) Before CNN Object Filtering. (b) After CNN Object Filtering.*

The outputs of the GAN from the KTH Human Action Dataset can be found in Figure 4.

[1] From Facebook AI Research, Mask R-CNN paper: "The Region-based CNN (R-CNN) approach to bounding-box object detection is to attend to a manageable number of candidate object regions and evaluate convolutional networks independently on each Region of Interest (RoI). R-CNN was extended to allow attending to RoIs on feature maps using RoI Pool, leading to fast speed and better accuracy. Faster R-CNN advanced this stream by learning the attention mechanism with a Region Proposal Network (RPN). Faster R-CNN is flexible and robust to many follow-up improvements and is the current leading framework in several benchmarks." (https://github.com/facebookresearch/Detectron)

Figure 4: *Example results obtained from our text-to-image generator after being trained on the KTH Human Action Dataset. Each row corresponds to the input sentence shown on the right.*

Comments

As we can observe from the results above, the performance on the filtered KTH action dataset is better. The images are more lifelike and distinguishable than the ones from the Tumblr dataset given its higher specificity. The KTH dataset, however, is particularly challenging in the text-to-image synthesis setting because of the similarity in the pose of a human moving left or right. Overall, the frame produced is similar to the distribution of the underlying data, but it is hard to determine if the person is walking left or right.

Thus, we observed that producing an image output from a text input is a realistic goal but only when implemented with more constrained data. Our initial hypothesis of problem complexity is proven right.

6.2. Text + Image to Video Synthesis (U-Net)

Inspired by the success of U-Net, an encoder-decoder architecture with skip-connections proposed for biomedical segmentation, we adapt a similar base structure for our problem. We extend U-Net to operate on 3D-volumes instead of single images as input. A 3D-volume here represents previous frames in the GIF sequence. We then incorporate the text captions by embedding a network-in-network: the encoder first transforms the input volume into a compact latent representation. We flatten the resulting volume and feed it into a series of densely connected layers. We also feed in a caption representation at this point. The output of this “network-in-network” is reshaped to a 3D-volume, and subsequently upsampled by the decoder. Our output is a single image which represents the next frame.

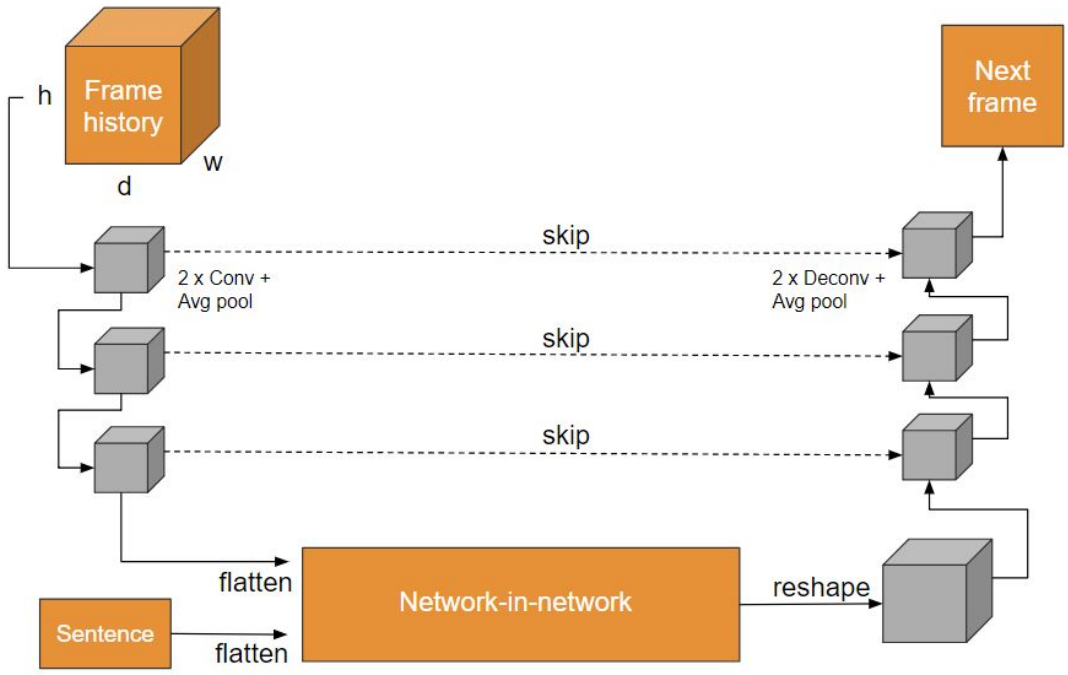


Figure 5: *Our altered U-Net architecture, taking a 3D-volume filled with images and a sentence representation, and outputting a single next frame prediction.*

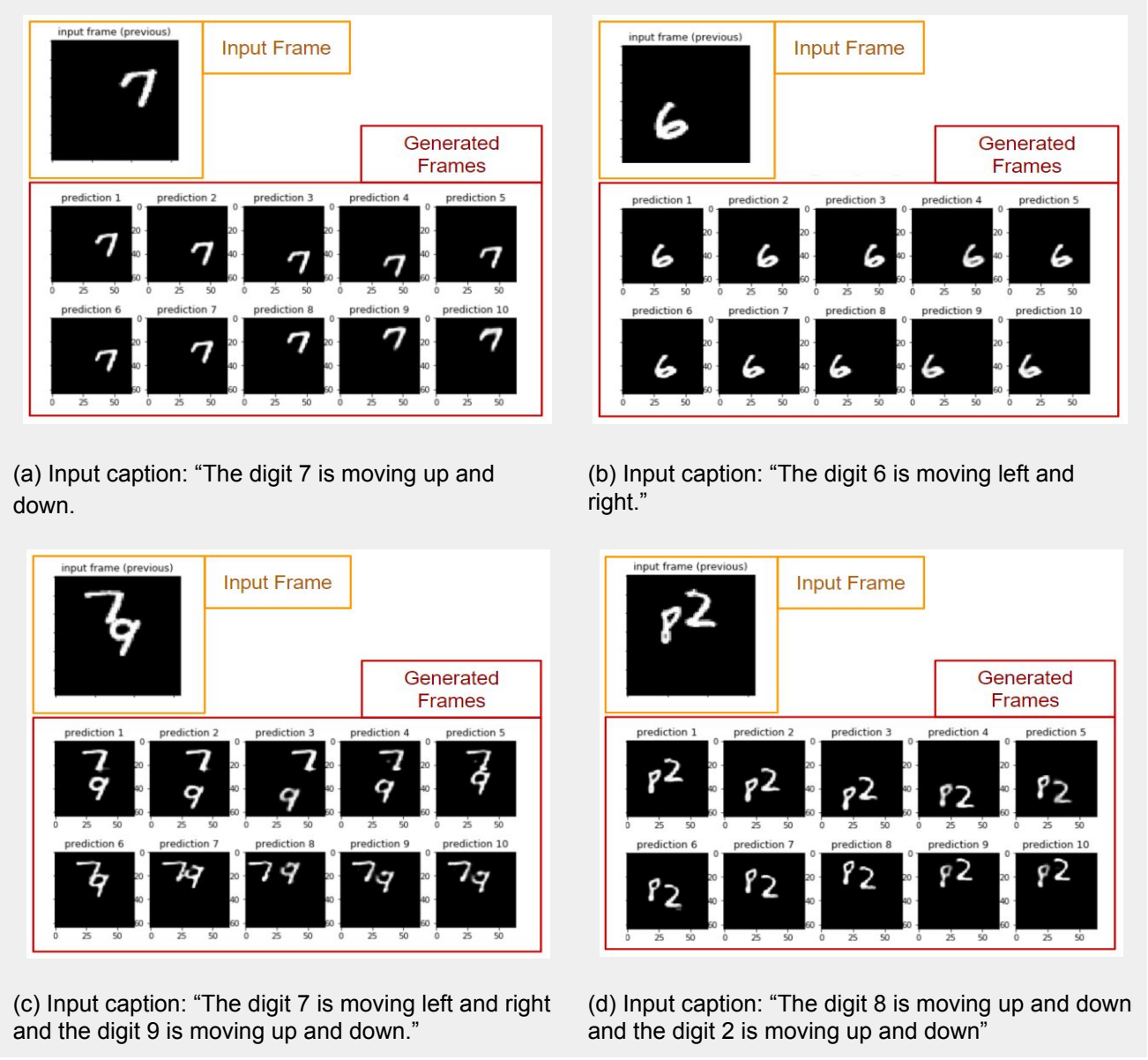
A graphical representation of our network is shown in Figure 5. The idea behind the structure is as follows: given the current frame and history, we first use the encoder structure to get a condensed representation of the image contents. Through our use of fully-connected layers in our internal network, we have a high level of expressiveness to manipulate image contents based on the sentence inputs. However, once it is manipulated, we are left with a highly downsampled and low-quality image. Here, the decoder comes into play and the skip-connections render themselves crucial: through them, the upsampling can re-include lost detail in the images to produce high-quality outputs. Following this logic, we can think about our encoder as bringing our input to a higher-level, more abstract representation, our network-in-network architecture as image transformer, and our decoder as quality and detail enhancing entity.

We perform experiments on our icons-in-motion and MNIST-in-motion datasets, as well as the KTH Human Action Dataset for walking. To train our network, we feed in the textual annotation, the current frame and up to d - 1 preceding frames, if available. The network is then trained to output the next frame. During inference, we just provide an initial frame and the textual description to predict the new frame, and then repeatedly feed in the previous predictions to predict further into the future. We usually execute this prediction cycle for between five and ten times, but it can be followed an indefinitely number of times.

We reached convergence within a few minutes, but found our architecture to be prone to overfitting, which we resolved through increased regularization. For our experiments, we set h = w = 64 for synthetic data and h = w = 128 for the activity data. We always choose d = 10, which is a reasonable

amount of history available for the network. Our sentences are simply represented as one-hot encoded matrices. We show some qualitative results in the following section.

Results



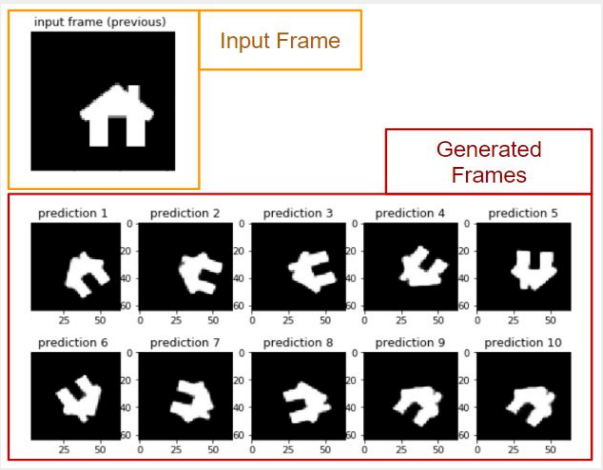
(a) Input caption: “The digit 7 is moving up and down.

(b) Input caption: “The digit 6 is moving left and right.”

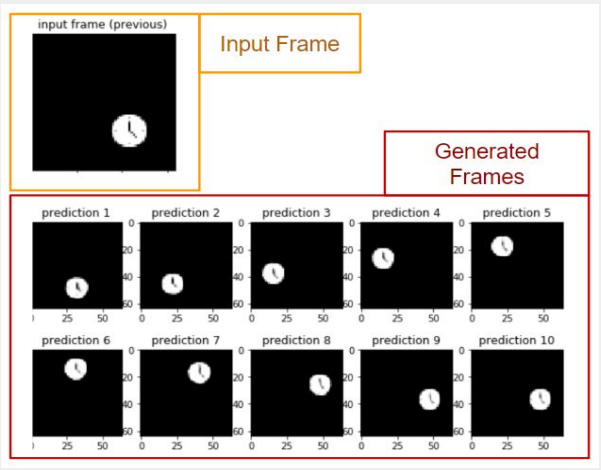
(c) Input caption: “The digit 7 is moving left and right and the digit 9 is moving up and down.”

(d) Input caption: “The digit 8 is moving up and down and the digit 2 is moving up and down”

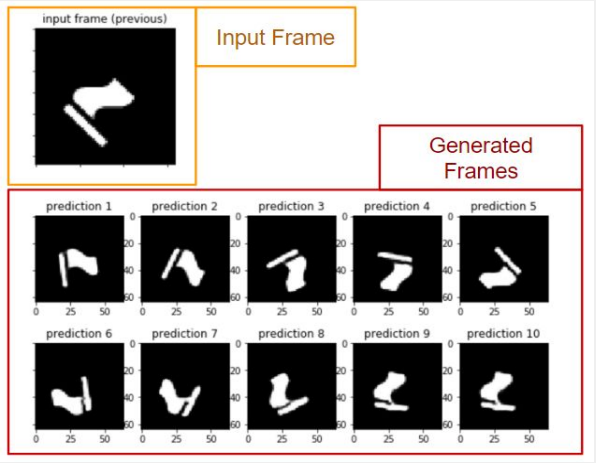
Figure 6: *Results on the MNIST-in-motion dataset. Our network manages to generate highly accurate frames, following the given caption and correctly maintaining the shape of the digits. On the more complex, 2 digit dataset, we also observe correct movement prediction.*



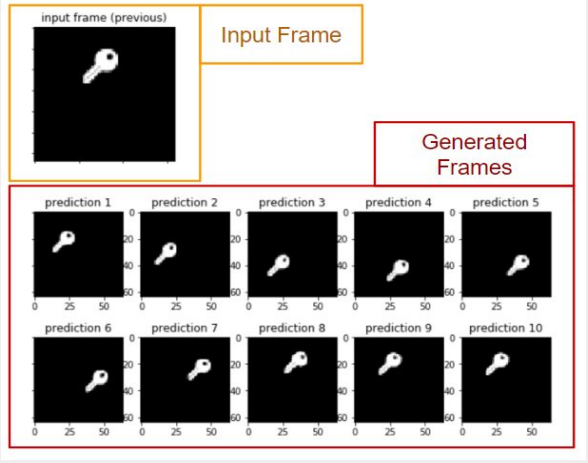
(a) Input caption: “The house is rotating in place counterclockwise.”



(b) Input caption: “The clock is rotating clockwise.”

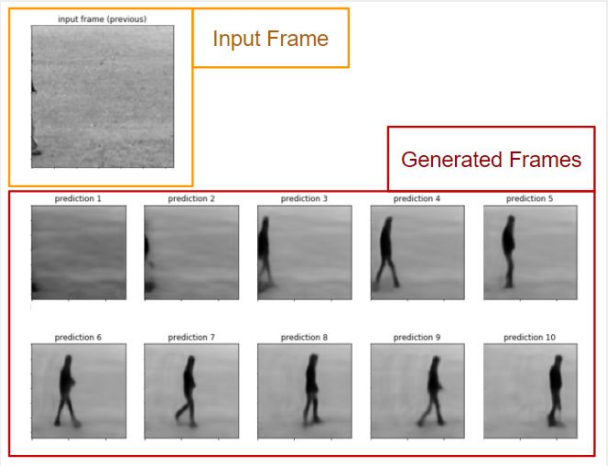


(c) Input caption: “The flag is rotating in place counter-clockwise.”

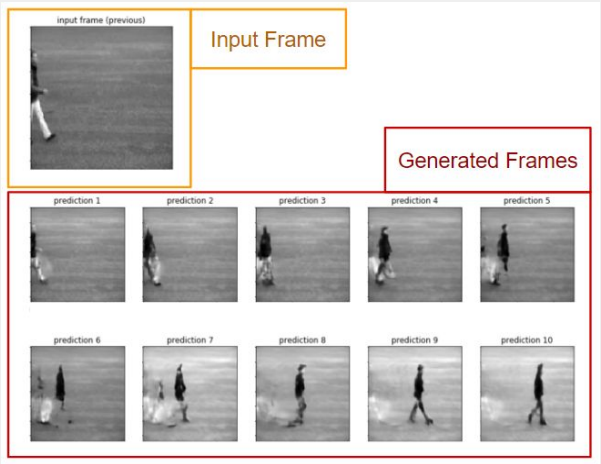


(d) Input caption: “The key is rotating counter-clockwise.”

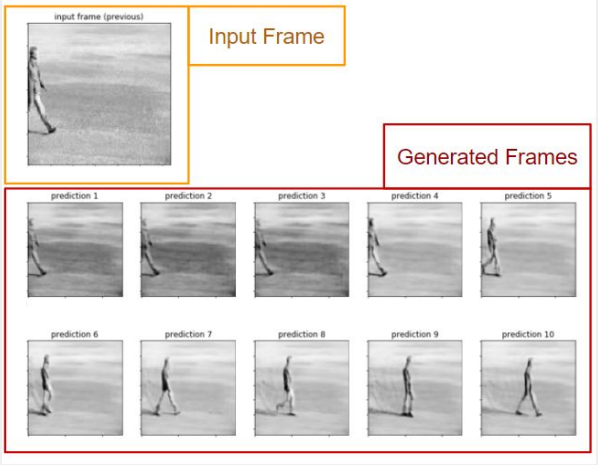
Figure 7: *Results on the Icons-in-motion dataset. We highlight rotation movements: rotating around its axis and rotating around the center of the image. Again, our network manages to generate accurate frames and follows the caption correctly. Even the small details are preserved: note the hole of the key and the clock hands.*



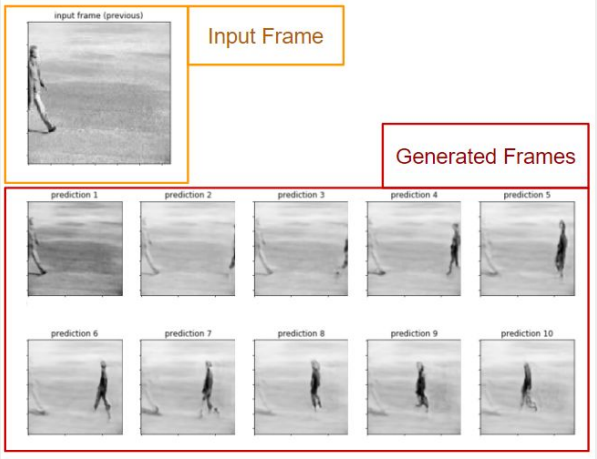
(a) Train set, “A person is walking right.”



(b) Test set, “A person is walking right.”



(c) Test set, “A person is walking right.”



(d) Test set, “A person is walking left.”, makes up a new person

Figure 8: *Results on the KTH dataset. Results on training set are realistic, while results on test set present some interesting artifacts like in fig. b, where the pants of the person walking change color from white to black. In fig. d, we observe what happens when we give the opposite caption to a starting frame: a new person is imagined.*

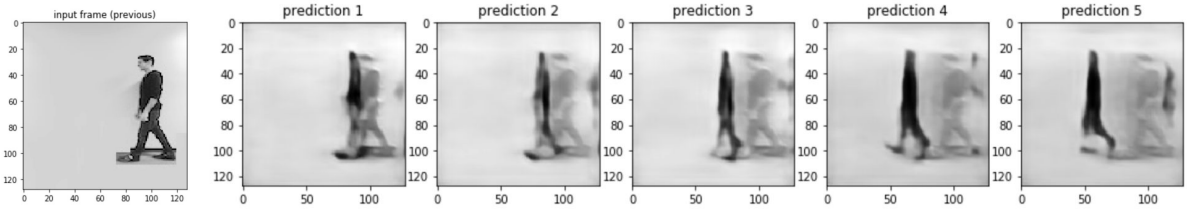


Figure 9: *We observed our results on a truly out of sample image: a picture of one of our team members, in front of a white wall to generate a similar background to the training set.*

Comments

On the Synthetic I datasets (Icons and MNIST, one per GIF), the results were surprisingly good: we manage to generate the correct video with very high fidelity. On the Synthetic II dataset (2 elements per GIF), we achieve good results except in some cases where the two digits merge during their movement. The network has trouble distinguishing between the two merged elements in subsequent frames.

On the KTH dataset, the networks performs very well on the training set, generating almost photo-realistic images. On the test set, the results are still very positive, although artifacts and a generally higher level of blur are present. It is interesting to notice what happens for subplot (d) of Figure 8: we take a starting frame corresponding to the caption “a man walks right”, but we input the caption “a man walks left”. The network then imagines a person dressed in black coming from the right and slowly blurs away the person coming from the left. This confirms that our network is indeed using the captions to determine its output.

In Figure 9, we show an extreme out-of-sample example, and we see that our network has some trouble generalizing to such an alien frame (which is to be expected). We observe some difficulty in generating truly realistic frames, and we see some “ghosting”, (persistence of the input image in subsequent frames). The results are nonetheless promising.

6.3. Text + Image to Video Synthesis (U-Net + Discriminator)

In an attempt to increase our network’s generalization performance, we added a discriminator network at the end of our frame-generating U-Net network. The idea here is as follows: we want to imitate movement in a realistic manner, but that doesn’t necessarily mean that we need to perfectly reproduce the frames of our training set. In fact, as long as the generated frame is temporally coherent and consistent with the caption, we accept it as a valid frame. When the problem is phrased like this, it becomes less obvious what the best loss function is. In fact, MSE might not be the best choice, as it draws the network towards reproducing as best as possible the training example - this somewhat over-constraints the output, and that leads to difficulty in training and overfitting.

In this situation, a discriminator could work well, as it may be thought as a network approximating a loss function. That is due to the adversarial setting, as explained in the introduction: we build a network to recognize real frames from fake ones, and that drives the network to output a measure of “validity” of any input frame. That validity is directly based on the real examples that were shown to the discriminator, and is therefore a dynamically changing loss adapted to the data we want to reproduce.

We added a discriminator working at a frame level as shown in Figure 10.

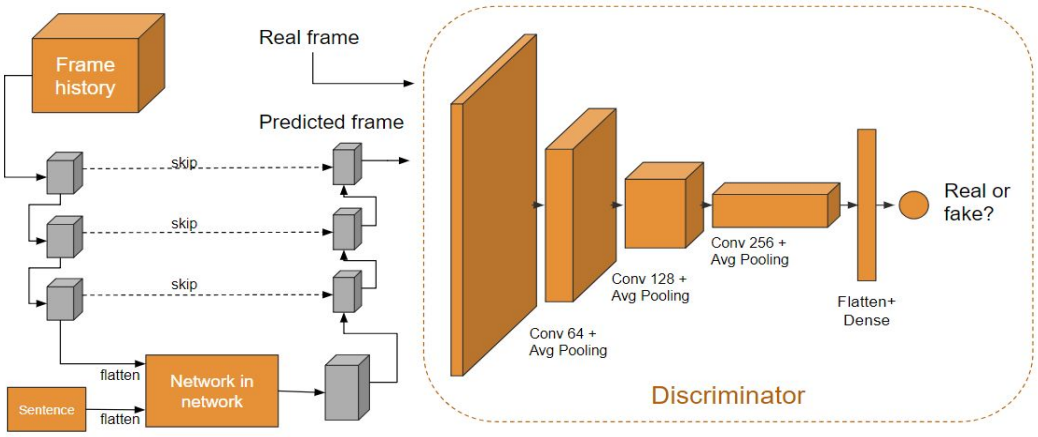


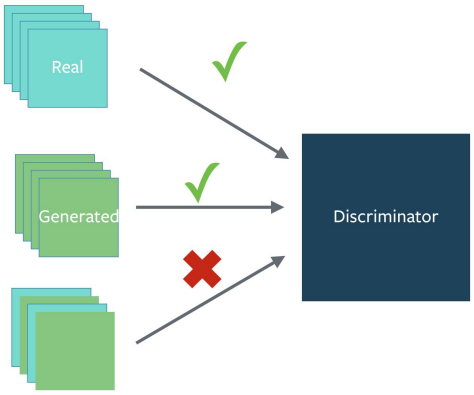
Figure 10: *U-Net + discriminator architecture. Each orange block of the discriminator represents the output of a 2D convolution, batch normalization, dropout with probability 0.1 and average pooling.*

Training: The discriminator was trained with batches comprised half-and-half of real frames and generated frames. When training the full architecture, we used a batch-size of 8. Note that the discriminator only takes one frame as an input, and outputs a value between 0 and 1 indicating the probability that the given frame is from the real set of frames. We tested multiple approaches to correctly train this system:

1. First, we tried random initializations, no batch normalization and max-pooling on the discriminator, and we obtained very poor results.
2. We then switched to pre-trained networks - we used a version of U-Net that already achieved good results at frame prediction from the previous experiments (early stopping version), and managed to improve its performance. We balanced the discriminator before starting training by pre-training it for 2000 iterations, showing both real frames and generated frames. We found that this pre-trained setting achieved much better results, but we observed several unwanted artifacts in later epochs.
3. To get rid of these, we devised a final training setting, where we alternated between updating the U-Net through the discriminator and updating it with MSE loss to maintain pixel-level coherence. The alternation was made every 100 iterations. This final approach yielded the best results.

Training a network in an adversarial setting is tricky. The generator can exhibit mode collapse, the discriminator can become too good and limit generator learning, and the learning rates have to be carefully tuned to avoid unbalancing the system. We managed to make our architecture work with learning rates of $2 \cdot 10^{-5}$ for the discriminator and 10^{-5} for the generator, updating the discriminator 5 times for each generator update, and tuning the dropout rate of both networks, ending up with .5 for the generator and .1 for the discriminator. We also found it extremely important to pretrain the discriminator for at least 1500 iterations (each iteration with a batch size of 16, 8 real and 8 fake frames) to achieve good results. We found the following training tricks very helpful:

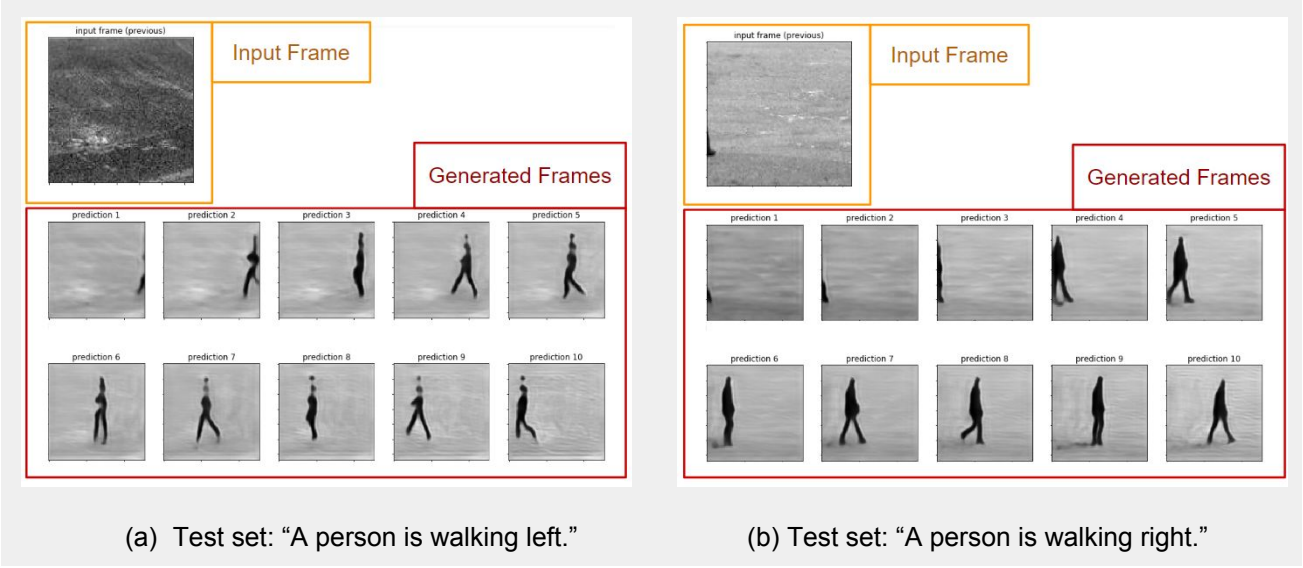
1. When training the discriminator, construct different minibatches for fake and real data - don't mix. This makes the most out of the BatchNormalization layer, normalizing images from the same category. We implemented this with batch sizes of 8 for real and fake data.



2. Avoid sparse gradients - move away from ReLU and MaxPool if possible. We used LeakyReLU and AvgPool on our discriminator. We did observe an increase in performance when these changes were added.
3. Use dropout in the generator: we used a generous amount of dropout (50%). This regularizes strongly the generator and avoids mode collapse.

Results

We show some results obtained with U-Net + discriminator in Figure 11 and 12. As can be seen, the results are positive and the movement of a person is correctly captured by the generator. We observe that although the images are still lacking high-resolution detail, we are able to generalize positively to the test set.



(a) Test set: "A person is walking left." (b) Test set: "A person is walking right."

Figure 11: *Results of the adversarial architecture on test set images. As can be seen, the results on the test set are good but still somewhat blurry.*

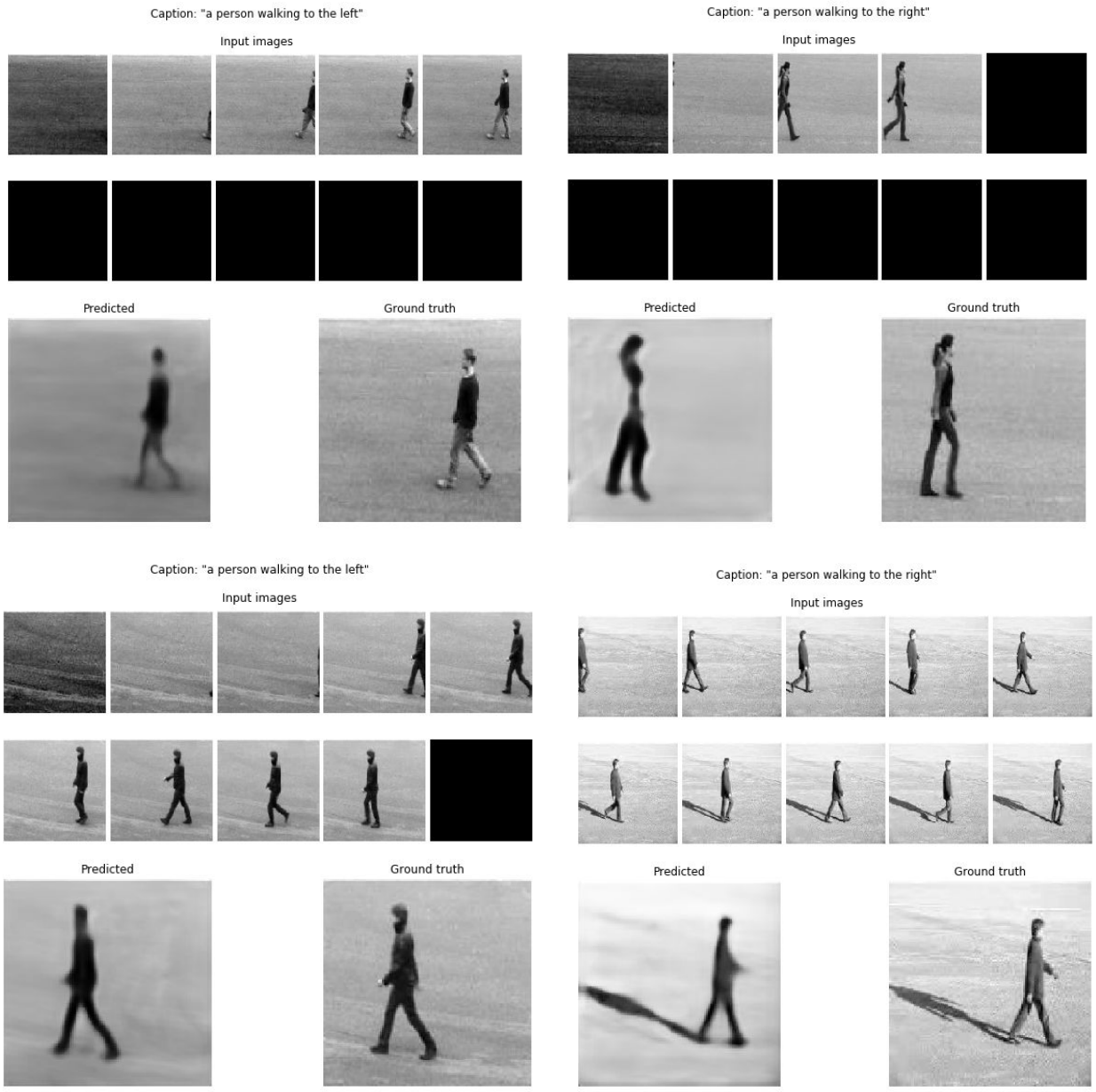


Figure 12: *Comparison between predicted frames and ground truth for 4 examples. At the top of each image, we show the input frames that went into the network. The black images correspond to padding (zeros), which are necessary because the network has a fixed input length of 10 frames.*

7. Conclusions and Future Work

As expected, we found the task of image sequence generation to be particularly challenging. For our initial experiments and validation purposes, we found generating synthetic datasets to be immensely

helpful, since they allowed for the generation of arbitrarily diverse, complex and large datasets in a controlled environment. Making extensive use of our synthetic data generation framework, we generated two main classes of datasets: Icons-in-motion and MNIST-in-motion, consisting of a variety of icons or handwritten digits moving or rotating in space on a dark background. Starting from our easy synthetic datasets and moving towards higher complexity, we were able to appropriately refine our model design choices in order to make operating within the more challenging domain of natural images attainable.

Our initial plan was to work extensively on the T-GIF dataset, noting its comprehensiveness, diversity and rich annotations. However, precisely because of this high variation in topics, scenes and captions, we found other datasets to be better suitable. In the case of natural images, we focused on the KTH Human Action Dataset, achieving very promising results for people walking.

We proposed two novel architectures that were weakly inspired by U-Net, a fully-convolutional segmentation network. Our first architecture consisted of a modified U-Net with a network-in-network, incorporating sentence representations to manipulate the hidden features. While we found the results in very early epochs to be slightly blurry, later epochs showed that it easily fully reproduces any training set we experimented with after very little training time. We found it relatively hard to regularize given its high number of parameters, but still find it to achieve promising results on unseen data.

For our second architecture, we make use of the same basic structure, but train the network in a truly adversarial setting. For this, we treat our basic network as generator, and extend it in a framework by a discriminator recognizing fake or real image predictions. We find that while we can not achieve the same amount of photorealism as in our first architecture, the generalization capabilities are improved. Clearly, being trained in an adversarial framework, it requires some tweaking and balancing to achieve convincing results.

Future work could mainly focus on different ways of regularizing our network as to generally improve generalization ability. However, the generalization ability would usually have to be assessed qualitatively, as there is not only one unique solution in image sequence generation, making such investigations subjective and complicated. Using our adversarial training framework, the role of a human judging the quality of results would be fulfilled by a suitable discriminator, and thus we believe future investigations should mainly focus on the adversarial setting.

Apart from technical follow-up work, the current approach could be benchmarked on even more diverse and comprehensive datasets such as other human action datasets.

8. References

1. Li, Y., Song, Y., Cao, L., Tetreault, J., Goldberg, L., Jaimes, A. and Luo, J., 2016. Tgif: A New Dataset and Benchmark on Animated GIF Description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4641-4650).

2. Mirza, M. and Osindero, S., 2014. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784*.

3. Pan, Y., Qiu, Z., Yao, T., Li, H., Mei, T., 2017. To Create What You Tell: Generating Videos from Captions.

4. Marwah, Tanya; Mittal, Gaurav; Balasubramanian, Vineeth N. Attentive Semantic Video Generation using Captions.

5. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B. and Lee, H., 2016. Generative Adversarial Text to Image Synthesis. *arXiv preprint arXiv:1605.05396.*

6. Tulyakov, S., Liu, M.Y., Yang, X. and Kautz, J., 2017. Mocogan: Decomposing Motion and Content for Video Generation. *arXiv preprint arXiv:1707.04993*.

7. Vondrick, C., Pirsiavash, H., Torralba, A., 2016. Generating Videos with Scene Dynamics.

8. Zhang, H., Xu, T., Li, H., Zhang, S., Huang, X., Wang, X. and Metaxas, D., 2017, October. Stackgan: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In *IEEE Int. Conf. Comput. Vision (ICCV)* (pp. 5907-5915).