Rewind: A Transcription Method and Website

Chase Carthen, Vinh Le, Richard Kelley, Tomasz Kozubowski, Frederick C. Harris Jr.
Department of Computer Science, University of Nevada, Reno
Reno, Nevada, 89557, USA
chase,vle@nevada.unr.edu, richard.kelley@gmail.com, tkozubow@unr.edu, Fred.Harris@cse.unr.edu

Abstract

Simple digital audio formats such as mp3s and various others lack the symbolic information that musicians and other organizations needed to retrieve the important details of a given piece. However, there have been recent advances for converting from a digital audio format to a symbolic format a problem called Music Transcription. Rewind is an Automatic Music Transcription (AMT) system that boasts a new deep learning method for generating transcriptions at the frame level and web application. The web app was built as a front end interface to visualize and hear generated transcriptions. Rewind's new deep learning method utilizes an encoder-decoder network where the decoder consists of a gated recurrent unit (GRU) or two GRUs in parallel and a linear layer. The encoder layer is a single layer autoencoder that captures the temporal dependencies of a song and consists of a GRU followed by a linear layer. It was found that Rewind's deep learning method is comparable to other existing deep learning methods using existing AMT datasets and a custom dataset. In other words, Rewind is a web app that utilizes a deep learning method that allows users to transcribe, listen to, and see their music.

1 Introduction

Many musicians, bands, and other artists make use of MIDI, a symbolic music instruction set, in popular software to compose music for live performances, portability across other formats, and recording. However, most music is often recorded into raw formats such as Wav, MP3, OGG, and other digital audio formats. These formats do not often contain symbolic information, but may contain some form of metadata that does not typically include symbolic information. Symbolic formats, such as sheet music have been used by bands, choirs, and artists to recreate or perform songs.

Automatic Music Transcription (AMT) is the process of converting an acoustic musical signal into a symbolic format [13]. There are a few music transcription applications having varying degrees of accuracy that have been built mostly for Windows, Linux, Mac and the web browser. Only a few of these applications of the ability to visualize the results of the transcription. A piano roll is an intuitive visualization of music that does not require a user to learn more complex symbolic available for music such as sheet music. These applications allow a user to get a symbolic format of their music that can be used for many different reasons such as changing a song, portability to other applications, live performances, and for generating sheet music. However, most of these applications do not use state of the art algorithms from advances in Deep Learning that have contributed to the Music Information Retrieval (MIR) field.

There has been recent work in the AMT field with [6, 5, 20] that have produced higher transcription accuracies than previous methods. These advances along with the creation of web audio frameworks such as WebAudio or WebMidi have made it possible to playback many different types of audio formats such as mp3, wav, and MIDI. Web frameworks such as Django and Flask make it possible to create a web application that does automatic music transcription and allows users to visualize the transcription and hear the results. Rewind [8] is a tool and method that will make use of a new Deep Learning method based on previous work, visualize the results of the transcribed file, and allow the user to edit the transcribed results.

The following paper is structured as follows: Section 2 covers background related to the MIR and Deep Learning field. Section 3 discusses the implementation and design of Rewind tool. Section 4 gives the results of the Rewind method. Finally Section 5 concludes and details future direction that Rewind can take.

2 Background

AMT systems are created to make transcriptions at different levels of detail in music being the stream, note, and frame level [13]. At the stream level one transcriptions are created based on a audio digital

978-1-943436-05-7 / copyright ISCA, SEDE 2016
September 26-28, 2016, Denver, Colorado, USA

format. The goal of the frame level is to capture all pitches within each frame provided by a spectrogram. At the note level a set of notes are used to generate a new set of notes or note tracking. Most AMT systems evaluate their effectiveness by means of various metrics, which include recall, accuracy, precision, and f-measure [3]. These metrics are calculated with true positives, false positives, and false negatives.

Precision determines how relevant a transcription is given irrelevant entries in a frame. It is defined as follows:

$$Precision = \frac{\sum_{t=1}^{T} TP(t)}{\sum_{t=1}^{T} TP(t) + FP(t)} \quad (1)$$

Recall is the percentage of relevant music transcribed, and is given by Equation 2.

$$Recall = \frac{\sum_{t=1}^{T} TP(t)}{\sum_{t=1}^{T} TP(t) + FN(t)} \quad (2)$$

The accuracy determines the correctness of a transcription, and is given by equation 3.

$$Accuracy = \frac{\sum_{t=1}^{T} TP(t)}{\sum_{t=1}^{T} TP(t) + FP(t) + FN(t)} \quad (3)$$

While the F-measure determines the overall quality between the precision and recall.

$$F - measure = \frac{2 * precision * recall}{precision + recall} \quad (4)$$

There has been some work using LSTMs and semitone filter banks to transcribe music [5]. In Sigtia's work [20], the idea of an acoustic model converting an audio signal to a transcription is introduced. Additionally this paper introduces using a music language model to improve the accuracy of a transcription of a acoustic model like Boeck [5] and others as well. Boulanger-Lewandowski [6] uses a deep belief network to extract features from a spectrogram and utilizes a rnn to create a transcription along with a innovative beam search to transcribe music. Boulanger-Lewandowski's beam search is possible thanks to the generative properties of the deep belief network that is merely a collection of restricted Boltzman machines or RBMs that are stacked. This beam search is also utilized in combination with recurrent neural network with an neural autoregressive distribution estimator (rnn-nade) as a music language model and an acoustic model that uses a deep neural network. A follow-up paper produces a hash beam search that finds a more probable transcription in fewer epochs [19]. Both the beam search and hash beam search produce the most accurate transcriptions.

Recently, encoder-decoder networks have been used for unsupervised learning in terms of autoencoders [21], translation [11], caption generation for images, video clip description, speech recognition [10, 12] or video generation. Autoencoders, like an encoder-decoder network, are commonly used for unsupervised learning to learn features contained inside the data, by using the identity of the data. An autoencoder is powerful for learning features contained within a dataset. However, there are more complex encoder-decoder networks [11, 10, 12], where they learn a context and then map English to French. They are less concerned with learning the identity and more for learning the context of the data presented. Rewind utilizes these type of encoder-decoder networks to learn an encoding for a spectrogram presented to it. An example layout of this network is demonstrated in Figure 1. These networks have proven to be beneficial, and are state of the art.

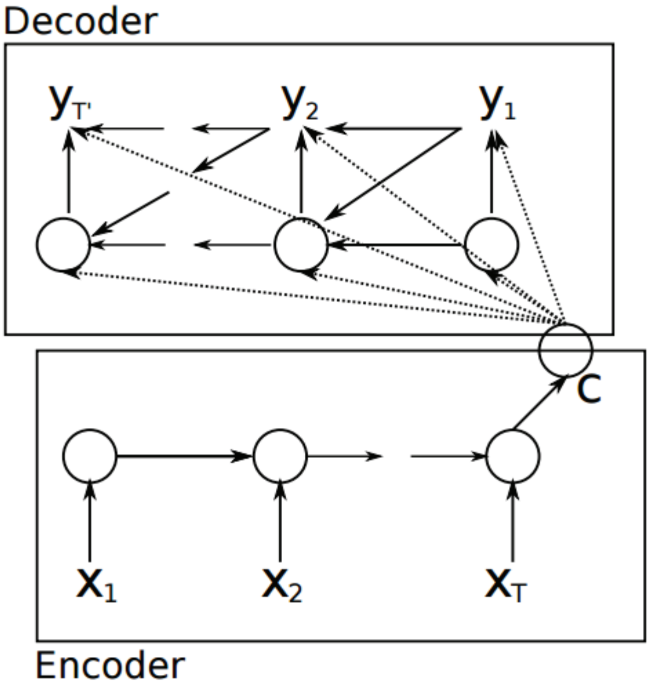

Figure 1: A picture of a encoder-decoder network with a context C demonstrated between the encode-decoder network [10].

3 Rewind

Rewind is very much like other AMT systems in that it determines the fundamental frequencies of the notes and what notes are active at the frame level. Like most other frame based systems, Rewind utilizes a spectrogram as its main input and a ground truth midi as the target. All audio samples are constructed

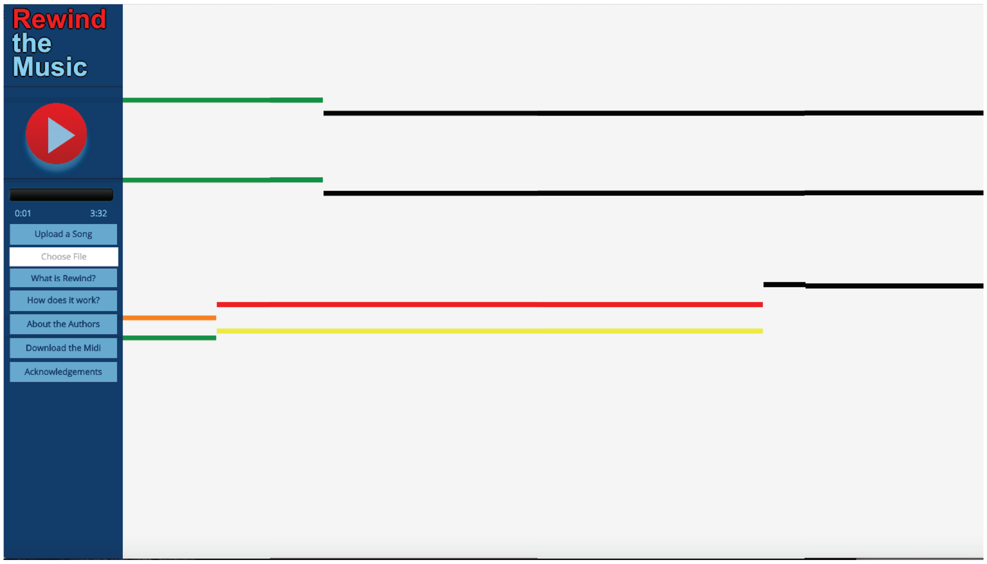

Figure 2: A screenshot of piano roll notes lighting up.

at a 22 kHz sample rate and turned into a normalized spectrogram with a 116 ms window size, being either a 10 ms stride or a 50 ms stride. It has been found that a window size larger than 100ms produces the most accurate results with a rnn-lstm [5]. A multitude of existing datasets were utilized for training Rewind's models: Nottingham [1], JSB Chorales [2], Poliner and Ellis [18], Maps [14], MuseData [9], and Piano.midi.de [15]. All of these datasets were split into 70% for training, 20% for testing, and 10% for validation. These datasets consisted of midi only or midi with aligned audio and made into datasets with timidity, Torch's audio library, and a midi library [4]. Rewind's models were implemented with rnn [17] and optim. A simple auto-correlation method was also constructed as a way to implement Rewind's web service and web site for quick testing. The auto-correlation is also compared against the encoder-decoder network. Rewind has two types of models: the encoder and the decoder model. The encoder and decoder is very similar to the encoder-decoder network in Figure 1 [10, 11, 12]. The encoder model of Rewind utilizes an autoencoder, which utilizes a single GRU for its encoder, whose output is squashed by a rectified linear unit and a linear layer for its decoding layer. While the decoder model has an identical layout, but its outputs are squashed with a sigmoid activation function and may have a second GRU in parallel.

The encoder network utilizes an autoencoder to create an encoding for spectrograms. An autoencoder was chosen because a deep neural network (stacked auto encoders) has been used for extracting features from spectrograms in the case of speech recognition [7] and other similar works that utilize deep belief networks (stacked restricted Boltzman machines) have been used to extract features [16]. A deep belief network, along with an autoencoder, are used to produce a generative model for spectrograms [12]. The generic representations generated by autoencoders can be further improved with recurrences [21], where the encoder and decoder of the autoencoder are both LSTMs for learning over video sequences and generating video sequences. Rewind's encoder model utilizes a linear neural network for the decoder and a GRU for the encoder with a rectified linear unit (ReLU) for it's activation function [21]. The encoder network is trained with a mean squared error function.

The decoder network consists of two types of networks being a GRU with a linear layer and two GRUs stacked onto of each in parallel with a linear layer. Both types of networks are squashed with a sigmoid function. The GRU in both networks was chosen because it produced the lowest error rate. This network's objective function is binary cross entropy, so that this decoder network will learn a distribution of notes where a probability of one indicates a note on and a probability of zero indicates a note off. Binary cross entropy is used for minimizing the log probability [6, 20], which also utilizes a sigmoid function to create a binary probabilities. The binary cross entropy function is demonstrated in Equation 5, where the sum is taken over all distributions [20]:

$$\sum_i t_i \log p_i + (1 - t_i) \log (1 - p_i) \tag{5}$$

The probabilities constructed from the sigmoid function can be used to construct a MIDI, and are utilized in

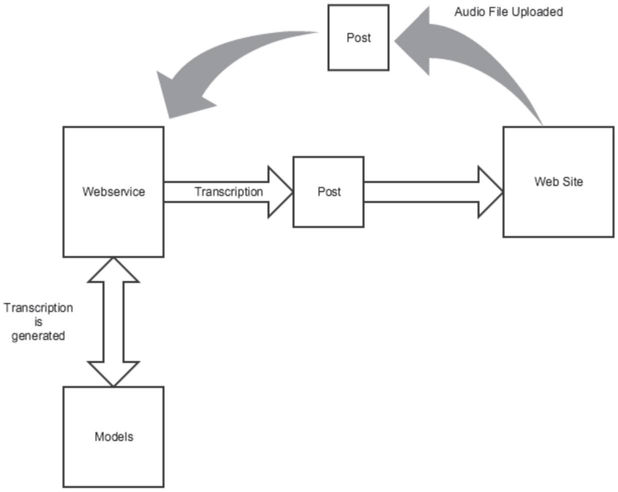

Figure 3: A diagram of Rewind's web service.

previously mentioned papers. The decoder network's job is to generate these probabilities for each encoding passed by the encoder network.

The auto-correlation method is a very noisy method. The process creates a spectrogram of the required audio file and then each bin of the spectrogram is normalized with the standard deviation and mean. After these transformations have been made, a threshold is applied, where anything greater than the threshold is a 1 and anything less is a 0. Subsequently, one simply only needs to go to each frequency bin that matches a midi note and extract the frequencies that have a value of 1. This auto correlation method is only meant as a test model for a web service. However, in Section 4, results are reported for its accuracy in comparison to Rewind's Network.

3.1 Web Site and Web Service

Rewind's web service was implemented in Flask as a small web service that could be utilized by Rewind's server for making transcriptions of uploaded audio files. All audio files and transcriptions are sent through post requests. Figure 3 demonstrates a diagram of the communication of audio files and transcriptions going in and out of the web service. This web service communicates with the models of Rewind and creates a midi file from the passed in audio file. All transcriptions generated by the web service are piano only. This is meant to make Rewind scalable for other web apps and servers.

Rewind's website was implemented in the Django web framework and utilized the following javascript libraries: remodal, jQuery, jQuery UI, and midi.js. Django was chosen for Rewind because it allows Rewind to be scalable for future web apps to be developed, easy database integration, and easy incorporation of security. Midi.js is utilized for its ability to parse MIDI files and generate sounds for those MIDI files. The jQuery and jQuery UI libraries has many useful features for designing interfaces, doing different web requests, and other functionality. The remodal library allow for modal windows to be displayed on the website. A small web service was implemented in Flask to wrap Rewind's models that could be utilized by Rewind's to generate transcriptions through get and post requests. This service was implemented so that the small web service could be used in other applications if needed. These libraries have made it possible to make a website for Rewind. An example of Rewind's website is demonstrated in Figure 2. This figure also demonstrates Rewind's ability to visualize the playback of a midi file in the form of a piano roll where the colors denote the note level. The user has the ability to scroll through the piano roll using the time bar and inspect the piano roll validity.

4 Results

In this section we present the precision, recall, f-measure, and accuracy of Rewind's transcriptions on the following datasets: Nottingham consisting of 1000 or more songs, JSB Chorales consisting of 200 or more songs, Poliner-Ellis consisting of 30 songs, MuseData consisting of 700 songs, the Maps dataset consisting of 169 songs, and a custom dataset that consists of 160 songs split evenly from country, rock, jazz and classical. The custom dataset was added since all of the benchmark datasets currently used in the AMT are currently only classical piano music and orchestral music. All datasets are primarily midi and a synthesizer is used to generate wav except for the Poliner-Ellis and Maps dataset that have a aligned wav file and midi file. In Table 1 and Table 2, the overall results of Rewind at a 10 ms stride, a standard for AMT systems, at the frame level are demonstrated and compared to Boulanger-Lewandowski's work [6, 19]. The 50 ms results are demonstrated in Table 3, but the results are not reported for the maps dataset. The 10 ms stride results were trained with two parallel GRUs with a linear layer and the 50 ms results were trained with a single GRU and linear layer. The results demonstrated in Table 2 are compared against ConvNet acoustic model at the frame level [19].

Upon examining the table, the Convnet is better overall in accuracy, recall, and f-measure, but Rewind has the higher precision. The ConvNet [19] utilizes a

Table 1: Rewind's results at 10 ms stride for the spectrogram (1 is the proposed model and 2 is the rnn-nade [6]).

	Accuracy		Precision		Recall		F-Measure	
Models	1	2	1	2	1	2	1	2
Nottingham	95.1%	97.4%	98.0%		96.9%		97.5%	
JSB	82.8%	91.7%	34.4%		88.8%		82.8%	
Poliner-Ellis	34.4%	79.1%	66.9%		41.5%		34%	
MuseData	34%	66.6%	56.8%		45.9%		50.8%	
Custom	16.2%		51.1%		19.2%		27.9%	

Table 2: Rewind's performance on the Maps dataset compared to [19] at 10 ms.

	Proposed	Simple Auto-Correlation	ConvNet[19]
Accuracy	51.6%	6.4%	**58.87%**
Precision	**76.5%**	21.8%	72.40%
Recall	61.4%	8.2%	**76.50%**
F-Measure	68.1%	11.2%	**74.45%**

Table 3: Rewinds results at a 50 millisecond stride for the spectrogram where 2 is the proposed model and 1 is the Simple Auto-Correlation model.

	Accuracy		Precision		Recall		F-Measure	
Models	1	2	1	2	1	2	1	2
Nottingham	21.5%	94.0%	29.2%	97.9%	44.7%	95.9%	35.3%	96.9%
JSB	20.8%	81.6%	32.9%	92.1%	36.2%	87.7%	34.5%	89.9%
MuseData	11.8%	23.0%	15.8%	60.2%	31.9%	27.2%	21.1%	37.4%
Poliner-Ellis	6.6%	42.6%	17.7%	70.5%	9.7%	51.8%	12.5%	55.8%
Custom	8.5%	20.4%	12.2%	44.5%	21.8%	27.3%	15.6%	33.9%

hash beam search to find the most probable sequence. If Rewind was to utilize the same hash beam search, it may have been able to achieve an even better accuracy, recall, and f-measure.

5 Conclusions and Future Work

Rewind demonstrated a encoder-decoder network that is comparable to the results of Boulanger-Lewandowski rnn-rbm [6] in terms of the Nottingham and JSB dataset. It also achieved a higher precision than the rnn-nade [19] on the Maps dataset. However, it suffered from issues in connection with choosing a threshold to generate an on value in the transcription on datasets such as MuseData and the custom dataset built by Rewind. The custom dataset demonstrated that AMT systems can work with multiple genres, but there may be other factors that cause transcription metrics to go down, such as multiple instruments being existent in the song or an improper threshold. Despite these issue, Rewind does manage to follow the underlying frame distribution in the lower classified datasets. Rewind's encoder-decoder has demonstrated a model that has a high precision and comparable results coupled with a web app that can generate transcriptions. Rewind's web site provides users with a way to hear and see their transcriptions.

Rewind's web has the potential for new features and interfaces for new problems. Rewind could be be expanded into an application that allows a user to edit existing music that has been transcribed. Another addition would be to allow Rewind to recognize the lyrics of the music being played. One more thing that Rewind could provide is a way for users to collaborate and learn about music.

Acknowledgement

This material is based in part upon work supported by: The National Science Foundation under grant number(s) IIA-1329469. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] James Allwright. ABC version of the nottingham music database. 2016. URL: http://abc.

sourceforge . net / NMD/ (Last accessed 04/10/2016).

[2] James Allwright. Bach choral harmony data set. 2016. URL: http://archive.ics.uci.edu/ml/datasets/Bach+Choral+Harmony (Last accessed 04/10/2016).

[3] Mert Bay, Andreas F. Ehmann, and J. Stephen Downie. Evaluation of multiple-f0 estimation and tracking systems. In *Proceedings of the 10th international society for music information retrieval conference*. http://ismir2009.ismir.net/proceedings/PS2-21.pdf. Kobe, Japan, 2009, pages 315–320.

[4] Peter J Billam. MIDI.lua. URL: http://www.pjb.com.au/comp/lua/MIDI.html (Last accessed 04/10/2016).

[5] Sebastian Bock and Markus Schedl. Polyphonic piano note transcription with recurrent neural networks. In *Acoustics, speech and signal processing (ICASSP), 2012 ieee international conference on*, 2012, pages 121–124. DOI: 10.1109/ICASSP.2012.6287832.

[6] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. High-dimensional sequence transduction. In *Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on acoustics, speech and signal processing*, 2013, pages 3178–3182. DOI: 10.1109/ICASSP.2013.6638244.

[7] Nicolas Boulanger-Lewandowski, Jasha Droppo, Mike Seltzer, and Dong Yu. Phone sequence modeling with recurrent neural networks. In *ICASSP*. IEEE SPS, 2014. URL: http://research.microsoft.com/apps/pubs/default.aspx?id=217321.

[8] Chase D. Carthen. Rewind: a music transcription method. Master's thesis. University of Nevada, Reno, 2016.

[9] Center for Computer Assisted Research in the Humanities. Musedata. 2016. URL: http://musedata.stanford.edu/ (Last accessed 04/10/2016).

[10] Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. Describing multimedia content using attention-based encoder-decoder networks. 2015. eprint: arXiv:1507.01053.

[11] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Corr*, abs/1406.1078, 2014. URL: http://arxiv.org/abs/1406.1078.

[12] Li Deng, Mike Seltzer, Dong Yu, Alex Acero, Abdel rahman Mohamed, and Geoff Hinton. Binary coding of speech spectrograms using a deep auto-encoder. In *Interspeech 2010*. International Speech Communication Association, 2010. URL: http://research.microsoft.com/apps/pubs/default.aspx?id=135405.

[13] Zhiyao Duan and Emmanouil Benetos. Automatic music transcription. ISMIR. 2015. URL: http://c4dm.eecs.qmul.ac.uk/ismir15-amt-tutorial/.

[14] Valentin Emiya. Maps database - a piano database for multipitch estimation and automatic transcription of music. 2016. URL: http://www.tsi.telecom-paristech.fr/aao/en/2010/07/08/maps-database-a-piano-database-for-multipitch-estimation-and-automatic-transcription-of-music/ (Last accessed 04/10/2016).

[15] Bernd Krueger. Classical piano MIDI page. 2016. URL: http://www.piano-midi.de/ (Last accessed 04/10/2016).

[16] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in neural information processing systems 22*, pages 1096–1104, 2009. URL: http://books.nips.cc/papers/files/nips22/NIPS2009_1171.pdf.

[17] Nicholas Leonard, Sagar Waghmare, Yang Wang, and Jin-Hwa Kim. Rnn : recurrent library for torch. 2015. eprint: arXiv:1511.07889.

[18] Graham Poliner. Automatic piano transcription. 2016. URL: http://labrosa.ee.columbia.edu/projects/piano/ (Last accessed 04/10/2016).

[19] S. Sigtia, E. Benetos, and S. Dixon. An End-to-End Neural Network for Polyphonic Piano Music Transcription. *Arxiv e-prints*, 2015. arXiv: 1508.01774 [stat.ML].

[20] Siddharth Sigtia, Emmanouil Benetos, Srikanth Cherla, Tillman Weyde, Artur S. dAvila Garcez, and Simon Dixon. An rnn-based music language model for improving automatic music transcription. In *15th international society for music information retrieval conference (ISMIR 2014)*, 2014.

[21] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. 2015. eprint: arXiv:1502.04681.