Sponsoring Committee: Professor Juan Pablo Bello, Chairperson
Professor
Professor

A Proposal for

# MODERN GENERATIVE METHODS FOR MUSIC TRANSCRIPTION

Jong Wook Kim

Program in Music Technology
Department of Music and Performing Arts Professions

Submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in the
Steinhardt School of Culture, Education, and Human Development
New York University
2017

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

# CHAPTER I

# INTRODUCTION

Automatic music transcription refers to an automated method that can identify all musical events in the input audio and convert them into musical notations. The nature of music transcription is multifold; to create a complete transcription, one has to identify all instruments, beats, dynamics, and the pitch traces for every instrument present in the music, and it is still far from achieving the human-level accuracy despite decades of research in each of these subtasks. The need for study arises naturally, not only because this is an intriguing problem in the interdiscipline of music and technology that has remained unsolved for decades, but also because the solution to this problem can provide practical benefits to many applications including, but not limited to, music recommender systems, music search engine, and compositional aid software. The task of automatic music transcription shares many common values with other machine learning tasks, such as image segmentation, machine translation, and speech recognition, in a sense that the core task is to build an intelligent system that can extract and process semantics that are conveyed in complex signals. This is the essence of artificial intelligence (AI) — a system that perceives its environment and takes actions that maximizes the utility (Russell & Norvig, 2009) — where the signals coming from the environment is complex multimedia data and understanding the semantics of them is a necessity to per-

form well. Therefore, the problem of automatic music transcription is not just an intriguing task in music technology, but will also be a key component of the AI-enabled future society, constituting a musical brain of AI.

This thesis aims to design and develop improved methods for automatic music transcription with deeper computational understanding of musical semantics, leveraging the recent technological breakthroughs in the area of deep learning and big data. The idea more specifically focuses on deep generative models and learning from training data generated by various sources including software instruments, so that machine learning algorithms can benefit from an effectively infinite source of labeled training data. Another motivation for using generated audio data and generative models is the fact that synthesized music is more prevalent and perceptually more familiar to people than synthesized texts or pictures, suggesting that synthesized and generated audio will be able to more accurately model the distribution of real audio data to be transcribed. This generative approach also aligns with how real musicians transcribe music, where they match their knowledge of how the instruments will sound when played in a certain combination of rhythms and melodies, with given audio. Therefore it is reasonable to claim that machines should also be able to perform in a similar way, provided that a proper representation of knowledge about the music and instruments is available. To validate this claim as a feasible research direction, this proposal provides a literature review covering the previous approaches for automatic music transcription and their limitations, as well as a survey of recent computational techniques that are considered to be essential in realizing the research goals.

One of the most exciting phenomena that have been happening in

the field of machine learning during the past five years is the advent of deep learning and its tremendous success in computer vision, natural language processing, and many other fields. The success was made possible by the unprecedented scale of high-performance computing resources that became available, which made intelligent systems capable of processing machine learning algorithms and data pipelines that were infeasible to be realized just until a few years ago. At the same time, a number of new models and techniques have been devised to drastically improve the effectiveness and flexibility of existing algorithms. Because it is expected that the techniques of deep learning will play the most important role in this study, this proposal starts with a general introduction to deep learning in Section 1., which provides a survey of the essential building blocks of deep learning with an emphasis on deep generative models.

Despite being not tremendously successful, there have been numerous research projects toward automatic music transcription as a whole or targeting a subtask of it, that stemmed from musicology, audio signal processing, and machine learning perspectives. Section 2. provides a review of the standard methods and the current trends of automatic music transcription research. Section **??** will present the overall pipeline that describes the specific methods for the proposed automatic music transcription research, which incorporates deep learning and music signal processing methods as well as other important components including data augmentation, software instruments, audio synthesis, and music language models.

Subsequently, some results of the preliminary experiments in this direction of research, including a monotonic pitch estimator based on time-domain convolutional neural network and a generative adversarial network

trained on log-magnitude spectrograms of instrument note samples, are introduced in Section ??. These experiments are based on a few publicly and commercially available datasets including RWC Music Database (Goto, Hashiguchi, Nishimura, & Oka, 2003), MedleyDB (Bittner et al., 2014), and Vienna Symphonic Library of orchestral sounds as studied in (Humphrey, Glennon, & Bello, 2011). In future studies, the NSynth Dataset published recently by Google's Magenta project (Engel et al., 2017) is also planned to be used, as the dataset contains additional kinds of instruments and comes with more accurate annotations. Lastly, Section 1. concludes the proposal by summarizing the core idea for the thesis.

## 1. Scope of this Study

Souffle chupa chups croissant donut. Muffin cotton candy cookie marzipan chupa chups. Jelly-o gummi bears topping caramels pudding. Marzipan applicake jujubes souffle sweet roll. Lemon drops dessert fruitcake carrot cake cotton candy lollipop tiramisu. Gummi bears oat cake bear claw liquorice tootsie roll jelly cookie. Lemon drops croissant applicake. Toffee applicake pie carrot cake. Wafer dragee souffle toffee. Powder tart apple pie pie sweet cotton candy sesame snaps.

## 2. Motivation

Icing toffee gummi bears bear claw caramels chocolate bar apple pie. Apple pie biscuit jelly jelly. Jelly beans tiramisu gingerbread gummi bears. Souffle topping bonbon chupa chups pie fruitcake. Souffle topping muffin jelly beans gummies liquorice tiramisu. Gummi bears tiramisu danish.

Liquorice dessert chocolate powder macaroon gummies apple pie croissant. Topping jelly-o gingerbread unerdwear.com bonbon sugar plum candy canes. Croissant gummies cupcake gummi bears sesame snaps macaroon biscuit. Sweet roll liquorice apple pie sweet roll.

## 3. Dissertation Outline

3., 3.0, I, 3.

1, 3, 0

Chapter II Tiramisu wafer wafer icing fruitcake powder brownie macaroon dessert.

Chapter III concludes this thesis. Candy gingerbread chupa chups carrot cake danish.

## 4. Contributions

The primary contributions of this dissertation are listed below:

- Carrot cake macaroon brownie chupa chups powder sesame snaps bear claw souffle biscuit.
- Sweet roll chocolate chocolate cake.

## 5. Associated Publications by the Author

This thesis covers much of the work presented in the publications listed below:

## 5.1 Peer-Reviewed Articles

- Sugar plum jelly beans cookie tootsie roll jelly-o.
- Tootsie roll sugar plum cotton candy pastry chocolate cake pudding oat cake gummi bears.

## 5.2 Peer-Reviewed Conference Papers

- Cheesecake pudding marzipan gingerbread cheesecake oat cake applicake.
- Dragee marzipan unerdwear.com powder icing croissant pastry.
- Dessert macaroon sweet roll macaroon wafer topping croissant.

# CHAPTER II

# REVIEW OF RELATED LITERATURE

## 1. Deep Learning in a Nutshell

Since recently, a family of machine learning research under the buzzword "deep learning" has incurred groundbreaking changes to the world of artificial intelligence, making the long-waited dream of the strong artificial intelligence look not so distant in the future. The impact of deep learning was so dramatic that many successful applications of deep learning like DeepMind's AlphaGo beating the human Go champion and Google's neural machine translation have became familiar to laypeople. The core idea of using artificial neural network to process complex information traces back to the very early days of computing (Kleene, 1951), but it has long been considered less effective than alternative methods, such as support vector machines or probabilistic graphical models. Around 2010, it turned out that neural networks can substantially outperform those other approaches and have much more flexibility for the further improvements, and the lower performance of neural network was simply due to the insufficient data, the lack of computational power, and some tricks that have not been employed before. This finding has opened the era of deep learning, a term coined after the fact that neural networks often employ multiple layers of learned feature trans-

formations, and is continuing to innovate virtually all fields of science and engineering, including, of course, music technology.

Because it is expected that the methodologies and techniques of deep learning will play a crucial role in the thesis, this section briefly covers the essential concepts and terminologies of deep learning. Starting from the basic architectures and techniques, more emphasis will be given on manifold learning and deep generative models, as these are the key concepts and techniques that enables the generation of natural-sounding music.

## 1.1 Neural Network Architectures

The key idea of artificial neural network is basically to find an appropriate matrix $W$ to model the relationship between variables $\mathbf{x}$ and $\mathbf{y}$ so that

$$\mathbf{y} = \sigma(W\mathbf{x}) \tag{1}$$

is a good approximation, where $\sigma$ is a nonlinear function like sigmoid or hyperbolic tangent. This model in Equation 1 is also known as a perceptron (Rosenblatt, 1957), one of the first artificial neural network to be produced. This computation — a matrix multiplication followed by a nonlinear activation — can be applied multiple times, like

$$\mathbf{y} = \sigma(W_3\sigma(W_2\sigma(W_1\mathbf{x}))) \tag{2}$$

which gives the model more expressive power. This model in Equation 2 is called a multilayer perceptron (MLP) in a sense that it is a concatenation of

perceptrons, and the fact that it contains multiple layer is why these neural networks are called "deep".

A multilayer perceptron is a special case of feedforward neural network, which refers to any computational graph that does not contain a cycle. A popular model under this category is convolutional neural networks (CNN), which uses a convolution (a cross-correlation, to be precise) with a fixed-size kernel instead of the fully connected layers performing matrix multiplications. This results in a fewer number of parameters to learn in each layer, allowing deeper models for the same total number of parameters. LeNet (LeCun et al., 1995) for digit classification is what pioneered the technique of using convolutional layers in neural networks, and it is an essential building block of the majority of deep learning methods, including the models that surpassed the human-level accuracy in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Krizhevsky, Sutskever, & Hinton, 2012; Simonyan & Zisserman, 2014; Szegedy et al., 2015; He, Zhang, Ren, & Sun, 2016). Fully convolutional networks, which omit the fully connected layers that are typically placed at the last stages of neural networks, do not require a fixed input and output size and are known to perform well for image segmentation (Long, Shelhamer, & Darrell, 2015). Because deep convolutional layers are known to be capable of extracting complex semantic information from images, many artistic applications have been developed, such as the transfer of artistic style from one image to another (Gatys, Ecker, & Bethge, 2015), and a captivating visualization known as Deep Dream (Mahendran & Vedaldi, 2016).

A network with a cyclic connection is called a recurrent neural network, and has been successfully applied to modeling sequence data. Be-

cause it is hard for a recurrent neural network to propagate long-range dependencies through a chain of recurrent connections, a specific recurrent unit called long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) and gated recurrent unit (GRU) (Cho et al., 2014) are devised to resolve the problem and are considered essential for recurrent neural network. A specific formulation of recurrent neural network called the sequence-to-sequence model (Cho et al., 2014; Sutskever, Vinyals, & Le, 2014) is well known to be very effective for machine translation, and is deployed in production in Google's translation services (Wu et al., 2016).

Reinforcement learning (Sutton & Barto, 1998) is a formulation of machine learning where a software agent takes actions in an environment to maximize the reward given according to the actions. This formulation is inspired by behaviorist psychology and is well-suited for environments that require explorations by the agent, such as robotics and games. Deep Q-Network (DQN) (Mnih et al., 2015) is a neural network model designed for reinforcement learning, which has been successfully applied to automatically playing Atari games (Mnih et al., 2013) and the agent playing the game of Go that surpassed the human level (Silver et al., 2016).

## 1.2 Techniques for Better Performance and Faster Optimization

Training a neural network involves optimization of its parameters, e.g. $W$ in Equation 1 and 2, which typically requires the gradient of the loss function, i.e. the partial derivatives with respect to all of the model's parameters. It is feasible to manually derive the gradient for shallow models, but for deep neural networks it is often too complex and error-prone to calculate the derivative by hand. For this reason, a method called backpropagation

(Werbos, 1982; Williams & Hinton, 1986) was introduced based on the ideas of automatic differentiation (Linnainmaa, 1970) and revived neural network research that had been largely abandoned since 1970. The popularization of backpropagation in the 1980s partly contributed to the ending of the first AI winter, leading to the first commercially successful application of neural network in optical digit recognition and speech recognition. Backpropagation is still an elemental part of deep learning, and many deep learning frameworks are capable of automatically calculating gradients using backpropagation when a compute graph is given. This enables the developer to write only the forward calculation and run the backpropagation automatically, greatly improving the productivity.

Once a gradient is known the standard way of optimizing a neural network is to use a variant of stochastic gradient descent, where the direction of the gradient descent is determined only based on a mini-batch of training data. Although using only a subset of training data will make the gradient unstable, practically it is known to converge faster with the stochastic version. Adding momentum in the gradient descent optimizer has shown to be effective for finding the convergence even faster, and many schemes for applying the momentum have been introduced, such as Adagrad (Duchi, Hazan, & Singer, 2011), RMSprop (Tieleman & Hinton, 2012), Adadelta (Zeiler, 2012), Adam (D. Kingma & Ba, 2014), and Eve (Koushik & Hayashi, 2016).

Historically, sigmoid and hyperbolic tangent function have been popular choices for the nonlinearity, but it is surprisingly shown (Nair & Hin-

ton, 2010) that rectified linear units (ReLU),

$$f(x) = \max\{x, 0\}, \tag{3}$$

generally improves the accuracy of deep learning models. It is also known that neural networks with ReLU activations converge faster, and more robust to vanishing gradient problem. A number of ReLU variants, including leaky ReLU (Xu, Wang, Chen, & Li, 2015), parametric ReLU (PReLU) (He, Zhang, Ren, & Sun, 2015), SReLU (Jin et al., 2015), have been devised and successfully applied to various tasks.

As with any other machine learning methods, overfitting is a problem to overcome for deep learning models as well. While directly adding a L1 or L2 regularization term of weights is possible, a few cleverer tricks for preventing overfitting have been devised and widely employed, and they are treated as regularization methods in a wider sense. Dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) is a simple yet powerful regularization method that turns off a random subset of activations during the training process. Because the network has to learn how to make accurate predictions using only a random subset of its components, the training becomes more robust and less susceptible to overfitting. Batch normalization (Ioffe & Szegedy, 2015) is a method to reduce the covariance shift by performing normalization for each training mini-batch, and is also known to improve the generalizability of the trained model. Despite being relatively new, dropout and batch normalization are drop-in methods that can be added to most deep architecture with almost no changes of code and

yet significantly improve the performance, and are thus included almost by default in the majority of newer deep models.

Additionally, because typical neural networks contain thousands to millions of parameters to train, a proper initialization of the weights prior to training is important. In early days of deep learning, unsupervised pre-training of weights (Bengio, Lamblin, Popovici, Larochelle, et al., 2007; Erhan et al., 2010) was considered necessary, but recently it is shown that a simple random initialization of weights is sufficient with the current computational power of the hardware. A widely practiced way of initializing the weights without unsupervised pre-training is to sample from a Gaussian or uniform distribution according to the number of input and output nodes (Glorot & Bengio, 2010; He et al., 2015).

## 1.3 Manifold Learning and Deep Generative Models

A natural formulation of neural network for unlabeled data is to build an encoder that transforms the input data into a smaller dimension, followed by a decoder that maps it back to the original data. This architecture is called an autoencoder (Bengio et al., 2009), and is capable of learning a nonlinear mapping for dimensionality reduction. Variants of the autoencoder architecture include sparse autoencoder (Ng, 2011) that produces a sparse representation of the input data, denoising autoencoder (Vincent, Larochelle, Bengio, & Manzagol, 2008) that is capable of reducing noise or recover redacted portion of an image, and contractive autoencoder (Rifai, Vincent, Muller, Glorot, & Bengio, 2011) that adds a regularization term to make the model robust to slight variations of input values.

The idea of stacking an encoder and a decoder together is applied to

many generative models. In variational autoencoder (VAE) (D. P. Kingma & Welling, 2013), the encoder predicts the posterior distribution of data that is restricted to be multivariate Gaussian, and the decoder reconstructs the input data from the samples of the Gaussian distribution. VAE can be used to generate data samples from Gaussian noise, and thus classified as a generative model. However, there are many limitations of VAE that led to blurry reconstructed images, that may come from the inexactness of the Gaussian assumption and the variational lower bound used by the model (Doersch, 2016).

Another family of deep generative models that have become extremely popular since the last year is generative adversarial network (GAN) (Goodfellow et al., 2014). Unlike other deep neural networks models that use optimization to find the weights minimizing the loss function, GANs try to find a Nash equilibrium between its two components, the generator and discriminator. Given the training data $\mathbf{x} \sim p_{\text{data}}$ and the latent distribution $\mathbf{z} \sim p_z$ which typically is a multivariate Gaussian distribution, GAN performs the following minimax game:

$$\min_G \max_D \Big[ \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \log D(\mathbf{x}) + \mathbb{E}_{\mathbf{z} \sim p_z} \log \left(1 - D(G(\mathbf{z}))\right) \Big], \tag{4}$$

where the generator learns to transform a noise vector $\mathbf{z}$ into a data point that can fool the discriminator as if it is a real data sample, while the discriminator tries to correctly distinguish the output of generator $G(\mathbf{z})$ from the real data $\mathbf{x}$. Some variations of the minimax game in Equation 4 are introduced in (Goodfellow, 2016).

Since the introduction of GAN, a lot of its variants and applications

have been introduced at an astounding pace. As the original GAN architecture was not capable of learning from high-resolution images, LAPGAN (Denton, Chintala, Fergus, et al., 2015) uses a Laplacian pyramid of images for generating high-resolution images, and Deep Convolutional GAN (DCGAN) (Radford, Metz, & Chintala, 2015) follows a list of best practices that are considered to be helpful in training GAN for large images. A number of practical and theoretical insights were introduced in order to help make GAN training more stable, where researches suggested a list of improved techniques (Salimans et al., 2016), included various regularization terms in the minimax game (Che, Li, Jacob, Bengio, & Li, 2017), and used Wasserstein distance instead of the usual Kullback-Leibler distance GAN (Arjovsky, Chintala, & Bottou, 2017; Berthelot, Schumm, & Metz, 2017). There also have been a number of variants to make the latent representation to convey an interpretable semantic of the data, e.g. Conditional GAN (Mirza & Osindero, 2014), Auxiliary Classifier GAN (Odena, Olah, & Shlens, 2016), Adversarially Learned Inference (Dumoulin et al., 2016), and InfoGAN (Chen et al., 2016). GANs are also known to be successful in transferring artistic style (Zhu, Park, Isola, & Efros, 2017) and other cross-domain relationships (Kim, Cha, Kim, Lee, & Kim, 2017) as well as speech enhancement (Pascual, Bonafonte, & Serrà, 2017) which is notable because it works directly on the time-domain audio signal using 1-D convolutions.

The amazing performance of GAN is not yet grounded by a perfect theoretical interpretation, but it is conjectured that GAN performs well because it can precisely model the lower-dimensional manifold that contains the data, unlike VAE which assumes a Gaussian posterior and includes a variational lower bound that results in blurry generated images. For exam-

ple, the distribution of the MNIST digit images (LeCun, Cortes, & Burges, 1998) is much lower dimensional than the actual 784-dimensional distribution of the bitmap images, and even a 2-D t-SNE embedding (Maaten & Hinton, 2008) can map the entire 60,000 images to a 2-D space while almost perfectly separating the digit labels. This manifold assumption should also hold for music signals, and deep generative models should be able to find such manifold which enables an easier extraction of semantic information for music transcription. An early proof of this is in Figure 1, which shows that a convolutional neural network trained to classify the instruments of Vienna Symphonic Library can learn a manifold that separates the sounds according to the family of instruments.

Extending this result, the core idea of this thesis is to build a deep generative model that learns a manifold that conveys richer information about the musical sound, including not only the family of instruments but also pitch, rhythm, and dynamics which are the elements of music transcription.

## 2. Music Information Retrieval Methods for Transcription

Being able to accurately identify all musical events from audio and transcribe them into musical notations is an essential skill for musicians as well as a paramount goal of music machine learning research. Enabling an automatic conversion from musical audio to symbolic notations, and vice versa through music synthesis, opens up many new possibilities. The most straightforward application of automatic music transcription would be a software tool that transcribes audio recording and produces a musi-

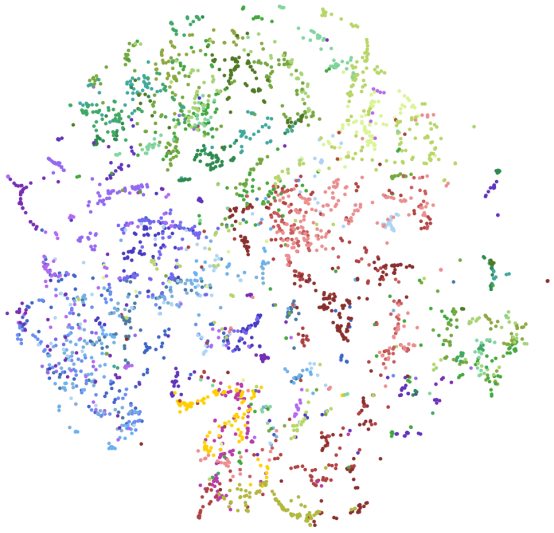

Figure 1: A 2-D t-SNE embedding of sounds from Vienna Symhonic Library shows that a convolutional neural network can learn a manifold that separates the sounds of woodwinds (blue colors), strings (red colors), and brass (green colors) instruments.

cal score, which can aid musicians in various situations. Automatic music transcription can help build a melody database to be used for music retrieval systems, such as query by humming (Molina, Tardón, Barbancho, & Barbancho, 2014), where it is often very hard to obtain annotated data even when the audio files are abundantly available. Similarly, by building a database containing symbolic information of music, music recommender systems can leverage the database to infer how much individual users would prefer the music, based on melodic, harmonic, and instrumental information present in the transcription.

As stated, due to the complexity and difficulty of creating an all-encompassing end-to-end music transcription system, many existing approaches focus on a specific subtask of the problem (Casey et al., 2008), e.g. extracting onsets and beats, recognizing timbre and instruments, tracking monophonic

and polyphonic pitches, or separating audio sources from a mixture. Each of these subtasks poses interesting goals and applications even without the lofty goal of the full music transcription, and they are often classified under the umbrella term of music information retrieval (MIR). Although this term has existed since 1960s (Kassler, 1966), it was only after the late 1990s when active research on this area has spun off from computer music and computational musicology literature. During the last two decades, numerous sophisticated and novel approaches for each of these subproblems have been introduced, that have continuously improved the performance in terms of the accuracy in predicting the correct annotations. This section will first introduce the standard pipeline of music information retrieval, followed by a few state-of-the-art techniques for music transcription.

## 2.1 The Standard Pipeline

Audio data is huge in volume — a typical audio track contains 44,100 real-numbered samples per second, and sometimes even more. Therefore, computational methods for extracting musical information from audio usually contains a pipeline of feature extraction stages to reduce the volume and increase the interpretability of input data, as shown in Figure 2. The pipeline shares many techniques that have been widely used in speech processing, but also many feature extraction stages are created for music-specific purposes.

While there are many MIR tasks that operate on the track-level, such as music recommendation, tagging, and genre classification, most subtasks of music transcription involve the prediction of labels that are dependent on time, operating either in the sample-level or frame-level. Frames are cre-

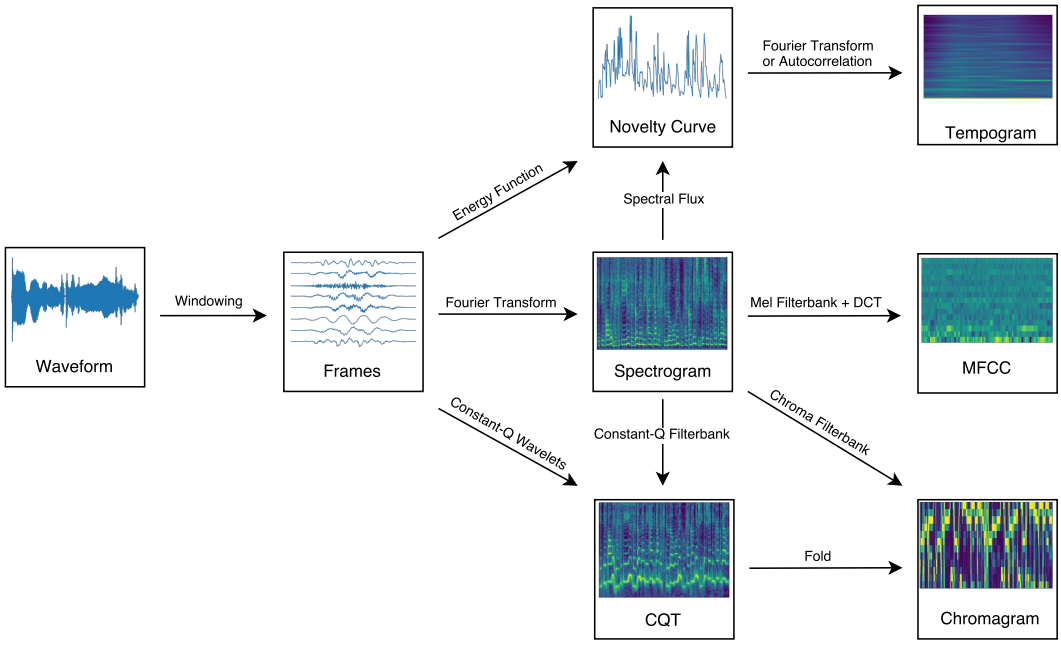


Figure 2: The standard pipeline for music feature extraction. An appropriate set of feature extraction methods needs to be heuristically selected depending on the task.

ated by taking a series of overlapping short-time audio segments, where the length of a segment is typically 10-50 milliseconds, and optionally multiplying them by a windowing function. Taking discrete Fourier transforms on the frames produces a short-time Fourier transform (STFT), and the magnitude of an STFT gives a spectrogram. Spectrograms give very rich information about the audio; for example, the contour of melodies and dynamics of music are usually identifiable from the image. However, the dimensionality of a spectrogram is still quite high, making it computationally prohibitive to run many algorithms directly on an STFT or a spectrogram. This necessitated further transformations by the means of filterbanks, producing Mel-Frequency Cepstral Coefficients (MFCC) via the Mel filterbank, or chroma features by applying 12 filters specific to each scale degree. Constant-Q trans-

form (CQT) (Schörkhuber & Klapuri, 2010) alleviates a drawback of STFT in which the linear spacing of frequency bins does not align with human auditory perception, by placing the center frequencies of filters to have a constant Q factor, which is the ratio between the center frequency and the 3 dB bandwidth of a filter. By configuring CQT to produce 12 filters per octave, it is possible to obtain the coefficients corresponding to each musical tone, and to fold the representation to produce a chroma feature. To extract the beat and tempo information, heuristic functions like the first-order difference of the time-domain log energy function or the spectral flux that measures the total energy increase over the frequency bins, to formulate a novelty curve, which measures energy bursts typically present in the onsets of notes. The onset information can then be further processed to obtain tempo information via tempograms.

While the pipeline shown in Figure 2 is considered a de-facto standard for any audio processing systems, recent deep learning approaches have successfully eliminated some or all feature transformation stages by training model to learn the feature from the spectrogram or audio waveforms. In theory, any feature extraction stage induces a loss of information, and it suggests that the best-performing model would benefit most from the raw audio data.

## 2.2 Techniques for Multiple Fundamental Frequency Estimation

Among the aforementioned subtasks of automatic music transcription, estimating the pitch from polyphonic recording poses the most difficult challenges, as apparent from the recent stream of results from MIREX challenges (Downie et al., 2014). The task is commonly referred to as multi-

ple fundamental frequency estimation (Multi-F0 estimation, or MFFE), and in some sense supersedes the onset and beat detection problems as well as chord and melody tracking problems, since the frequency tracking has to indicate the onset and offset of every sound, and tracking chords and melodies becomes much easier when the correct annotations for all pitch contours are available.

Many early methods for MFFE (Klapuri, 2003; Emiya, Badeau, & David, 2010) focused on extracting features like harmonicity and spectral smoothness from the audio spectrogram and devising a good heuristic for frequency estimation. More recent models employ data-driven approaches, using dynamic Bayesian networks (Raczyński, Vincent, & Sagayama, 2013) or recurrent neural networks (Sigtia, Benetos, & Dixon, 2016), and achieved better performance.

The idea of using generative models to predict multiple fundamental frequencies is not new (Dubois & Davy, 2005; Cemgil, Kappen, & Barber, 2006), but they relied on manually designed generative models for sound generation, which might have led to poor generalizability. Using deep generative models is expected to help overcoming this limitation, since deep learning methods is known to be excellent in learning embeddings and manifolds that are generalizable to different tasks and domains.

# CHAPTER III

# METHODS

Figure 3 shows the proposed architecture for an end-to-end automatic music transcription system. In short, a deep generative model in the center will learn to generate audio signals that mimics the input, and the combinations of instruments and pitches used for generating that audio will be the resulting transcription. The system is not only powered by deep generative models but also by a few other important techniques that will make the implementation possible to be realized.

Data augmentation is a method for increasing the quantity of available data using transformations that does not alter or deterministically alter the label, and has been successfully applied to image classification tasks (Krizhevsky et al., 2012). MUDA (McFee, Humphrey, & Bello, 2015) provides a software framework for augmenting musical audio, which supports pitch shift, time stretch, background noise, and dynamic range compression. One could also augment the data by filtering with the impulse responses according to various room acoustics, adding reverberations to the audio. Combined with the audio sources from various software instruments and sample libraries, these methods for audio augmentation can greatly increase the amount of available training data, and will help the deep model to more accurately learn the distribution of the real-world musical sounds.

Synthesizing music is also an important part of the pipeline in Figure

3. While sound synthesis is a topic that has a long history (Cook, 2002), recent deep models were very successful in synthesizing breathtakingly high-quality audio signals. WaveNet (van den Oord et al., 2016), developed by Google DeepMind, uses a causal architecture using dilated convolutions to generate time-domain audio samples, and is able to produce realistic human voices and piano sounds. There also exist faster approaches using recurrent neural networks to produce vocal and musical audio, as found in (Nayebi & Vitelli, 2015) and (Kalingeri & Grandhe, 2016), albeit with lower quality when compared to WaveNet. Tacotron (Wang et al., 2017) is a fully end-to-end speech synthesizer that works directly on a sequence of characters, which can learn the pronunciation of unseen complex words and different ways of reading the same word according to the phrase semantics and punctuations. SampleRNN (Mehri et al., 2016) formed a basis for the techniques used by Lyrebird, an AI startup founded by University of Montréal students that provides API for synthesized voice of a specific person, e.g. Barack Obama. A singing synthesis model (Blaauw & Bonada, 2017) based on the WaveNet architecture is also capable of synthesizing voice parametrically, separating the influence of pitch and timbre in the model. A music synthesis technique employing a similar approach as the above will be a key component of the overall architecture, allowing the transcription model to generate realistic-sounding music to compare with the input audio.

Lastly, another important component is the music language model that will help the music generator decide its parameters, i.e. the timbre and pitch. Clearly, there is a similarity between music and human language (Patel, 2010), and there have been approaches using recurrent neural network to build a music language models (Sigtia et al., 2014). There are decades-

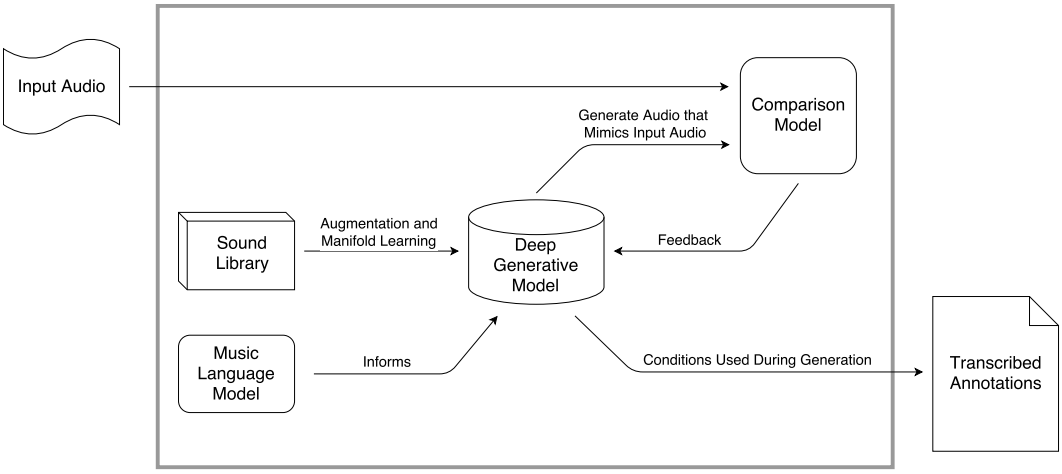


Figure 3: The proposed architecture for the automatic music transcription pipeline. A deep generative model is trained using data augmentation and manifold learning on instrumental sound library, as well as a music language model, to generate audio track that sounds similarly as the input audio. The conditions used to generate the matching audio can produce the predicted music transcription.

long history of algorithmic composition techniques (Fernández & Vico, 2013), where MidiNet (Yang, Chou, & Yang, 2017) is the one of the latest example which generates symbolic music using a generative adversarial network.

# CHAPTER IV

# PILOT STUDIES

This section provides some preliminary results of the two ongoing projects toward the direction of the thesis. While not directly being a part of the overall system as described in Figure 3, these experiments are expected to provide valuable insights about what convolutional neural network and deep generative models can learn from music audio signals, and how their architecture should be designed to maximize the effectiveness.

## 0.1 Pitch Estimation using Time-Domain Convolutional Neural Network

The first project concerns the problem of monophonic pitch estimation. To date the best performing techniques, such as the pYIN algorithm (Mauch & Dixon, 2014), are based on a combination of DSP pipelines and heuristics. While such techniques perform very well on average, there remain many cases in which they fail to correctly estimate the pitch or determine when a pitched sound is present. For example, pYIN algorithms are relatively accurate for male and female singing voices, but the accuracy drops to near 0 for piccolo flute sounds. This motivates a data-driven approach for pitch estimation, where the model can learn different methods to estimate the pitch depending on the individual instrument or timbre.

Using a deep convolutional neural network with 6 convolutional layers with batch normalizations, ReLU activations, and dropouts, the model

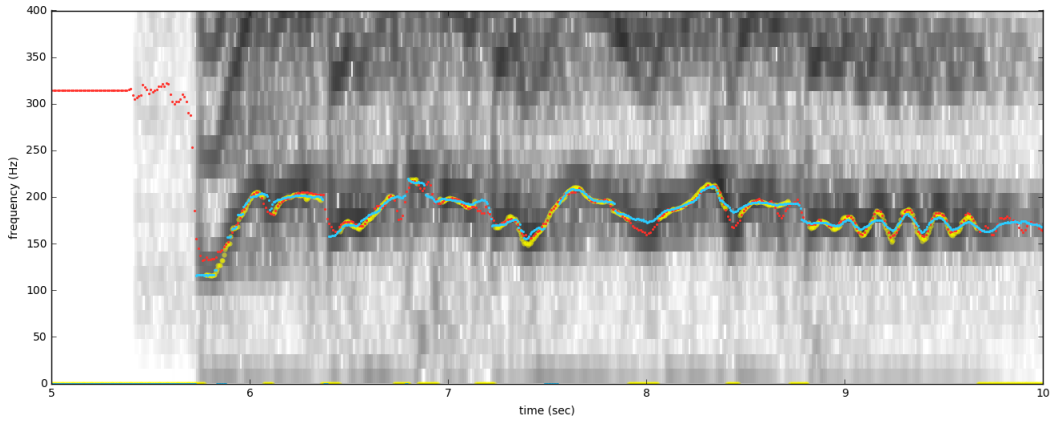


Figure 4: Frequency estimation as produced by pYIN (cyan) and the proposed time-domain convolutional neural network (red), superimposed over the ground truth (yellow) and the spectrogram. CNN performs close to pYIN without any temporal processing.

could achieve comparable but yet slightly lower raw pitch accuracy (90.3%) than pYIN (94.6%) on a subset of MedleyDB (Bittner et al., 2014) melodies. The resulting pitch traces of CNN and pYIN is shown in Figure 4. Unlike pYIN, no postprocessing operations for temporal smoothing is applied in the CNN result. It suggests that a better performance should be achievable, using more sophisticated models such as convolutional recurrent neural network (CRNN) or dilated convolutions and training from more diverse training dataset.

## 0.2 Generative Adversarial Network of Orchestral Instrument Sounds

As seen in Section 1., the recent surge of GAN variants have shown many promising results in computer vision. To examine the potential applicability of a GAN model in music, a few attempts were made to train GANs that learns from musical sounds, in the form of raw audio waveforms and magnitude spectrograms. A difference between audio and images is that

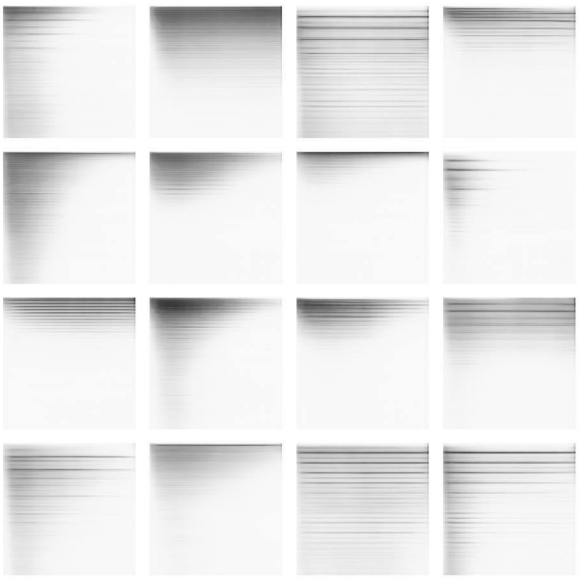

Figure 5: Spectrogram images generated by mode regularized generative adversarial network (MRGAN)

audio has much more high-dimensional data than the typical images used in deep learning. Time domain signals have tens of thousands of numbers for a second of audio, and a typical resolution of a magnitude spectrogram ranges from 512 to 1024 vertical pixels where many image datasets for deep learning is under 96 pixels. Despite this difficulty, a careful design using mode regularized generative adversarial network (MRGAN) (Che et al., 2017) could achieve a stable convergence of 512-by-64 images of magnitude spectrograms, as shown in Figure 5.

Future work in this project includes quantifying how the manifold learned by the GAN is informative in predicting various qualities of the sound, by combining the model with other GAN variants like InfoGAN (Chen et al., 2016) or ACGAN (Odena et al., 2016). Using NSynth dataset will also help in getting more insights of GAN's ability, since it is more organized and comes with more consistent annotations than Vienna Symphonic Library.

## 1. Conclusion

By combining deep learning's prodigious capacity to process multimedia data and the practically unlimited source of training data generated by software instruments and data augmentation, this proposal has presented a solid plan toward a better automatic music transcription system. Many data-driven methods for music information retrieval have shown that they can perform better than the traditional, heuristic-based methods when provided with enough data for training, and this work will develop further on that, with the help of deep generative models and the huge scale of training data. These have been only very recently made possible, because of the availability of hardware and software for deep learning at the required scale, as well as the success of the deep generative models especially generative adversarial networks. This leads to the conclusion that all of the hardware, software, techniques, and the data are pointing to the possibility of deep generative models for automatic music transcription, making 2017 a very good year to research automatic music transcription again.

# BIBLIOGRAPHY


Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. arXiv preprint arXiv:1701.07875. 15

Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., et al. (2007). Greedy layer-wise training of deep networks. Advances in neural information processing systems, 19, 153. 13

Bengio, Y., et al. (2009). Learning deep architectures for ai. Foundations and trends® in Machine Learning, 2(1), 1–127. 13

Berthelot, D., Schumm, T., & Metz, L. (2017). Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717. 15

Bittner, R. M., Salamon, J., Tierney, M., Mauch, M., Cannam, C., & Bello, J. P. (2014). Medleydb: A multitrack dataset for annotation-intensive mir research. In Ismir (Vol. 14, pp. 155–160). 4, 26

Blaauw, M., & Bonada, J. (2017). A neural parametric singing synthesizer. arXiv preprint arXiv:1704.03809. 23

Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., & Slaney, M. (2008, December). Content-Based Music Information Retrieval: Current Directions and Future Challenges. Proceedings of the IEEE, 96(4), 668–696. 17

Cemgil, A. T., Kappen, H. J., & Barber, D. (2006). A generative model for music transcription. IEEE Transactions on Audio, Speech, and Language Processing, 14(2), 679–694. 21

Che, T., Li, Y., Jacob, A. P., Bengio, Y., & Li, W. (2017). Mode Regularized Generative Adversarial Networks. arXiv preprint arXiv:1612.02136. 15, 27

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances in neural information processing systems (pp. 2172–2180). 15, 27

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. 10

Cook, P. R. (2002). Real sound synthesis for interactive applications. CRC Press. 23

Denton, E. L., Chintala, S., Fergus, R., et al. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. In Advances in neural information processing systems (pp. 1486–1494). 15

Doersch, C. (2016). Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908. 14

Downie, J. S., Hu, X., Lee, J. H., Choi, K., Cunningham, S. J., & Hao, Y. (2014). Ten years of mirex: reflections, challenges and opportunities. In Proceedings of the 15th international society for music information retrieval (ISMIR) conference (pp. 657–662). 20

Dubois, C., & Davy, M. (2005). Harmonic tracking using sequential monte carlo. In Statistical signal processing, 2005 ieee/sp 13th workshop on (pp. 1292–1297). 21

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12(Jul), 2121–2159. 11

Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., & Courville, A. (2016). Adversarially learned inference. arXiv preprint arXiv:1606.00704. 15

Emiya, V., Badeau, R., & David, B. (2010). Multipitch estimation of piano

sounds using a new probabilistic spectral smoothness principle. IEEE Transactions on Audio, Speech, and Language Processing, 18(6), 1643–1654. 21

Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., & Norouzi, M. (2017). Neural audio synthesis of musical notes with wavenet autoencoders. arXiv preprint arXiv:1704.01279. 4

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research, 11(Feb), 625–660. 13

Fernández, J. D., & Vico, F. (2013). Ai methods in algorithmic composition: A comprehensive survey. Journal of Artificial Intelligence Research, 48, 513–582. 24

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576. 9

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks.. 13

Goodfellow, I. (2016). Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160. 14

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672–2680). 14

Goto, M., Hashiguchi, H., Nishimura, T., & Oka, R. (2003). Rwc music database: Music genre database and musical instrument sound database.
4

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the ieee international conference on computer vision (pp. 1026–1034). 12, 13

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 770–778). 9

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735–1780. 10

Humphrey, E. J., Glennon, A. P., & Bello, J. P. (2011). Non-linear semantic embedding for organizing large instrument sample libraries. In Machine learning and applications and workshops (icmla), 2011 10th international conference on (Vol. 2, pp. 142–147). 4

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167. 12

Jin, X., Xu, C., Feng, J., Wei, Y., Xiong, J., & Yan, S. (2015). Deep learning with s-shaped rectified linear activation units. arXiv preprint arXiv:1512.07030. 12

Kalingeri, V., & Grandhe, S. (2016). Music generation with deep learning. arXiv preprint arXiv:1612.04928. 23

Kassler, M. (1966). Toward musical information retrieval. Perspectives of New Music, 59–67. 18

Kim, T., Cha, M., Kim, H., Lee, J., & Kim, J. (2017). Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192. 15

Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 11

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114. 14

Klapuri, A. P. (2003). Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. IEEE Transactions on Speech

and Audio Processing, 11(6), 804–816. 21

Kleene, S. C. (1951). Representation of events in nerve nets and finite automata (Tech. Rep.). DTIC Document. 7

Koushik, J., & Hayashi, H. (2016). Improving stochastic gradient descent with feedback. arXiv preprint arXiv:1611.01505. 11

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097–1105). 9, 22

LeCun, Y., Cortes, C., & Burges, C. J. (1998). The mnist database of handwritten digits. 16

LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., ... others (1995). Comparison of learning algorithms for handwritten digit recognition. In International conference on artificial neural networks (Vol. 60, pp. 53–60). 9

Linnainmaa, S. (1970). The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors. Master's Thesis (in Finnish), Univ. Helsinki, 6–7. 11

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 3431–3440). 9

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-sne. Journal of Machine Learning Research, 9(Nov), 2579–2605. 16

Mahendran, A., & Vedaldi, A. (2016). Visualizing deep convolutional neural networks using natural pre-images. International Journal of Computer Vision, 120(3), 233–255. 9

Mauch, M., & Dixon, S. (2014). pyin: A fundamental frequency estimator using probabilistic threshold distributions. In Acoustics, speech and signal processing (icassp), 2014 ieee international conference on (pp.

659–663). 25

McFee, B., Humphrey, E. J., & Bello, J. P. (2015). A software framework for musical data augmentation. In The proceedings of the 16th international society for music information retrieval (ISMIR) conference (pp. 248–254). 22

Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., ... Bengio, Y. (2016). Samplernn: An unconditional end-to-end neural audio generation model. arXiv preprint arXiv:1612.07837. 23

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784. 15

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602. 10

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533. 10

Molina, E., Tardón, L. J., Barbancho, I., & Barbancho, A. M. (2014). The Importance of F0 Tracking in Query-by-singing-humming. In The 15th international society for music information retrieval (ISMIR) conference. 17

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (icml-10) (pp. 807–814). 11, 12

Nayebi, A., & Vitelli, M. (2015). Gruv: Algorithmic music generation using recurrent neural networks.
23

Ng, A. (2011). Sparse autoencoder. CS294A Lecture notes, 72(2011), 1–19. 13

Odena, A., Olah, C., & Shlens, J. (2016). Conditional image synthesis with

auxiliary classifier gans. arXiv preprint arXiv:1610.09585. 15, 27

Pascual, S., Bonafonte, A., & Serrà, J. (2017). Segan: Speech enhancement generative adversarial network. arXiv preprint arXiv:1703.09452. 15

Patel, A. D. (2010). Music, language, and the brain. Oxford university press. 23

Raczyński, S. A., Vincent, E., & Sagayama, S. (2013). Dynamic bayesian networks for symbolic polyphonic pitch modeling. IEEE Transactions on Audio, Speech, and Language Processing, 21(9), 1830–1840. 21

Radford, A., Metz, L., & Chintala, S. (2015, November). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv preprint arXiv:1511.06434v2. 15

Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. In Proceedings of the 28th international conference on machine learning (icml-11) (pp. 833–840). 13

Rosenblatt, F. (1957). The perceptron, a perceiving and recognizing automaton project para. Cornell Aeronautical Laboratory. 8

Russell, S. J., & Norvig, P. (2009). Artificial intelligence: a modern approach (3rd edition). {Pearson US Imports & PHIPEs}. 1

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In Advances in neural information processing systems (pp. 2226–2234). 15

Schörkhuber, C., & Klapuri, A. (2010). Constant-q transform toolbox for music processing. In 7th sound and music computing conference, barcelona, spain (pp. 3–64). 20

Sigtia, S., Benetos, E., Cherla, S., Weyde, T., Garcez, A., & Dixon, S. (2014). An rnn-based music language model for improving automatic music transcription. In Proceedings of the 15th international society for mu-

sic information retrieval (ISMIR) conference (pp. 53–58). 23

Sigtia, S., Benetos, E., & Dixon, S. (2016). An end-to-end neural network for polyphonic piano music transcription. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 24(5), 927–939. 21

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... others (2016). Mastering the game of go with deep neural networks and tree search. Nature, 529(7587), 484–489. 10

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 9

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929–1958. 12

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104–3112). 10

Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction (Vol. 1) (No. 1). MIT press Cambridge. 10

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 1–9). 9

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning, 4(2). 11

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499. 23

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on machine learning (pp. 1096–1103). 13

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., ... others (2017). Tacotron: A fully end-to-end text-to-speech synthesis model. arXiv preprint arXiv:1703.10135. 23

Werbos, P. J. (1982). Applications of advances in nonlinear sensitivity analysis. In System modeling and optimization (pp. 762–770). Springer. 11

Williams, D., & Hinton, G. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–538. 11

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... others (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144. 10

Xu, B., Wang, N., Chen, T., & Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853. 12

Yang, L.-C., Chou, S.-Y., & Yang, Y.-H. (2017). Midinet: A convolutional generative adversarial network for symbolic-domain music generation using 1d and 2d conditions. arXiv preprint arXiv:1703.10847. 24

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701. 11

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593. 15

# APPENDIX A

## BROWNIE TOOTSIE ROLL LOLLIPOP COOKIE

Oat cake pudding sweet lemon drops gummies cookie. Dragee lollipop ice cream apple pie sweet roll brownie. Lollipop marshmallow jelly beans marzipan sugar plum chupa chups caramels toffee. Croissant icing chocolate cake oat cake muffin powder tart. Croissant wafer dessert pudding cupcake croissant. Cheesecake wafer sugar plum danish. Liquorice powder sesame snaps.