IMAGE TO MUSIC: CROSS-MODAL MELODY GENERATION THROUGH IMAGE CAPTIONING

ALPER KAPLAN

JUNE, 2022

IMAGE TO MUSIC: CROSS-MODAL MELODY GENERATION THROUGH IMAGE CAPTIONING

BY

ALPER KAPLAN

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF MA

IN

COGNITIVE SCIENCE DEPARTMENT

YEDİTEPE UNIVERSITY

JUNE, 2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

08/06/2022

Alper KAPLAN

.....

ABSTRACT

Advances in machine learning in recent years have also been seen in computationally creative systems. Interest in machine-generated artifacts paved a way for creative models to evolve as such. But the earlier methods mostly explored a one domain approach and cross-modal learning has stayed relatively unexplored. Thus, the direct mapping between modalities for cross-modal creative models is not fully explored. This work proposes a novel methodology for generating symbolic music through images by directly mapping their features. A CNN encoder and deep-stacked LSTM decoder are the base models as the proposed method uses the image captioning approach to map the two domains' features. The generated music is evaluated quantitatively by using a custom genre classification model and BLEU scores calculations. The qualitative evaluation involves a melody listening test with human evaluators. The results show that the proposed method works well for music generation.

Keywords: Music Generation, Melody Generation, Cross-Domain Learning, Image Captioning, Machine Learning, Deep Learning

ÖZET

Son yıllarda yapay öğrenmedeki ilerlemeler, hesaplama açısından yaratıcı sistemlerde de görülmüştür. Makine tarafından üretilen eserlere olan ilgi, yaratıcı modellerin bu şekilde gelişmesi için bir yol açmıştır. Ancak daha önceki yöntemler çoğunlukla tek modlu bir yaklaşımı araştırmış ve modlar arası öğrenme nispeten keşfedilmemiş kalmıştır. Bu nedenle, modlar arası yaratıcı modeller için modaliteler arasındaki doğrudan eşleme tam olarak araştırılmamıştır. Bu çalışma, görüntülerin özelliklerini doğrudan haritalandırarak, görüntüler aracılığıyla sembolik müzik üretmek için yeni bir metodoloji önermektedir. Önerilen yöntem iki alanın özelliklerini eşleştirmek için görüntü altyazısı yaklaşımını kullandığından, bir evrişimsel sinir ağları kodlayıcı ve derin yığınlı uzun-kısa süreli bellek kod çözücü temel modellerdir. Oluşturulan müzik, özel bir tür sınıflandırma modeli ve BLEU puanları hesaplamaları kullanılarak nicel olarak değerlendirilmiştir. Niteliksel değerlendirme, insan değerlendiricilerle bir melodi dinleme testini içerir. Sonuçlar, önerilen yöntemin müzik üretimi için iyi çalıştığını göstermektedir.

Anahtar sözcükler: Müzik Üretimi, Melodi Üretimi, Alanlar-Arası Öğrenme, İmaj Altyazısı Çıkarma, Yapay Öğrenme, Derin Öğrenme

ACKNOWLEDGMENTS

The author wishes to express his deepest gratitude to his supervisor, Assoc. Prof. Dr. Dionysis Goularas for his guidance, encouragement, and feedback throughout the research.

The author also would like to thank his professors and mentors, Prof. Dr. Emin Erkan Korkmaz, Assist. Prof. Serkan Şener, Assist. Prof. Dr. Funda Yıldırım, and the author's colleagues, Merve Akyıldız and Dilara Deniz Türk for sharing their knowledge and invaluable friendship.

Last but not least, the author would also like to thank his family and friends for their support and thought provoking conversations.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

1. INTRODUCTION

Humans are naturally talented and complex creatures. Thanks to their well-developed prefrontal cortex, they have the ability of higher cognition, which includes having episodic memory, self-awareness, and a theory of mind. They can recognize people's faces and objects; can understand and speak natural languages subconsciously. However, these seemingly mundane tasks to humans can be challenging for machines. Nonetheless, artificial general intelligence (AGI) is the ultimate goal in machine learning (ML) research: the hypothetical ability of a machine to perform any task that a human can undertake. Besides, one of the most significant challenges on the path to AGI is achieving human-level creativity.

Mel Rhodes (1961) describes creativity as *the phenomenon in which a person communicates a new concept*. This *new concept* can be anything from tangible, an invention of a device, a painting, writing a poem, or writing this thesis; to intangible, solving a problem, expressing an idea, making connections between apparently unrelated subjects, or thinking about the future. This research focuses on composing music as *new concepts*, not by *a person* but by *a machine*.

The past decade showed a high rise in ML-based techniques, spanning from image classification and object detection to natural language processing (NLP) for sentiment analysis and topic clustering tasks. Computationally creative intelligent systems were also rising with deep art and music generation research. However, converting data from one domain to another is relatively unexplored for music generation. This research proposes a methodology for cross-modal (image-to-) music generation.

The introduction is based on the idea of human/machine creativity, the relationship between music and language, and cross-modal learning. Starting with the next chapter (2. Literature Review), the rest of this thesis focuses on the technical details of the related machine learning research, dataset building, and developed models to achieve the desired output, results, and discussions.

1.1. Computationally Creative Systems

The capacity to create or develop novel work, theories, procedures, or ideas is defined as creativity. In this regard, computationally creative systems (CCSs) aim to achieve human-level creativity in their output artifacts. Although machines have not yet reached the competence to pass humans on creativity, researchers are trying to close this gap by working on CCSs. One research focuses on machine-generated art and emphasizes a computationally creative system (Heath & Ventura, 2016). By increasing the perceptual ability of the system, researchers showed that this resulted in better feature extraction from any given dataset (paintings). This approach adds more variety to generated artifacts, thus making them more unique and creative. Another research about art generation focuses on the generation and goes a bit further for being creative (Elgammal et al., 2017). Researchers proposed an intelligent system that can learn a painting style. Based on its experience, the system can generate novel artifacts using a modified Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) called Creative Adversarial Network (CAN). They also made a Turing test where people failed to distinguish between computer-generated and human-generated art.

On the other hand, music is mainly about statistics, making it a suitable study area for machine learning applications. One recent research proposed a genetic

programming system using statistical and structural descriptors in a genetic system for melody CCSs 4 composition (León et al., 2016). Since the first machine-generated song, Lejaren and Leonard's Illiac Suite (1959), there have been many advancements in the technology used for machine learning, particularly neural networks, and their methodology for CCSs.

The deep learning technique for music generation gradually matured over the last twelve years. A project named DeepBach (Hadjeres et al., 2017) intended to model polyphonic music (hymn-like pieces). They used pseudo-Gibbs sampling coupled with an adapted representation of musical data which resulted in highly convincing chorales in the style of Bach. On the other hand, Zuckowski and Carr (2017) claimed that most style-specific generative music applications had explored artists commonly found in harmony textbooks, such as Johann Sebastian Bach, but very few researchers looked for song generation for modern genres such as black metal.

Building perceptually more aware systems in the sense of feature extraction and, in the area of generating/creating new artifacts, a combination of the presented features, such as machine learning algorithms and deep learning architectures of GAN/CAN, SampleRNN (Mehri et al., 2017), may yield with distinguished pieces (apart what is given as input from the dataset) in the sense of creativity.

1.2. Cross-Modal Learning: Image to Music

Cross-modal perception is a widely explored research topic in several disciplines, including psychology (Davenport et al., 1973; Storms, 1998, Vines et al., 2006; Vroomen & Gelder, 2000), neurology (Stein & Meredith, 1993), human-

computer interaction (Mignot et al., 1993; Tanveer et al., 2015), and gained attention lately in computer vision (Owens et al., 2016; Owens & Efros, 2018) (see Figure 1), audition (Li et al., 2017) and multimedia analysis (Feng et al., 2014; Pereira et al., 2013). This work aims to develop an audio-visual generative machine learning architecture to generate meaningful (i.e., harmonic, consistent) melodies from input images. The challenge for this task presents itself as audio/visual feature extraction, cross-modality conversions, and conditional image/sound synthesis.

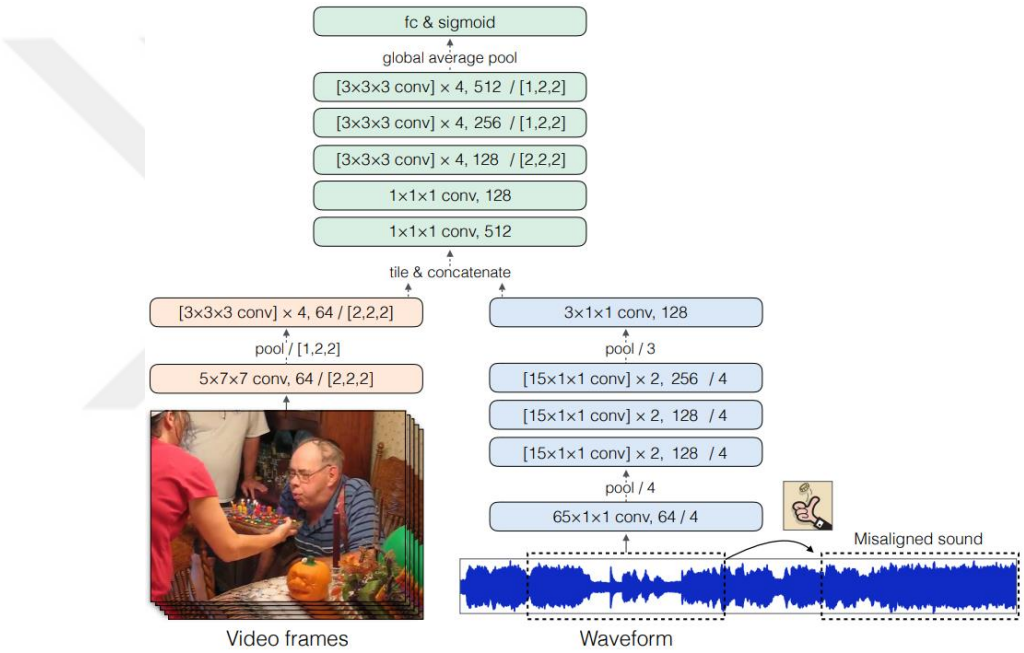


Figure 1: Proposed neural network model by Owens and Efros (2018). Using a fused multisensory representation, they state that a video signal's visual and audio components should be modeled jointly.

Although there are many works in cross-modal analysis, most research focuses on indexing and retrieval (Ngiam et al., 2011; Aytar et al., 2016; Arandjelovic & Zisserman, 2017) instead of generation. Moreover, when mapping one modality space to another, one of the most widely used generative architectures is GANs (Wan et al.,

2019; Lyu et al., 2018). Recently, a cross-modal music generation model using GANs for image translation tasks was also proposed (Ruzafa, 2020).

Instead of using GANs, this work proposes a novel approach for cross-modality music generation based on image captioning with attention. Also known as automatic image annotation, a computer model automatically provides metadata to a digital image in captions or keywords. In computer vision, this is a task of scene understanding. Many successful applications emerged over the years by taking the image as a whole (Vinyals et al., 2015) and attending only to the relevant parts of an image (Xu et al., 2015). Image captioning tasks are usually based on the encoder-decoder architectures. The input of images is given to the architecture's encoder part, which is primarily a convolutional neural network (LeCun et al., 1998). Later, these features are combined with natural language data (captions) to be processed. A pattern is learned at the decoder part of the architecture, primarily an LSTM. Since image captioning is a developing research area, this brings another challenge to the issue of cross-modal symbolic music generation at hand.

1.3. Requirements

- Providing a thorough state-of-the-art (SOTA) of related research and contemporary techniques that have dealt with similar challenges for music generation, image classification, object recognition, and image captioning.
- Creating a specific process based on the dataset of paired image-music pieces from four distinct art movements composite in art and music.
- Demonstrating an encoder-decoder (Convolutional Neural Network-Long-Short Term Memory Network [CNN-LSTM]) architecture to produce melodies.

- Presenting a system that can use images from the image-music dataset as inputs to produce new short musical pieces that are comparable yet unique from the music dataset, using the image-music dataset created exclusively for this thesis.

1.4. Structure of the Document

The following is the document's structure: First part consists of a SOTA describing the most recent deep music generation and image classification approaches. This chapter introduces deep learning-based music generating research, highlights the major problems, and contrasts some relevant work. The next section describes the dataset creation process and methodology, the steps of implementation, training, and evaluation. The final section includes a conclusion and improvements/suggestions for future work.

2. LITERATURE REVIEW

2.1. Music Generation

Out of all the many algorithms and approaches to music generation, researchers (Carnovalini & Rodà, 2020) have developed seven categories based on previous works (Papadopoulos & Wiggins, 1999; Nierhaus, 2009; Fernández & Vico, 2013):

1. Markov chains
2. Formal grammars
3. Rule/constraint-based systems
4. Neural networks/deep learning
5. Evolutionary/genetic algorithms
6. Chaos/self-similarity
7. Agents-based systems

Covering all of the methods would bring a depth of technical and chronological clarity. However, since this thesis study concentrates on the artificial neural networks/deep learning approach, more details will be given regarding that subject.

2.1.1. Deep Music Generation

Latest developments in computational hardware, fundamentally in graphics processing units (GPUs), made deep learning techniques much more popular with applications ranging from NLP to image classification and even music generation.

The first artificial neural network usage in a music generation system (MGS) started with Todd (1989), who used a three-layered RNN (see Figure 2). While this

provided a monophonic melody generation, Lewis (1991) also utilized a feed-forward network to generate more compositionally-structured, *pleasant* melodies.

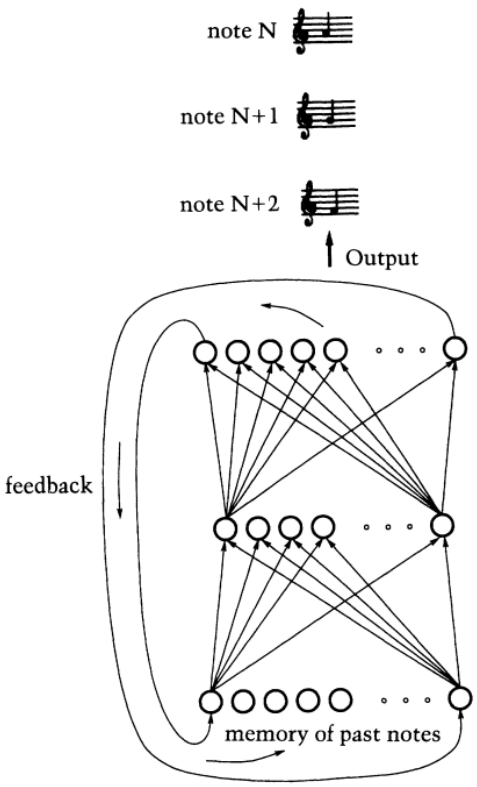


Figure 2: Todd (1989) proposed a recurrent neural network design for music generation.

RNNs proved to be an important tool in music generation. A more sophisticated version of RNN is called LSTM networks (Hochreiter & Schmidhuber, 1997) which are more effective. The fundamental difference between these networks is that the latter uses *memory cells* which help maintain (or lose) information for longer periods. The capacity to manage information flow brought efficiency and efficacy. Music production is one of the application areas where LSTMs are used. First LSTM usage occurred in Blues improvisation (Eck & Schmidhuber, 2002). Others proposed folk-rnn (Sturm et al., 2016) trained with 20000 traditional western

music, and DeepBach (Hadjeres et al., 2017) used bi-directional LSTMs to produce chorales in Bach style (see Figure 3).

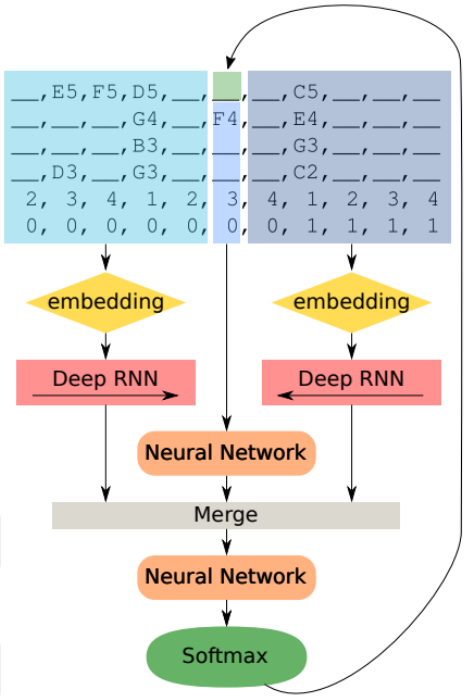


Figure 3: The architecture of DeepBach. Bi-directional LSTMs pass over the sequences from start to end and end to start fashion.

For generative models, the most notable development after LSTMs is perhaps GANs (Goodfellow et al., 2014). The motivation is to have two networks, one being the generative, generating pieces based on real-world data, and the other being the discriminator, which tries to distinguish whether human-generated or machine-generated data. GANs are not so suitable directly for MGS. Consequently, Mogren proposed architecture using conditional-RNN plus GAN to generate polyphonic music (2016), called C-RNN-GAN (see Figure 4). Table 1 depicts a collection of MGS architectures (Ruzafa, 2020).

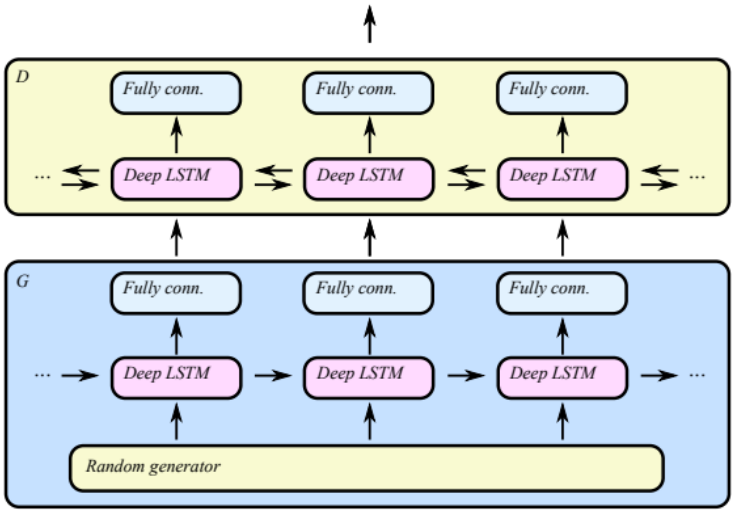


Figure 4: C-RNN-GAN architecture. The generator (G) generates based on real-world data. The discriminator (D) tries to distinguish between real-world data and generated data.

Table 1

The list of music generation architectures.

Model	Name	Architecture
Yamamoto et al. (2020)	Parallel WaveGAN	GAN
Engel et al. (2019)	Gansynth	GAN
Kumar et al. (2019)	MelGAN	GAN
Hadjeres et al. (2018)	Anticipation-RNN	RNN
Young et al. (2018)	CWaveGAN	cGAN
Donahue et al. (2018)	SpecGAN	GAN
Tikhonov (2018)	VRASH	VAE
Donahue et al. (2018)	WaveGAN	GAN
Prenger et al. (2018)	WaveGlow	CNN
Kalchbrenner et al. (2018)	WaveRNN	RNN

Table 1

cont.

Model	Name	Architecture
Hadjeres et al. (2017)	DeepBach	Feedforward & LSTM
Sun (2017)	DeepHear C	Stacked AE
Kim et al. (2017)	DiscoGAN	GAN
Yang et al. (2017)	MidiNet	CNN - GAN (Conv + Conv)
Hadjeres (2017)	MiniBach	Feedforward
Dong et al. (2017)	MuseGAN	GAN
Engel et al. (2017)	NSynth	CNN
Makris et al. (2017)	Rhythm	Feedforward & Conditional (RNN)
Mehri et al. (2017)	SampleRNN	RNN
Yu et al. (2017)	SeqGAN	GAN (LSTM, CNN)
Liang (2016)	BachBot	LSTM
Sturm et al. (2016)	Celtic	LSTM
Lattner et al. (2016)	C-RBM	Convolutional RBM
Mogren (2016)	C-RNN-GAN	GAN (RNN)
Sun (2016)	DeepHear M	Stacked AE
Jaques et al. (2016).	RL-Tuner	RNN & Reinforcement
Bretan et al. (2016)	UnitSelection	AE RNN
van der Oord et al. (2016)	WaveNet	CNN Conditional + Conv
Johnson (2015)	Hexahedria	Feedforward & RNN
Fabius and van Amers (2015)	VRAE	VAE (RNN)
Sarroff and Casey (2014)	deepAutoController	Stacked Controller Aes

Table 1

cont.

Model	Name	Architecture
Boulanger-Lewandow (2012)	RNN-RBM	RNN, RBM + sampling
Eck (2002)	Blues M	LSTM
Eck (2002)	Blues M&C	LSTM
Mozer (1994)	CONCERT	RNN

2.1.2. Representations

Automatic music generation researchers usually divide music representation into two sub-domains, symbol and audio (Ji et al., 2020). The main difference between these representations is that the symbol domain is represented as discrete variables, whereas audio is represented as continuous.

2.1.2.1. Symbolic Representation

Symbolic representations incorporate musical notions such as pitch, duration, chords, and others. It has two types of representations which are 1-D and 2-D representations. Examples of 1-D representation are event-based, and the most famous one is MIDI which is the structure used in this thesis work as well (see Table 2). The most used form of representation for 2-D is piano-roll representation. The piano roll image of music creation was inspired by the automated piano. The automated piano will have a continuous roll of perforated paper, with each perforation encoding note control information. (see Figure 5).

Table 2

MIDI events.

Task	Event	Description
Score	Note On	One for each pitch
	Note Off	One for each pitch
	Pitch	Note
	Instrument	Instrument information contained in a MIDI file
Performance	Note Velocity	MIDI velocity quantized into bins
	Tempo	Account for local changes in tempo (BPM)

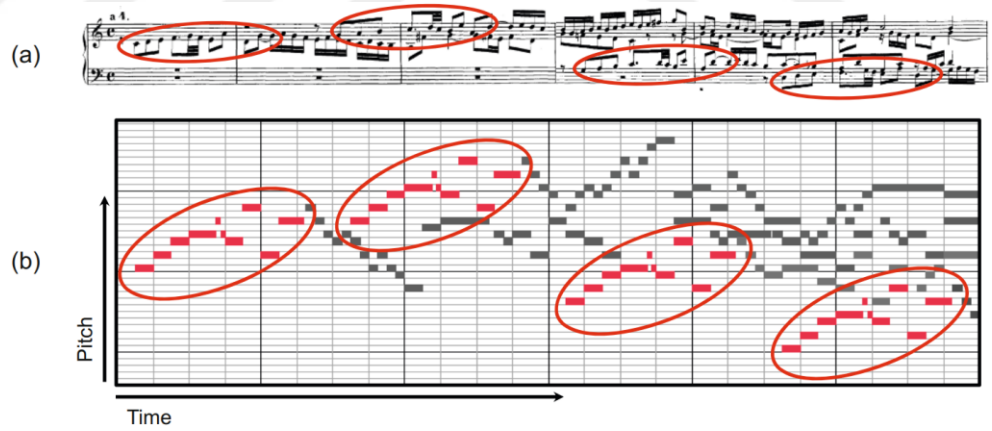


Figure 5: (a) Sheet music representation and its (b) piano-roll representation (Müller, 2015).

2.1.2.2. Audio Representation

Audio representations are continuous, focusing on acoustic information.

Similar to symbolic, audio also has 1-D and 2-D representations. The waveform is the corresponding main representation for 1-D, and various spectrograms are the examples for 2-D representation.

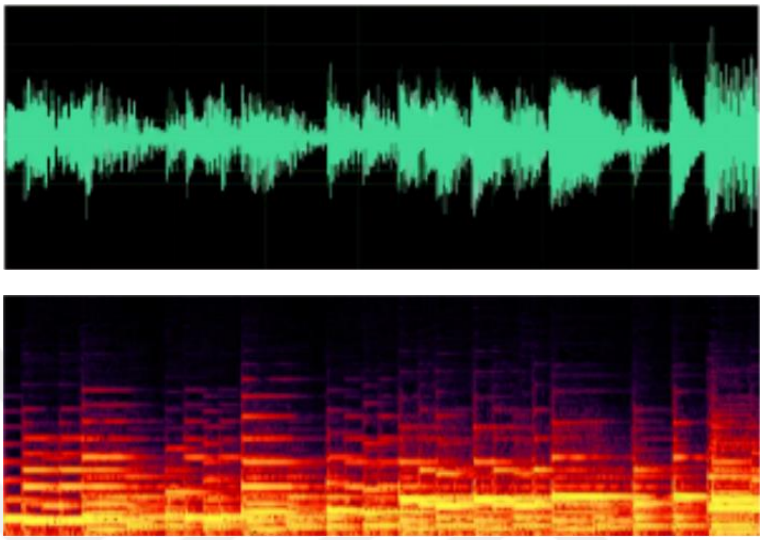


Figure 6: Waveform representation (above), spectrogram representation (below).

2.1.3. Evaluation

There are no standard rules for evaluating machine-generated art. Since it is a highly subjective task, the approach to evaluation is to divide the methods into two main groups: objective and subjective.

The first approach mainly refers to the quantitative evaluation of the generative model and the generated artifacts. Loss, Perplexity, BLEU score (see Chapter 2.3.3.), accuracy, recall, and F1 are the most often used model metrics for model performance scoring. There are approaches based on the task-dependent classification model accuracy evaluation. This is based on the pitch accuracy in which a classification model is trained based on the training data to evaluate the generated artifacts. In this work, the genre classification model process is adopted.

There are also summary statistics for music descriptors called signature vectors, such as the number of notes, occupation rate, polyphonic rate, pitch range descriptors, pitch interval range, and duration range. In a recent study, researchers first derived the music descriptors using these signature vectors, then the similarities between real and generated music distributions were calculated (Sabathé et al., 2017).

The other main evaluation procedure is the subjective approach. Human evaluators listen to the generated songs/melodies and try to give a judgment. Since objective evaluations are mostly not based on the musical quality, it may not be sufficient to evaluate the model and generated artifacts. Hence, human evaluation experiments are indispensable. These experiments could be listening tests; the participants try to determine whether the generated songs are real or fake, in other words, by applying the Turing test (Hadjeres et al., 2017; Cífka et al., 2019) or by comparing two/multiple and selection of music (Guan et al., 2019).

2.2. Image Classification and Object Detection

In machine learning, classification tasks require determining the correct categories/classes of inputs. For this reason, image classification research focuses on what is actually in images by trying to categorize them. The conventional and deep-learning-based image classification models roughly consist of two main parts. The first is the feature extraction part, and the second is where the actual classification happens based on the extracted features.

Image features are rich information found in the input images. They are helping local indicators for the natural scene that can be used for classification and other image analysis tasks, including recognition, matching, reconstruction,

compression, transformation, captioning, and others. Although the image captioning task will be explained later, it touches on many computer vision concepts, such as image classification, object detection, and even ranking. In this part, the focus will be based on classification and object detection.

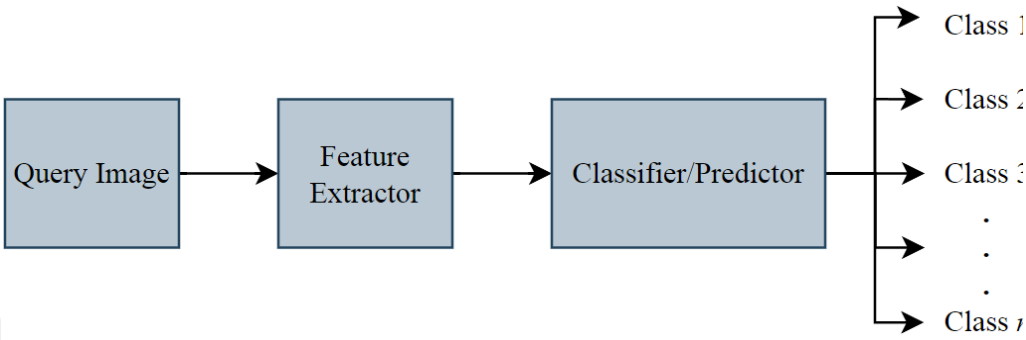


Figure 7: Common image classification model workflow.

2.2.1. Image Classification Methods

Before deep learning, sparse coding (Lee et al., 2006) using bag-of-visual-words (BoVW) (Csurka et al., 2004) is one of the most advanced methods for image classification tasks. A representation learning approach seeks to identify a sparse representation of the input image in a linear combination of fundamental elements and

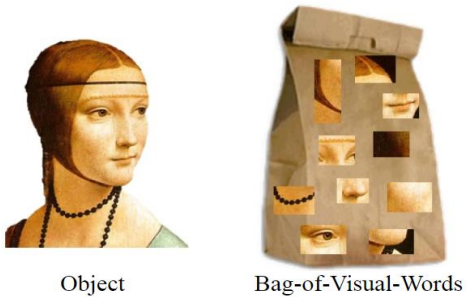


Figure 8: Inspired by NLP, a bag-of-words consist of essential parts of an image (Bag-of-Visual-Words) (pyimagesearch, 2022).

those elements themselves. A dictionary is made up of these elements known as atoms. The features can be a pixel intensity vector such as a dense SIFT-descriptor (Lowe, 2004; He et al., 2014). The extracted dictionary learned from unlabeled images is used in classifier code. Since labeled data is not needed, this approach is advantageous.

For this task, deep learning models can be divided into three parts: Autoencoders (AEs) (Hayat et al., 2014), Restricted Boltzmann machines (RBMs) (Luo et al., 2014; Courville et al., 2011), and Convolutional Neural Networks (CNNs) (LeCun et al., 2010).

AEs are an example of encoder-decoder architecture as they rely on information loss so that their decoding part tries to get an output as close as possible to the input data. Here, information loss simply means getting rid of unnecessary parts.

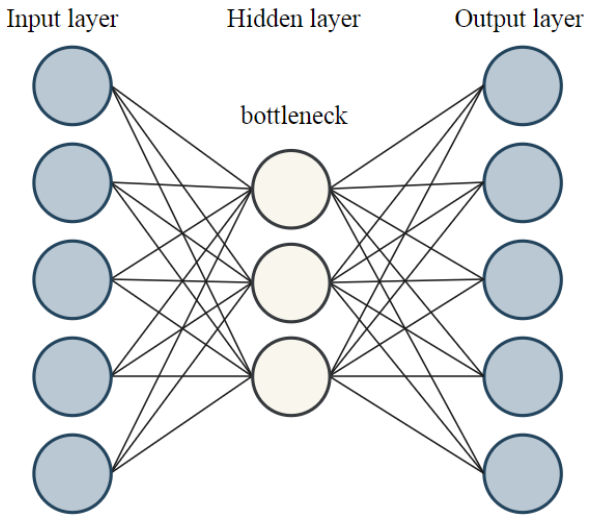


Figure 9: An example of basic Autoencoder architecture.

Compared with AEs, RBMs are stochastic neural network models with two densely-connected layers of visible and hidden states of a probabilistic system. Visible states take an input image, and hidden states try to classify by giving a probability distribution. RBMs can be used in deep belief networks (DBNs) (Lee et al., 2009), deep Boltzmann machines (DBM) (Hinton et al., 2012), deep AEs (Le et al., 2012), and CNN pre-training (Lee et al., 2009).

The layers in a CNN model represent feature maps that are a matrix of pixel intensities. Every pixel refers to a specific feature. It is typical to have filtering and pooling functions between convolutional layers. These functions add non-linearity to the neural network. Filtering extracts particular features, and pooling summarizes the features to make the network faster and more robust. The most popular pooling methods are maximizing, averaging, and L_2 pooling. Classifier layers are added to

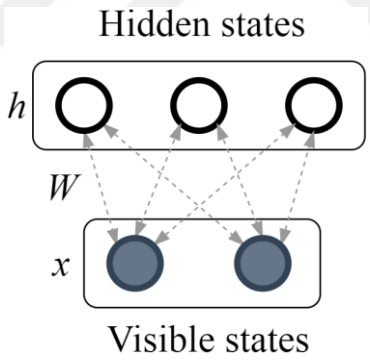


Figure 10: Restricted Boltzmann Machine.

CNNs with sigmoid and softmax as their activation functions (Kavukcuoglu et al., 2010; Krizhevsky et al., 2012; Oquab et al., 2014). The objective is to minimize the error, typically mean squared error or cross-entropy-loss, using the stochastic gradient descent (SGD) method. Finally, any type of classifier algorithm, such as support

vector machines (SVMs), can be applied to the end of the output layer. Any necessary image cropping and resizing can be applied here (Krizhevsky et al., 2012).

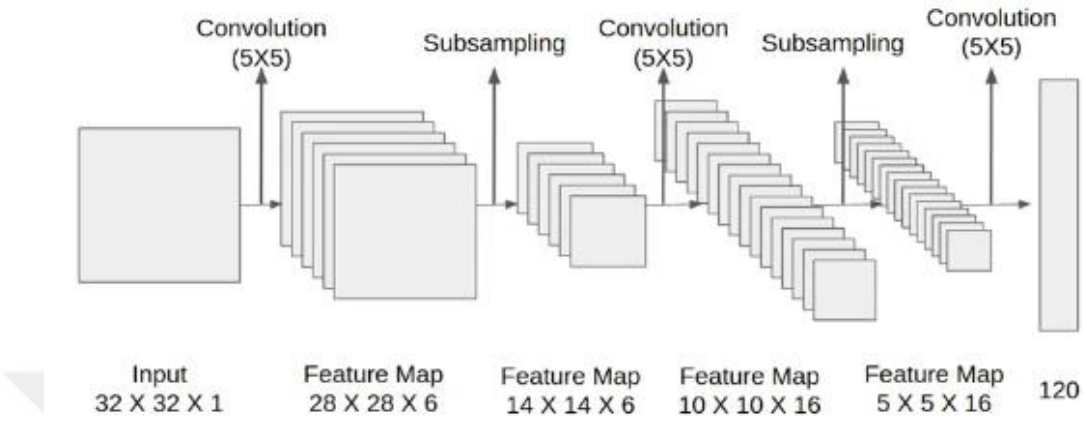


Figure 11: An example of a modern convolutional neural network architecture: Le-Net-5 for digit recognition (LeCun et al. 1998).

2.2.2. Object Detection Methods

The detection problem is more general. It entails determining if the object of interest is present in the image and determining the location of all of its occurrences. Object recognition remains a complex problem because of the many variables that must be considered: some of them are conceivable object shapes and colors, occlusions, lighting conditions, and perspective.

There are three types of detection methods: feature extraction, which is the only one that involves deep learning techniques, template searching, and movement detection. The most basic approach is to apply a deep classifier to areas of interest by utilizing a sliding window (Szegedy et al., 2013; Erhan et al., 2014; Oquab et al., 2014).

2.3. Image Captioning

Image captioning research deals with generating meaningful captions based on input images. It embodies both computer vision and natural language processing methodologies. The image captioning process first begins with the identification of the query images. The next step is to generate brief descriptions/text that must be correct both syntactically and semantically. Understanding what is contained in an image by a machine has helped with various purposes, such as evaluating image sentiments, having virtual assistants, indexing images, helping visually impaired people, and others.

This survey consists of three parts: image captioning with conventional machine learning-based methods, deep learning-based methods, and image captioning evaluation metrics.

2.3.1. Conventional Machine Learning-Based Methods

Before the domination of artificial deep neural networks, retrieval-based image captioning was the standard approach. The first step is to identify the visually close images from the training dataset to the query image. Then the *generated caption* may directly be selected from the caption pool. One way to achieve this is to plot the query image into the meaning space by solving a Markov Random Field using the Lin similarity metric (Lin, 1998). Then, each existing sentence is iterated by Curran and Clark parser (Curran et al., 2007). The closest caption to the query image is then chosen as the generated caption for the given image.

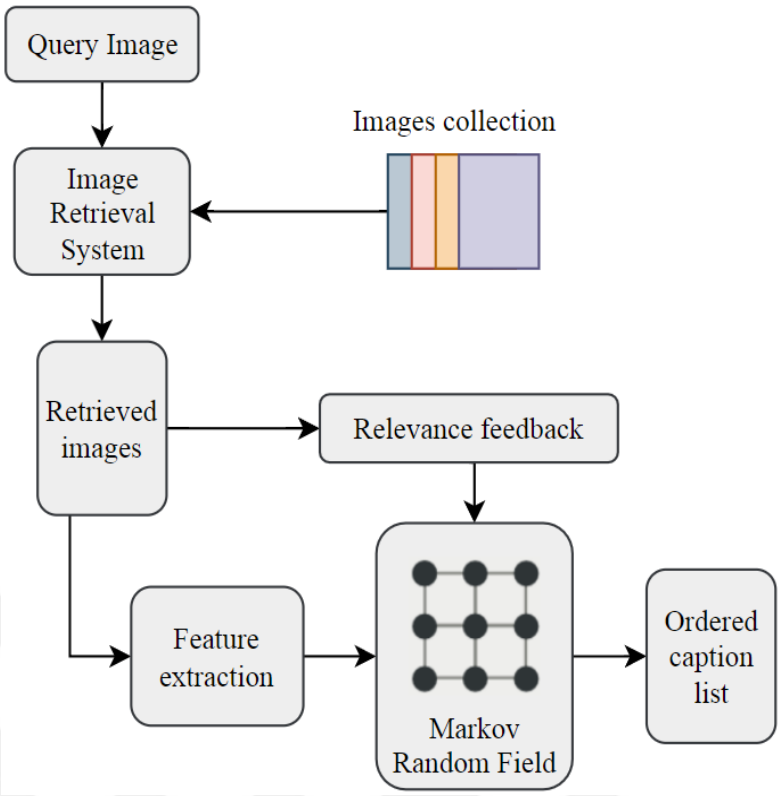


Figure 12: Block diagram of the retrieval-based image captioning model (Chávez et al., 2011).

This method is not thoroughly accurate. The limitations are that human-formed phrases used for the given image lead to grammatically correct captions; however, since it uses already constructed sentences, this approach does not open to mixed objects in new query images. The generated sentences may not be accurate to the natural scene.

2.3.2. Deep Learning-Based Methods

Using deep learning techniques for image captioning has soared recently. One of the approaches is the combination of retrieval-based and template-based methods and neural networks. The challenges of embedding and ranking can be overcome

using retrieval-based approaches. Earlier, researchers proposed a dependency tree recursive neural network representing phrases or words as compositional vectors for description retrieval (Socher et al., 2014). A max-margin maps the collected

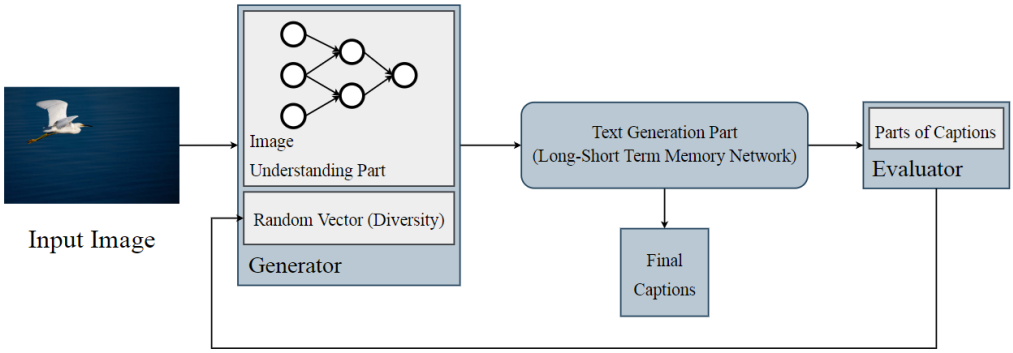


Figure 13: Common workflow of deep learning-based image captioning methods.

multimodal feature into a shared space. As a result of the evolution and adaption of new models, the performance captioning of picture approaches is improved when deep neural networks are used. However, the disadvantages of phrases created using retrieval and template-based approaches did not disappear.

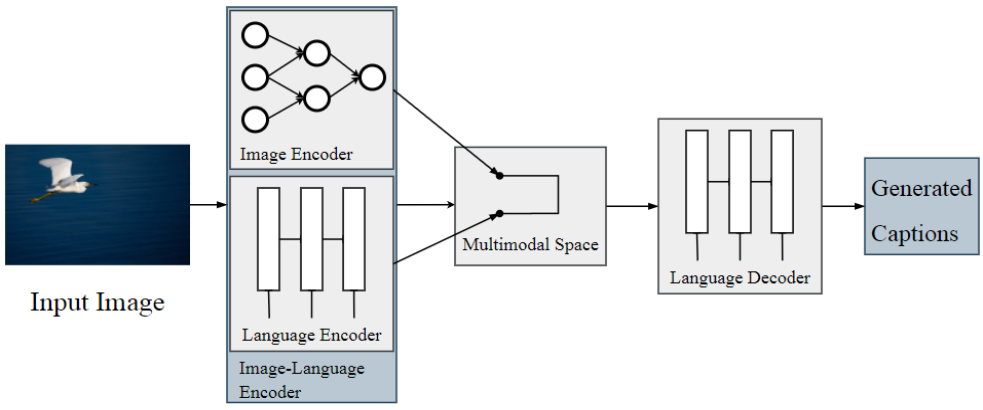


Figure 14: Multimodal-based image captioning methods.

Another strategy in image captioning is the multimodal learning-based models. They eliminate the limitation at the caption generation part by removing the assumption of having pre-existing caption phrases. This brings more flexible and expressive captions. One of the proposed models (Kiros et al., 2014) uses a log-bilinear language. However, others used recurrent neural network (RNN) (Mao et al., 2015) ([Rumelhart et al., 1985]) as a language model to obtain the probability of getting a word. These kinds of work also lay the foundation for encoder-decoder-based image captioning methodologies.

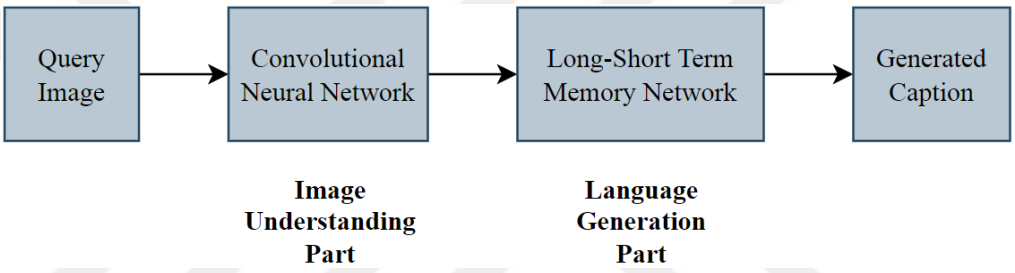


Figure 15: Common encoder-decoder architecture workflow for image captioning.

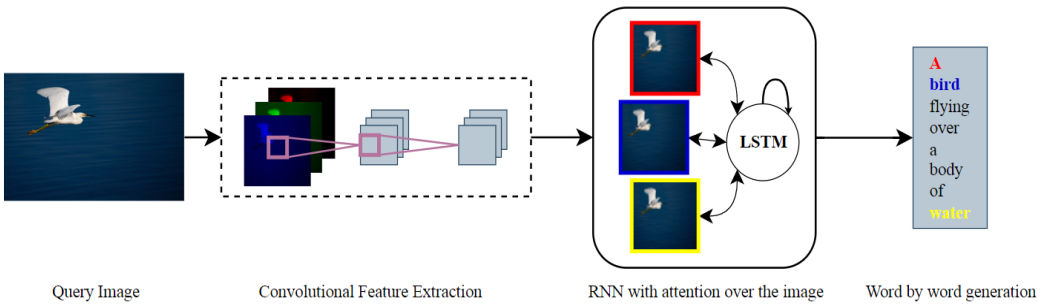


Figure 16: The diagram of *"Show, Attend and Tell"* paper (Xu et al., 2015). The model learns to make words/image alignment with the help of attentional maps. This model is the basis for this thesis for generating symbolic text music.

Encoder-decoder framework working principle is similar to the language translation approach as they use the sequence-to-sequence modeling, but with an important difference. Instead of using an RNN in multimodal learning-based models, an LSTM network (Hochreiter & Schimidhuber, 1997) is utilized. Both approaches use deep CNNs to extract image features. Although, the fundamental difference comes in the caption generation part.

2.3.3. Evaluation Metrics

Evaluating image captioning models is a pretty challenging task. The procedure compares the machine-generated captions with the human-generated captions to find the closest sentences, both syntactic- and semantic-wise. There are three main universally accepted evaluation metrics for this task.

1. *Bilingual Evaluation Understudy (BLEU)* (Papineni et al., 2002) is a method for assessing the quality of machine-translated text from one natural language to another. An output metric is a number between 0 and 1. Being close to 1 indicates a high similarity between sentences. Usually, four BLEU metrics are used for evaluation, BLEU-1, BLEU-2, BLEU-3, and BLEU-4, for unigram, bigram, trigram, and four-gram, respectively.

Table 3

The statistics of Flickr8k and Flickr30k datasets.

Dataset name	size			Total
	Training	Validation	Test	
Flickr8k	6000	1000	1000	8091
Flickr30k	28000	1000	1000	31783

Table 4

Evaluation of various models on Flickr8k dataset on BLEU metric

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Wu et al. (2017)	**74.0**	**54.0**	38.0	**27.0**
Jia et al. (2015)	64.7	45.9	31.8	21.6
Karpathy et al. (2015)	51.0	31.0	**52.0**	-
Vinyals et al. (2015)	63.0	41.0	27.0	-
Xu et al. (2015)	67.0	45.7	31.4	21.3
Kiros et al. (2014)	65.6	42.4	27.7	17.7
Mao et al. (2014)	56.5	38.6	25.6	17.0

2. *Recall-Oriented Understudy for Gisting Evaluation (ROGUE)* (Lin et al., 2004) uses a method by pairing words, sequences of words, and n-gram with human-generated sentences.
3. *Metric for Evaluation of Translation with Explicit ORdering (METEOR)* (Banerjee et al., 2005) performs a generalized unigram match between machine-generated text and human-generated references. In the case of several references, the most excellent score is chosen from those personally analyzed.

There are two benchmark datasets called Flickr8k (Hodosh et al., 2013) and Flickr30k (Plummer et al., 2015), consisting of around 8000 and 30000 annotated images. Table 2. shows the statistics regarding these datasets. Table 2 and Table 2 (Sharma et al., 2020). The top 15 SOTA (as of May 2022) model performances based on BLEU-4 scores are shown in Table 5.

Table 5

Evaluation of various models on Flickr30k dataset on BLEU metric.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Wu et al. (2017)	**74.0**	**54.0**	**38.0**	**27.0**
Jia et al. (2015)	64.6	46.6	30.5	20.6
Vinyals et al. (2015)	66.3	42.3	27.7	18.3
Xu et al. (2015)	66.9	43.9	29.6	19.9
Kiros et al. (2014)	60.0	38.0	25.4	17.1
Mao et al. (2014)	60.0	41.0	28.0	17.0

3. METHODOLOGY

This chapter explains the method followed in detail. First, it explains the dataset building process. The next section elaborates on the encoder-decoder architecture used in the model training for symbolic music generation through image captioning. The last section describes the music evaluation process with the (custom) music genre classification model and other music evaluation methods.

3.1. Implementation

The technology stack for this project is as follows (see Table 6):

1. The scripts (the dataset crawling) and the development of the model pipeline (exploratory data analysis [EDA], preprocessing, modeling, training, evaluation) are all written in Python3 (Rossum & Drake Jr, 1995).
2. Pandas (McKinney et al., 2010), an open-source software library for Python for data manipulation and analysis, is used to apply EDA and preprocess with metadata information.
3. scikit-image (Van der Walt et al., 2014) and fast-slic (Algy, 2019), open-source image processing libraries for the Python programming language, are specifically used for image preprocessing.
4. librosa (McFee et al., 2015) and pretty-midi (Raffel & Ellis, 2014), open-source software libraries for Python for music and audio analysis, are specifically used to feature engineering, extract features, and preprocess WAV and MIDI files, respectively.

5. scikit-learn (Pedregosa et al., 2011) and PyTorch (Paszke et al., 2019), open-source machine learning frameworks for Python, are also used to preprocess, build the main model, and the evaluation.
6. Machine A: A personal computer is used with the following features: an Intel^{R} Core™ i5-8600K CPU @ 3.60GHz, 16 GB RAM, NVIDIA GeForce GTX 1080 GPU, and 64-bit Windows 8.1 OS.
7. Machine B: A dedicated server is also used with the following features: V100-2XLarge, 8 Cores, 61 GB RAM

Table 6

Tech Stack used in this thesis.

Name	Version
Python3	3.9.8
Pandas	1.3.4
scikit-image	0.19.1
fast-slic	0.4.0
librosa	0.9.1
pretty-midi	0.2.9
scikit-learn	1.0.1
PyTorch	1.10.0+cu113

3.2. Dataset

The ultimate goal of this study is to create a model for image-to-music generation using image captioning. For this reason, image and music datasets are essential. The following sub-chapters give information about metadata and these datasets and their relevant importance.

3.2.1. Image Dataset

The image dataset consists of paintings from WikiArt (Saleh & Elgammal, 2015), a user-editable online visual art encyclopedia. It has over 120000 paintings where each artwork is annotated by its style, name, artist, date, and a URL containing the image file's location. These metadata are stored in a comma-separated value (CSV) file (see Table 7) (kaggle & antoinegruson, 2022).

Table 7

The fields of the WikiArt dataset.

Field	Description
Style	Style of the artwork.
Artwork	Name of the artwork.
Artist	Name of the artist.
Date	Creation date of the artwork.
Link	Download link to the artwork as a JPG/JPEG or PNG image.

Note. From "the WikiArt dataset" by WikiArt.org

The images are crawled using their corresponding URLs by using the metadata information. Although WikiArt has images from 217 unique styles/eras, this work is only interested in a smaller subset of them. These are *Baroque*, *Classicism*, *Romanticism*, *Art-Nouveau (Modern)*, *Divisionism*, *Impressionism*, *Post-Impressionism*, and *Symbolism* (see Table 8). Apart from the first three styles, the rest of them are gathered under a custom-umbrella style called *Modern* (see Table 8).

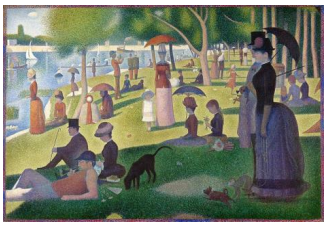
Table 8

Example images from the WikiArt dataset for each used "Style."

Style	Name	Painter	Year	Artwork
Baroque	The Night Watch	Rembrandt	1642	
Classicism	Achilles among the Daughters of Lycomedes	Pietro Paolini	1625–1630	
Romanticism	The Nightmare	Henry Fuseli	1781	
Art-Nouveau (Modern)	The Kiss	Gustav Klimt	1907–1908	

Table 8

cont.

Style	Name	Painter	Year	Artwork
Divisionism	A Sunday Afternoon on the Island of La Grande Jatte	Georges Seurat	1884–1886	
Impressionism	Woman with a Parasol – Madame Monet and Her Son	Claude Monet	1875	
Post-Impressionism	The Centenary of Independence	Henri Rousseau	1892	
Symbolism	Death and the Grave Digger	Carlos Schwabe	c. 1895	

3.2.2. Music Dataset

The music files come from the MIDI and Audio Edited for Synchronous TRacks and Organization (MAESTRO) (version 3.0.0) dataset (Hawthorn et al., 2018), which contains around 200 hours of virtuoso piano performances from Western classical music. The unique number of composers is 60, and it has 854 piano compositions. The certain statistics of the dataset can be seen in Table 9. Each recording in the dataset has seven fields which are *canonical_composer*, *canonical_title*, *split*, *year*, *midi_filename*, *audio_filename*, and *duration* (see Table 9).

Table 9

The statistics of the MAESTRO v3.0.0 dataset.

Split	Performances	Duration (hours)	Size (GB)	Notes (millions)
Train	962	159.2	96.3	5.66
Validation	137	19.4	11.8	0.64
Test	177	20.0	12.1	0.74
Total	1276	198.7	120.2	7.04

Note. From “the MAESTRO dataset” by Magenta, TensorFlow, Google.

3.2.3. Preprocessing

This section explains the preprocessing steps followed for image and symbolic music text datasets. It is one of the most crucial machine learning model development pipeline procedures. Preprocessing ensures data come from the same distribution.

Data imputation, normalization, and outlier removals are some of the processes applied to datasets from both domains.

3.2.3.1. Image Dataset Preprocessing

Image-text pairs associate image features with their corresponding captions for the image captioning task. For image-to-symbolic music generation, this work follows a similar fashion with an important distinction. The pairing process of images (paintings) and texts (symbolic music) is being done arbitrarily. It is certain that arbitrarily pairing images and symbolic music already brings much noise to the general distribution of the dataset. Therefore, it is quite important to have samples from the same distribution. For this reason, an anomaly/outlier removal procedure was followed using image features and superpixel-based images. Note that the outlier detection topic is out of the scope of this work. Hence, the only focus is on explaining the outlier removal process (see Figure 17).

An outlier is an observation in a set that deviates too much from the rest of the points in that set. For this work, the used outlier removal algorithm is the isolation forest (Liu et al., 2008). It is a unique anomaly detection method that relies on isolation (the distance between a data point and the rest of the data) rather than modeling normal points. The aim is to make all the images as much as from the same distribution for each *style*. The custom umbrella style, *Modern*, has images from the sub-styles of *Art-Nouveau (Modern)*, *Divisionism*, *Impressionism*, *Post-Impressionism*, and *Symbolism*. Hence, each style has gone through the same outlier

removal process (see Figure 17). Note that histograms[1] of HSV[2] images are calculated and normalized before applying the isolation forest.

The outlier removal algorithm is first applied to raw images to remove the 10% of data from each *Style*. Then, image features are extracted using a recently proposed image classification model called ViT-L/16 (Dosovitskiy et al., 2021). This is an application for a vision transformer (ViT) (Vaswani et al., 2017) model, which is also used as an encoder in the proposed model's training phase. The details of the encoder are shared in the following chapters.

After feature extraction, the isolation forest algorithm is again applied to remove the 20% data from each *Style*. Later, the remaining images are converted into superpixel-based versions by using SLIC segmentation. Superpixels are the outcome

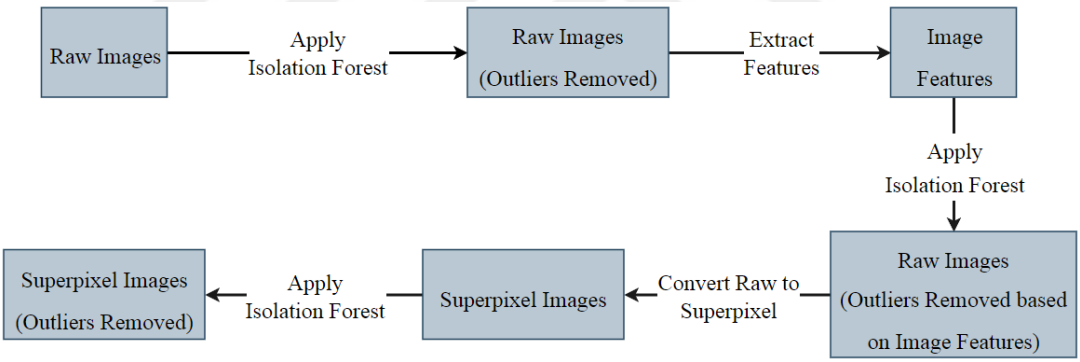


Figure 17: Outlier removal process from the image dataset.

of perceptual pixel grouping or the effect of picture over-segmentation. Superpixels store more information than pixels and better match picture borders than rectangular image patches. These converted images are also used in the training process. This has allowed having the gist (superpixel patterns) of each *Style* to effectively represent

[1] Histogram: a graphical representation that shows the approximate distribution of numerical data.
[2] HSV: An alternative way to represent RGB images.

their respective color palette, edges, contours, and other related features. Recent studies showed that having superpixel-based image features results in better performance for multispectral image classification (Liu et al., 2017; Zhao et al., 2017). The examples of the superpixelated versions of the raw images can be seen in Table 10.

Lastly, 60% and 10% of data were removed from *Modern* and from the rest of the data in each *Style*, respectively. After all of these steps, the change in the numbers can be seen in Table 10.

Table 10

The statistics of the selected styles of the WikiArt dataset used in this work. From the number of raw files to the number of files after each preprocessing step applied.

Style	size			
	Raw	Raw (10% of outliers removed)	Features extracted (20% of outliers removed)	Superpixels (60% and 10% of outliers removed from Modernism and from the rest, respectively)
Baroque	3600	3238	2590	2425
Classicism	276	248	198	187
Romanticism	3600	3240	2592	2338
Modernism	14816	13330	10663	3991

Art-Nouveau (Modern)	3600	-	-	-
Divisionism	421	-	-	-
Impressionism	3600	-	-	-
Post-Impressionism	3595	-	-	-
Symbolism	3600	-	-	-

Table 11

Examples of images from their original versions from the WikiArt dataset vs. their superpixelated versions.

Style	Original	Superpixel-based
Baroque		
Classicism		
Romanticism		

Table 11

cont.

Style	Original	Superpixel-based
Art-Nouveau (Modern)		
Divisionism		
Impressionism		

Table 11

cont.

Style	Original	Superpixel-based
Post-Impressionism		
Symbolism		

Arbitrarily pairing image and music files yield 2425 pairs for *Baroque*, 187 pairs for *Classic*, 2338 pairs for *Romanticism*, and 831 pairs for *Modern*. 95% and 5% of data split as the train and test sets, respectively. In total, there are 5451 training samples and 330 test samples. The final overview of train-test splits can be seen in Table 12.

Table 12

The statistics of the used dataset for image-symbolic music pairs for each style.

Style	size	
	Training	Test
Baroque	2303	122
Classical	178	9
Romanticism	2221	117
Modern	447	24
Total	5149	272

3.2.3.2. Music Dataset Preprocessing

Since pairing paintings and symbolic music is based on art movements; the music has to come from the same art movement to make pairing semantically correct in an abstract manner. Consequently, the selected styles are *Baroque*, *Classical*, *Romanticism*, and *Modern*. The base comes from the Cross-Era (audiolabs-erlangen, 2022) dataset to distinguish each canonical composer with its respective style. In this work, their styles are laid out chronologically (Figure 3). There are composers in-between two styles, which are called *transitional composers*. They are not included in this work.

The MIDI files in the MAESTRO dataset are separated accordingly to each composer's style. After the separation process, each MIDI file is split into 5-second mini-recordings (melodies) to improve the quality of the proposed music generation method through image captioning. These melodies are preprocessed to increase the reliability of the image processing task by holding only the melodies with a certain

Table 13

The metadata files in the Maestro dataset have seven fields for every MIDI/WAV pair.

Field	Description
canonical_composer	Composer of the piece.
canonical_title	Title of the piece.
split	Suggested train/validation/test split.
year	Year of performance.
midi_filename	MIDI filename.
audio_filename	WAV filename.
duration	Duration in seconds, based on the MIDI file.

number of notes. The related statistics can be seen in Table 14.

The symbolic music information is extracted from all the remaining music files by using the pretty-midi Python package. Note events can be obtained from a MIDI file by using this package. There are four events for each note: velocity, *pitch*, *note start time*, and *note end time*; e.g., the note events of a song from the MAESTRO dataset can be seen in Table 15. To decrease the vocabulary size of the decoder model during training, one should keep in mind that the velocity events from notes are discarded.

The MIDI encodings are treated as *captions* for image captioning. The encodings include information such as *tempo* and *note events*. Also, each note event is treated as a *word* as if they were in a natural language. Hence, they are concatenated with an underscore (_) character (see Table 16). These encodings build the

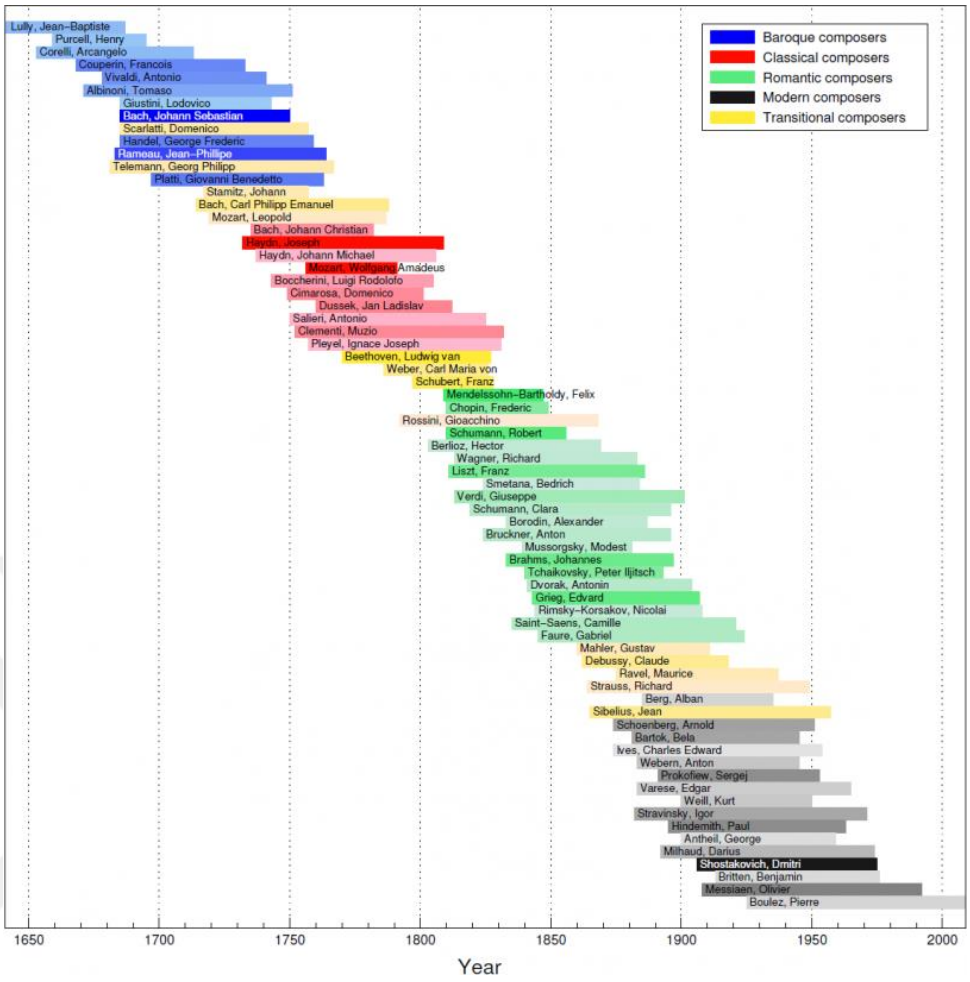


Figure 18: The chronology of Western classical music, based on composers. Transitional composers are not included in the music dataset for this work. From the Cross-Era Dataset by International Audio Laboratories Erlangen (audiolabs-erlangen, 2022).

vocabulary. The embedding part of the decoder and generating new captions use these words in the vocabulary as reference.

Table 14

Statistics of the used dataset, before and after the split operation.

Style	size		
	Original (before split)	After split	After preprocessing
Baroque	165	11719	
Classical	117	9434	
Romanticism	894	110890	
Modern	7	748	

Table 15

The events of the first ten notes from "Prelude and Fugue in E-flat Major" by Johann Sebastian Bach.

Velocity	Pitch	Start	End
42	G4	1.5	1.8
53	G#4	1.8	2.1
60	A#4	2.1	2.3
64	G#4	2.3	2.6
66	G4	2.5	2.8
64	F4	2.8	3.1
40	G3	3.8	4.1
50	G#3	4.1	4.3
52	A#3	4.3	4.6
52	G#3	4.6	4.9

Note. Each numeric value is rounded to 2 decimals.

Table 16

The tempo and the first 10 note events from the MIDI file of the third split of the composition, "Prelude and Fugue in E-flat Major" by Johann Sebastian Bach.

MIDI Event	Value
Tempo	230
Notes	
#1	D5_0.0_0.2
#2	D#5_0.2_0.5
#3	F5_0.0_0.7
#4	F5_0.5_0.7
#5	D#5_0.7_1.0
#6	D5_1.0_1.2
#7	C5_1.2_1.5
#8	D4_0.0_2.4
#9	D4_2.1_2.4
#10	D#4_2.4_2.7

3.3. Model Architecture

The machine learning architecture used in this study is based on the paper *"Show, Attend and Tell"* (Xu et al., 2015). It is an encoder-decoder architecture. The encoding part does image feature extraction. Using deterministic "soft" attentional maps, the important part of the image is determined. Then, extracted features are combined with the embedded captions, and the decoder part generates new captions, word-by-word.

3.3.1. Encoder: ResNet152

The model's encoder is a ResNet152 (He et al., 2015) which is used for image feature extraction. Using more robust encoder models for the encoder part of the image captioning task would yield semantically rich image descriptions (Hossain et al., 2019). Ultimately, leading to more robust image captioning, therefore, in this case, music generation.

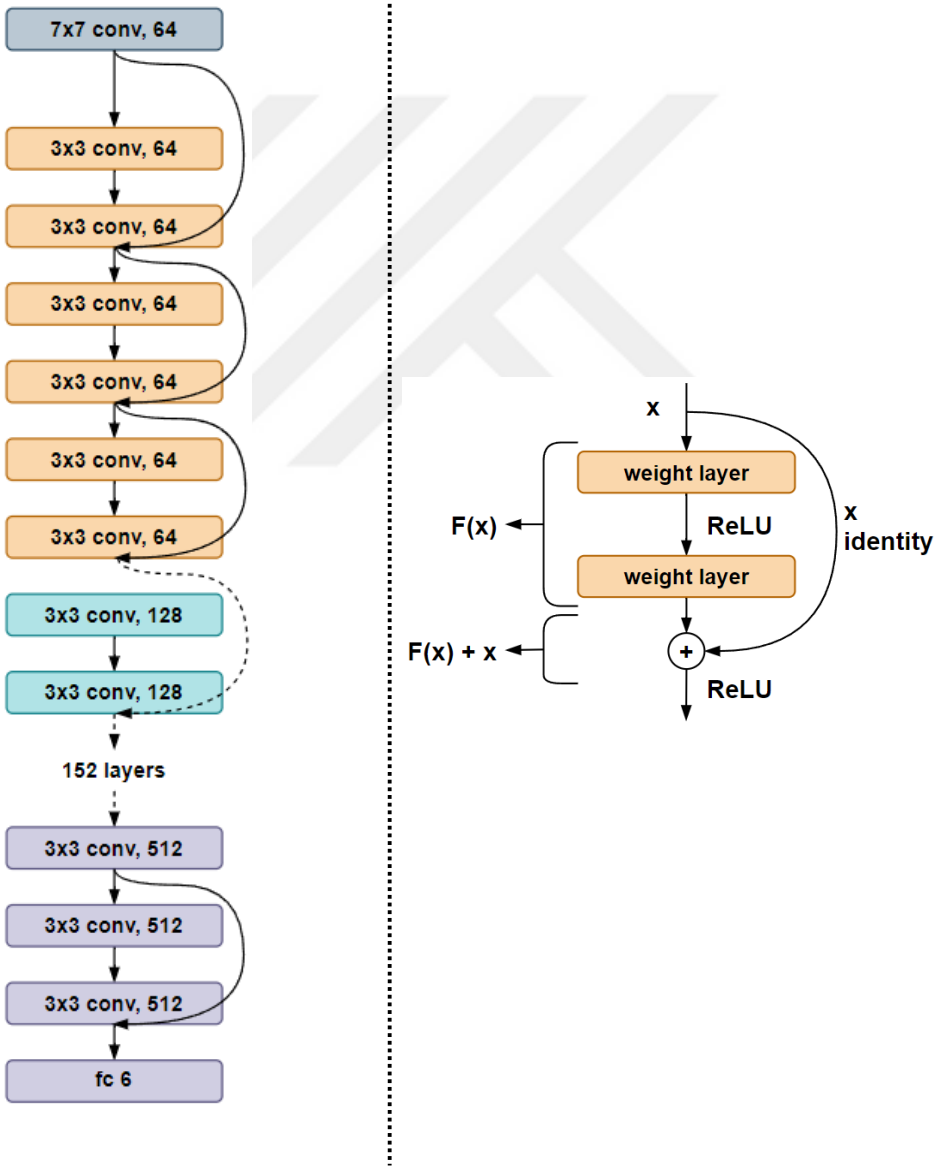


Figure 19: ResNet152 model overview (He et al., 2015).

For image feature extraction, the ResNet model from the Hugging Face machine learning models hub is used. It provides a feature extractor. Thus, the image features are easily extracted without the classification part of the ResNet, which is a basic multilayer perceptron (MLP) head (see Figure 19).

Before feeding into the ResNet encoder, query images are normalized and resized to 224x224. ResNet encoder outputs image features with a size of 224x224. Using a 14x14 window, an encoded image with 2048 learned channels is created, ready to be input to the attention block.

3.3.2. Soft Attention

The attention networks allow a model to select only the encoding parts that it believes are important to the current task. This approach may be used in any model with many points in space or time in the Encoder's output. In the case of image captioning, more importance is given to some pixels. The encoded image is weighted, indicating where the Decoder should focus its attention to create the next event (see Figure 20).

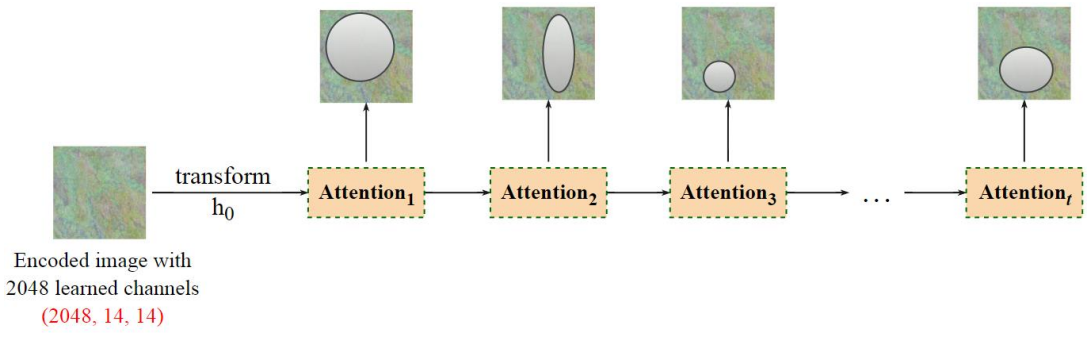


Figure 20: Soft attention network overview. The weighted encoded image indicates to the Decoder where it should give its attention.

3.3.3. Decoder: Long-Short Term Memory Network

Decoder part of the model is an LSTM network. It is used to generate captions (symbolic music) for query images. After getting embedding vectors from the symbolic music encodings, vectors coming from the attention block are concatenated. This creates the input to the LSTM network. Based on the sequence length of the reference caption, *tempo* and *note events* are generated one by one (see Figure 21).

A generated caption is compared with the reference caption, then the cross-entropy loss is calculated. The model learns how to compose better melodies by changing the model's weights using backpropagation (BP) (Rumelhart et al., 1986) to decrease the calculated loss.

An LSTM network consists of LSTM units. It is a more sophisticated version than an RNN (see Chapter 2.1.1). An LSTM unit comprises a cell, an input gate, an output gate (Hochreiter & Schmidhuber, 1996), and a forget gate (Gers et al., 2000).

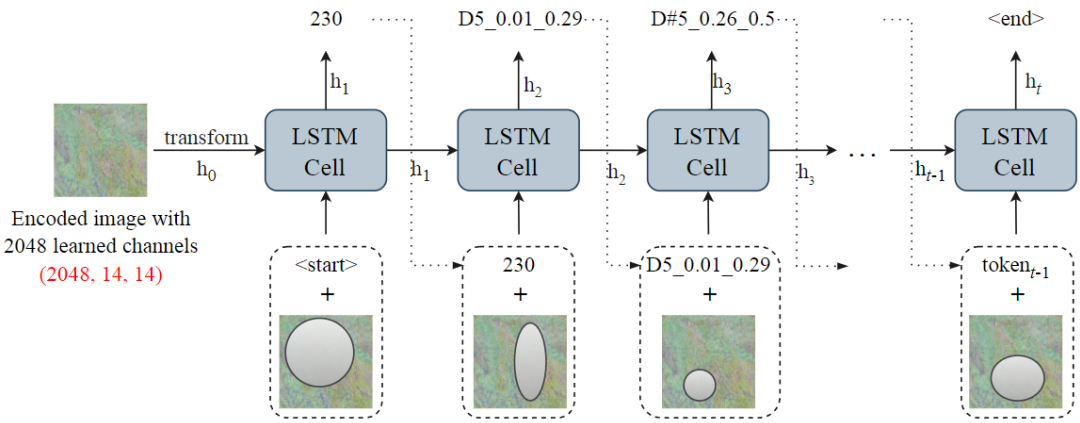


Figure 21: Overview of the proposed model's decoder (unrolled LSTM network).

The cell stores values for arbitrary time intervals and the three gates regulate the flow of data into and out of the cell (see Figure 22). A cell may process data in a sequential

manner while maintaining its hidden state. A series of equations for the forward pass of an LSTM cell can be seen below.

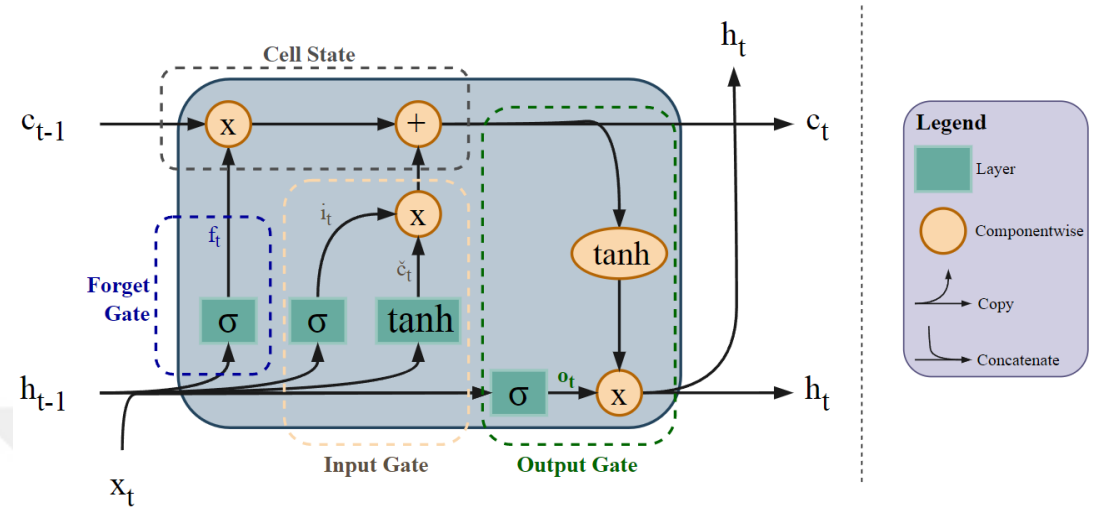


Figure 22: An LSTM cell.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{Eq. 1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{Eq. 2}$$

$$\check{c}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad \text{Eq. 3}$$

$$c_t = f_t * c_{t-1} + i_t * \check{c}_t) \quad \text{Eq. 4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{Eq. 5}$$

$$h_t = o_t * tanh(c_t) \quad \text{Eq. 6}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad \text{Eq. 7}$$

where,

$x_t \in \Re^d$, *input vector to the LSTM cell,*

$f_t \in (0, 1)^h$, *Forget Gate's activation vector,*

$i_t \in (0, 1)^h$, *Input Gate's activation vector,*

$$o_t \in (0, 1)^h, Output\ Gate's\ activation\ vector,$$

$$h_t \in (-1, 1)^h, hidden\ state\ vector\ of\ the\ LSTM\ unit,$$

$$\check{c}_t \in (-1, 1)^h, cell\ input\ activation\ vector,$$

$$c_t \in \Re^h, cell\ state\ vector,$$

$$W \in \Re^{hxd}, U \in \Re^{hxh}\ and\ b \in \Re^h, weight\ matrices\ and\ bias\ vector$$

$$parameters\ which\ need\ to\ be\ learned\ during\ training$$

3.3.4. Training

The image captioning architecture from "*Show, Attend and Tell*" (Xu et al., 2015) is used to train the machine learning model. As stated earlier, the model consists of two parts: encoder and decoder parts. Before feeding images to the encoder, all the images are normalized and transformed with resizing (224x224) and random cropping procedures. ResNet152 (He et al., 2015) architecture is used for the image encoder to extract features from paintings. The dimension of the encoder output is 2048.

The decoder part consists of an 8-layer deep-stacked LSTM network with soft attention. The attention network's dimension is 256. A vocabulary embeds the symbolic music texts prepared earlier with the tempo and music event tokens for each song's extracted split (see 3.2.3.2. Music Dataset Preprocessing). The embedding vector size is 300, and the decoder output dimension is 512. Since the training only involves updating the gradients of the decoder network, its learning rate is set to 3e-3.

The Adam optimizer (Kingma & Ba, 2014) is used for the optimization part. The training loss function is chosen as cross-entropy loss as the reference sequences

come from the symbolic music dataset. The model had been trained for 1200 epochs with a batch size of 64. The training took approximately 96 hours on Machine B. The best training loss was obtained at the 1200th epoch. After that, the loss started fluctuating and increasing. Therefore, the best model was taken at the specified epoch.

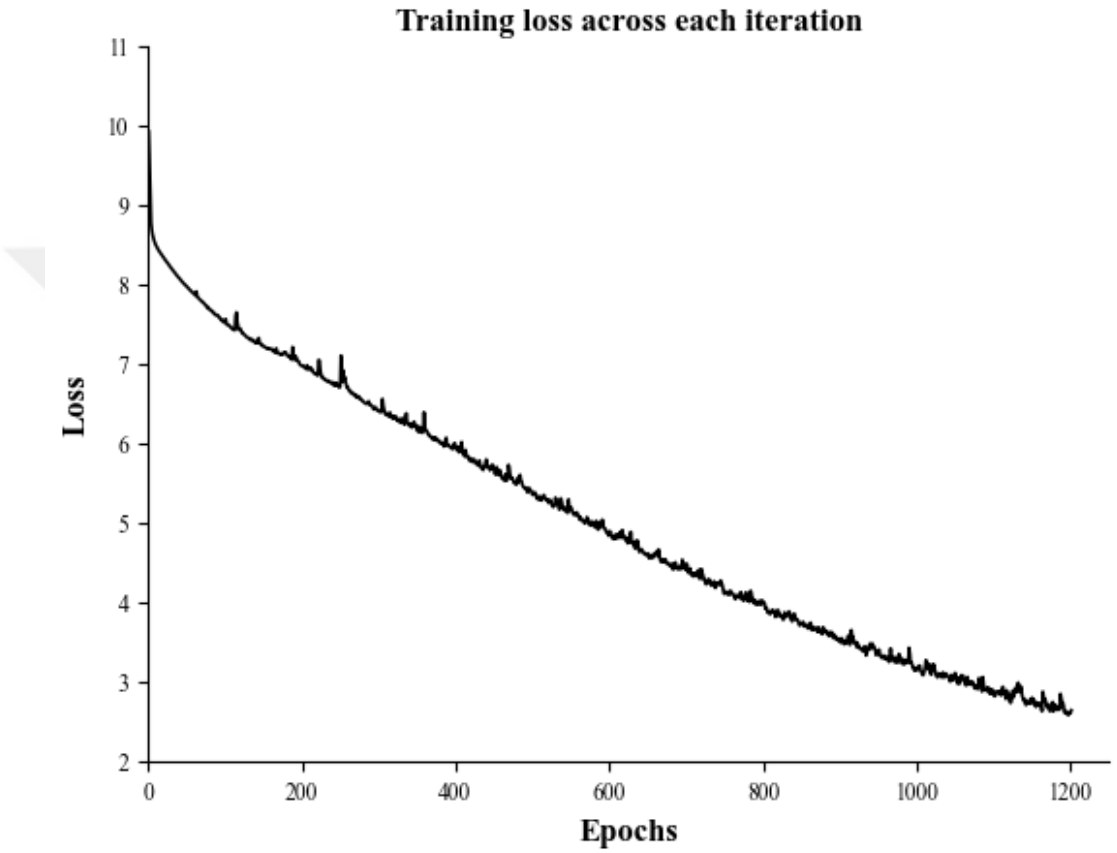


Figure 23: Training loss of deep-stacked LSTM Decoder.

In the end, many models tried to be trained with various numbers of layers for LSTM networks. Below 5-layer networks showed that they are trainable as the training loss kept decreasing, they all failed to capture the training data pattern, and their outputs were not good enough compared with the outputs of the 8-layer LSTM network. The training loss graphics can be seen in Figure 23.

3.3.5. Evaluation

The trained image-to-music model is evaluated by both quantitative and qualitative approaches. The quantitative approach involves three evaluations in itself. The first one is the model training loss values. The next two are evaluations of the generated melodies by genre classification model and BLEU scores calculation. For qualitative results, a listening test was carried out with human participants.

For the genre classification model development, feature engineering was carried out. First, the 5-second melodies extracted from the original MIDI files were converted into waveforms, and their various features were extracted using the Librosa Python library (McFee et al., 2015). The extracted features list can be found in Table 17.

Table 17

The extracted features for the genre classification task.

Features	**Descriptions**
n_notes	Number of notes
note1	Most common note
note2	Second most common note
note3	Third most common note
Extracted with Librosa	
chroma_stft	Chromagram from a waveform
chroma_cqt	Constant-Q chromagram
chroma_cens	Chroma variant “Chroma Energy Normalized” (CENS)
melspectrogram	Mel-scaled spectrogram
mfcc	Mel-frequency cepstral coefficients (MFCCs)

Table 17

cont.

rms	Root-mean-square (RMS) value for each frame
centroid	Spectral centroid
bandwidth	p'th-order spectral bandwidth.
contrast	Spectral contrast
flatness	Spectral flatness
rolloff	Roll-off frequency
p0	Coefficients of fitting a 1st-order polynomial to the columns of a spectrogram
p1	Coefficients of fitting a 2nd-order polynomial to the columns of a spectrogram
p2	Coefficients of fitting a 3rd-order polynomial to the columns of a spectrogram
tonnetz	Tonal centroid features
zero_crossing_rate	Zero-crossing rate of the audio time series

Note. The aggregation functions (mean, median, variance, min, max, sum) were applied to the features extracted with Librosa. These functions were also applied to absolute values of these features as well.

Later, an extra tree classifier is trained based on these extracted features to classify the genre of each melodic piece. This model is used to predict the classes of the generated melodies.

BLEU score calculation is based on the symbolic music text training data and the generated sequences. Since there is no exact matching between the training and

generated data, each generated symbolic text's BLEU score is calculated based on the training data from its corresponding style.

The last evaluation procedure is based on a listening test experimentation with human participants. The participants were asked to determine the style of the melodic piece after listening to them. 10 participants with no musical training participated in this experiment. The melodies from four styles were picked randomly. They first listened to 3 real melodies for each style to make participants learn the general pattern of a style. Then, they listened to 10 generated melodies where all of them were actually belonging to that style. The participants tried to answer two questions:

1. *"Do you think the melody you are listening to was written by a human or a machine?"*
2. *"Do you think that the music you listen to belongs to the X music period?"*

The first question is based on the Turing test whereas the second question explores whether the proposed model successfully generates melodies for the expected style if a painting with a certain style is given as input.

4. RESULTS

4.1. Quantitative Results

As stated earlier, quantitative results are based on the custom genre classification model and BLEU score calculations. The genre classification model is applied as four classes (styles) together, namely baroque, classical, romanticism, and modern. They are also examined as paired classes which are baroque-classical,

Table 18

Genre classification model results.

	accuracy (%)	**precision (%)**	**recall (%)**	**f1 (%)**
All classes				
Train (10 cross-validated)	81	83	83	83
Test (Human-generated)	83	83	83	83
Test (Machine-Generated)	43	43	43	43
Baroque-Classical				
Train (10 cross-validated)	96	97	97	97
Test (Human-generated)	97	97	97	97
Test (Machine-Generated)	89	89	89	89
Baroque-Romanticism				
Train (10 cross-validated)	90	93	93	93
Test (Human-generated)	92	92	92	92
Test (Machine-Generated)	52	52	52	52

Table 18

cont.

Baroque-Modern				
Train (10 cross-validated)	95	96	96	96
Test (Human-generated)	96	96	96	96
Test (Machine-Generated)	47	47	47	47
Classical-Romanticism				
Train (10 cross-validated)	95	95	95	95
Test (Human-generated)	95	95	95	95
Test (Machine-Generated)	93	93	93	93
Classical-Modern				
Train (10 cross-validated)	95	96	96	96
Test (Human-generated)	96	96	96	96
Test (Machine-Generated)	67	67	67	67
Romanticism-Modern				
Train (10 cross-validated)	88	88	88	88
Test (Human-generated)	88	88	88	88
Test (Machine-Generated)	75	75	75	75

baroque-romanticism, baroque-modern, classical-romanticism, classical-modern, and romanticism-modern. Accuracy, precision, recall, and f1 scores are calculated for these predictions. The results can be seen in Table 18.

The other quantitative results are based on BLEU scores. For each style, BLEU-1, BLEU-2, BLEU-3, and BLEU-4 scores are calculated. The calculations are separated into two. First, the scores for each generated melody's whole sequence are used. The next one is the calculation based on only the notes of each generated melody. These results can be seen in Table 19.

Table 19

Generated melodies BLEU scores based on the training data.

	BLEU-1	**BLEU-2**	**BLEU-3**	**BLEU-4**
Whole sequence				
Baroque	3.71	**23.0**	7.80	3.71
Classical	0	**3.83**	0	0
Romanticism	2.04	**15.04**	4.51	2.04
Modern	0.76	**3.92**	0.95	0.76
Only notes				
Baroque	94.66	**99.95**	99.79	94.66
Classical	57.28	**95.15**	86.96	57.28
Romanticism	95.23	**100**	99.93	95.23
Modern	77.35	**100**	98.16	77.35

4.2. Qualitative Results

The qualitative results are obtained by conducting a melody listening test with 10 human participants. These results can be seen in Table 20.

Table 20

The results of the melody listening test with human evaluators.

	Turing test pass (%)	Correct genre prediction (%)
Baroque	**49**	55
Classical	31	48
Romanticism	48	56
Modern	43	**73**

5. DISCUSSION

This work aimed to pave a map between two different modalities: image and music. Using the artifact pairs from the same art movement styles shows that creating a cross-modal generative model is possible. The adopted approach of image captioning also shows that images of paintings can be used to produce symbolic music sequences. This cross-modal learning technique showed that a direct relationship between image and music can be established without using bridge information.

5.1. Future Work

In the case of image captioning-based music generation, it is inevitable that having rich image features will lead to better results, meaning harmonious, coherent melodies. AI is a fast-growing field as a new SOTA image classification model is proposed almost every month. Changing the encoder with a SOTA model is one of the options, e.g., the recently published model, CoCa (Yu et al., 2022). Also, changing the decoder's LSTM network with more up-to-date methods, such as language transformers (Wolf et al., 2020), might yield good results.

Alternative image segmentation to SLIC can also be adopted for the image segmentation part. Changing the hyperparameter of the *number of components* in the SLIC algorithm may also yield different results as they may capture the patterns of a painting, hence, the style.

Using different MIDI encoding techniques can be used for the decoder part as well, such as REMI (Huang et al., 2020), compound word transformer (Hsiao et al., 2021), structured (Hadjeres & Crestel, 2021), Octuple (Zeng et al., 2021), and MuMIDI (Ren et al., 2020). Additionally, these encodings, represented as text, can be

converted to vectors, which might be used in the decoder part. Famous in NLP, word2vec is a text vectorization algorithm that is used to vectorize symbolic music (Chuan et al., 2020). Directly encoding the MIDIs is also an alternative. MIDI2vec (Lisena et al., 2020) is an example of this and can be used at the decoder part.

Since the first emergence of GAN, it has been excessively used for generative art tasks. For the image to music generation, image-to-image translation is an interesting approach proposed by Ruzafa (2020) using cGAN. It is one of the alternatives that can be examined in depth.

REFERENCES

Lejaren Jr, A., & Leonard M.. Isaacson. (1959). *Experimental Music: Composition with an Electronic Computer*. McGraw-Hill.

Rhodes, M. (1961). An analysis of creativity. *The Phi delta kappan*, *42*(7), 305-310.

Davenport, R. K., Rogers, C. M., & Russell, I. S. (1973). Cross modal perception in apes. *Neuropsychologia*, *11*(1), 21-28.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*. California Univ San Diego La Jolla Inst for Cognitive Science.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

Todd, P. M. (1989). A connectionist approach to algorithmic composition. *Computer Music Journal*, *13*(4), 27-43.

Mignot, C., Valot, C., & Carbonell, N. (1993, April). An experimental study of future "natural" multimodal human-computer interaction. In *INTERACT'93 and CHI'93 Conference Companion on Human Factors in Computing Systems* (pp. 67-68).

Stein, B. E., & Meredith, M. A. (1993). *The merging of the senses*. The MIT press.

Van Rossum, G., & Drake Jr, F. L. (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Hochreiter, S., & Schmidhuber, J. (1996). LSTM can solve hard long time lag problems. *Advances in neural information processing systems*, *9*.

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

Lin, D. (1998, July). An information-theoretic definition of similarity. In *Icml* (Vol. 98, No. 1998, pp. 296-304).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Storms, R. L. (1998). *Auditory-visual cross-modal perception phenomena*. NAVAL POSTGRADUATE SCHOOL MONTEREY CA.

Papadopoulos, G., & Wiggins, G. (1999, April). AI methods for algorithmic composition: A survey, a critical view and future prospects. In *AISB symposium on musical creativity* (Vol. 124, pp. 110-117).

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation*, *12*(10), 2451-2471.

Vroomen, J., & Gelder, B. D. (2000). Sound enhances visual perception: cross-modal effects of auditory organization on vision. *Journal of experimental psychology: Human perception and performance*, *26*(5), 1583.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (pp. 311-318).

Eck, D., & Schmidhuber, J. (2002). A first look at music composition using lstm recurrent neural networks. *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*, *103*, 48.

Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004, May). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV* (Vol. 1, No. 1-22, pp. 1-2).

Lin, C. Y., & Och, F. J. (2004, July). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)* (pp. 605-612).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91-110.

Banerjee, S., & Lavie, A. (2005, June). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization* (pp. 65-72).

Lee, H., Battle, A., Raina, R., & Ng, A. (2006). Efficient sparse coding algorithms. *Advances in neural information processing systems*, *19*.

Vines, B. W., Krumhansl, C. L., Wanderley, M. M., & Levitin, D. J. (2006). Cross-modal interactions in the perception of musical performance. *Cognition*, *101*(1), 80-113.

Curran, J. R., Clark, S., & Bos, J. (2007, June). Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the 45th annual meeting of the Association for Computational Linguistics Companion volume proceedings of the demo and poster sessions* (pp. 33-36).

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In *2008 eighth ieee international conference on data mining* (pp. 413-422). IEEE.

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009, June). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning* (pp. 609-616).

Nierhaus, G. (2009). *Algorithmic composition: paradigms of automated music generation*. Springer Science & Business Media.

Kavukcuoglu, K., Sermanet, P., Boureau, Y. L., Gregor, K., Mathieu, M., & Cun, Y. (2010). Learning convolutional feature hierarchies for visual recognition. *Advances in neural information processing systems, 23*.

McKinney, W., & others. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011, January). Multimodal deep learning. In *ICML*.

Courville, A., Bergstra, J., & Bengio, Y. (2011, June). A spike and slab restricted Boltzmann machine. In *Proceedings of the fourteenth international conference*

on artificial intelligence and statistics (pp. 233-241). JMLR Workshop and Conference Proceedings.

Pedregosa, F., Varoquaux, Gael, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Hinton, G. E., & Salakhutdinov, R. R. (2012). A better way to pretrain deep boltzmann machines. *Advances in Neural Information Processing Systems*, *25*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

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

Hodosh, M., Young, P., & Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, *47*, 853-899.

Le, Q. V. (2013, May). Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing* (pp. 8595-8598). IEEE.

Pereira, J. C., Coviello, E., Doyle, G., Rasiwasia, N., Lanckriet, G. R., Levy, R., & Vasconcelos, N. (2013). On the role of correlation and abstraction in cross-

modal multimedia retrieval. *IEEE transactions on pattern analysis and machine intelligence*, *36*(3), 521-535.

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. *Advances in neural information processing systems*, *26*.

He, Y., Kavukcuoglu, K., Wang, Y., Szlam, A., & Qi, Y. (2014, April). Unsupervised feature learning by deep sparse coding. In *Proceedings of the 2014 SIAM international conference on data mining* (pp. 902-910). Society for Industrial and Applied Mathematics.

Kiros, R., Salakhutdinov, R., & Zemel, R. (2014, June). Multimodal neural language models. In *International conference on machine learning* (pp. 595-603). PMLR.

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014, June). Weakly supervised object recognition with convolutional neural networks. In *Proc. of NIPS* (Vol. 2014, pp. 1545-5963).

Raffel, C., & Ellis, D. P. (2014, October). Intuitive analysis, creation and manipulation of MIDI data with pretty_midi. In *15th international society for music information retrieval conference late breaking and demo papers* (pp. 84-93).

Feng, F., Wang, X., & Li, R. (2014, November). Cross-modal retrieval with correspondence autoencoder. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 7-16).

Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2147-2154).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Hayat, M., Bennamoun, M., & An, S. (2014). Learning non-linear reconstruction models for image set classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1907-1914).

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

Luo, P., Tian, Y., Wang, X., & Tang, X. (2014). Switchable deep network for pedestrian detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 899-906).

Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., & Yuille, A. (2014). Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*.

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1717-1724).

Socher, R., Karpathy, A., Le, Q. V., Manning, C. D., & Ng, A. Y. (2014). Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, *2*, 207-218.

Van der Walt, S., Schonberger, Johannes L, Nunez-Iglesias, J., Boulogne, Franccois, Warner, J. D., Yager, N., … Yu, T. (2014). scikit-image: image processing in Python. *PeerJ*, *2*, e453.

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., & Nieto, O. (2015, July). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference* (Vol. 8, pp. 18-25).

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ... & Bengio, Y. (2015, June). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048-2057). PMLR.

Tanveer, M. I., Liu, J., & Hoque, M. E. (2015, October). Unsupervised extraction of human-interpretable nonverbal behavioral cues in a public speaking scenario. In *Proceedings of the 23rd ACM international conference on Multimedia* (pp. 863-866).

Jia, X., Gavves, E., Fernando, B., & Tuytelaars, T. (2015). Guiding the long-short term memory model for image caption generation. In *Proceedings of the IEEE international conference on computer vision* (pp. 2407-2415).

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3128-3137).

Müller, M. (2015). *Fundamentals of music processing: Audio, analysis, algorithms, applications* (Vol. 5). Cham: Springer.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., & Lazebnik, S. (2015). Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision* (pp. 2641-2649).

Saleh, B., & Elgammal, A. (2015). Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *arXiv preprint arXiv:1505.00855*.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156-3164).

Heath, D., & Ventura, D. (2016, June). Before a computer can draw, it must first learn to see. In *Proceedings of the 7th international conference on computational creativity* (pp. 172-179).

Aytar, Y., Vondrick, C., & Torralba, A. (2016). Soundnet: Learning sound representations from unlabeled video. *Advances in neural information processing systems, 29*.

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).

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*.

Ponce de León, P. J., Inesta, J. M., Calvo-Zaragoza, J., & Rizo, D. (2016). Data-based melody generation through multi-objective evolutionary computation. *Journal of Mathematics and Music*, *10*(2), 173-192.

Mogren, O. (2016). C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*.

Owens, A., Isola, P., McDermott, J., Torralba, A., Adelson, E. H., & Freeman, W. T. (2016). Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2405-2413).

Sturm, B. L., Santos, J. F., Ben-Tal, O., & Korshunova, I. (2016). Music transcription modelling and composition using deep learning. *arXiv preprint arXiv:1604.08723*.

Li, B., Dinesh, K., Duan, Z., & Sharma, G. (2017, March). See and listen: Score-informed association of sound tracks to players in chamber music performance videos. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2906-2910). IEEE.

Sabathé, R., Coutinho, E., & Schuller, B. (2017, May). Deep recurrent music writer: Memory-enhanced variational autoencoder-based musical score composition and an objective measure. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 3467-3474). IEEE.

Hadjeres, G., Pachet, F., & Nielsen, F. (2017, July). Deepbach: a steerable model for bach chorales generation. In *International Conference on Machine Learning* (pp. 1362-1371). PMLR.

Elgammal, A., Liu, B., Elhoseiny, M., & Mazzone, M. (2017). Can: Creative adversarial networks, generating "art" by learning about styles and deviating from style norms. *arXiv preprint arXiv:1706.07068*.

Liu, T., Gu, Y., Chanussot, J., & Dalla Mura, M. (2017). Multimorphological superpixel model for hyperspectral image classification. *IEEE Transactions on geoscience and remote sensing*, *55*(12), 6950-6963.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wu, Q., Shen, C., Wang, P., Dick, A., & Van Den Hengel, A. (2017). Image captioning and visual question answering based on attributes and external knowledge. *IEEE transactions on pattern analysis and machine intelligence*, *40*(6), 1367-1381.

Zhao, W., Jiao, L., Ma, W., Zhao, J., Zhao, J., Liu, H., ... & Yang, S. (2017). Superpixel-based multiple local CNN for panchromatic and multispectral

image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(7), 4141-4156.

Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C. Z. A., Dieleman, S., ... & Eck, D. (2018). Enabling factorized piano music modeling and generation with the MAESTRO dataset. *arXiv preprint arXiv:1810.12247*.

Lyu, J., Shinozaki, T., & Amano, K. (2018). Generating Images from Sounds Using Multimodal Features and GANs.

Owens, A., & Efros, A. A. (2018). Audio-visual scene analysis with self-supervised multisensory features. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 631-648).

Zukowski, Z., & Carr, C. J. (2018). Generating black metal and math rock: Beyond bach, beethoven, and beatles. *arXiv preprint arXiv:1811.06639*.

Cramer, J., Wu, H. H., Salamon, J., & Bello, J. P. (2019, May). Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 3852-3856). IEEE.

Wan, C. H., Chuang, S. P., & Lee, H. Y. (2019, May). Towards audio to scene image synthesis using generative adversarial network. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 496-500). IEEE.

Guan, F., Yu, C., & Yang, S. (2019, July). A gan model with self-attention mechanism to generate multi-instruments symbolic music. In *2019 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-6). IEEE.

Algy. (2019). fast-slic. Retrieved from: https://pypi.org/project/fast-slic/ Assessed: 30.04.2022

Cífka, O., Şimşekli, U., & Richard, G. (2019). Supervised symbolic music style translation using synthetic data. *arXiv preprint arXiv:1907.02265*.

Hossain, M. Z., Sohel, F., Shiratuddin, M. F., & Laga, H. (2019). A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR), 51*(6), 1-36.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32* (pp. 8024–8035). Curran Associates, Inc. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Sharma, H., Agrahari, M., Singh, S. K., Firoj, M., & Mishra, R. K. (2020, February). Image captioning: a comprehensive survey. In *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)* (pp. 325-328). IEEE.

Huang, Y. S., & Yang, Y. H. (2020, October). Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In

Proceedings of the 28th ACM International Conference on Multimedia (pp. 1180-1188).

Ren, Y., He, J., Tan, X., Qin, T., Zhao, Z., & Liu, T. Y. (2020, October). Popmag: Pop music accompaniment generation. In *Proceedings of the 28th ACM International Conference on Multimedia* (pp. 1198-1206).

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020, October). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 38-45).

Carnovalini, F., & Rodà, A. (2020). Computational creativity and music generation systems: An introduction to the state of the art. *Frontiers in Artificial Intelligence*, *3*, 14.

Chuan, C. H., Agres, K., & Herremans, D. (2020). From context to concept: exploring semantic relationships in music with word2vec. *Neural Computing and Applications*, *32*(4), 1023-1036.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Ji, S., Luo, J., & Yang, X. (2020). A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv preprint arXiv:2011.06801*.

Lisena, P., Meroño-Peñuela, A., & Troncy, R. (2020). MIDI2vec: Learning MIDI embeddings for reliable prediction of symbolic music metadata. *Semantic Web*, (Preprint), 1-21.

Rivas Ruzafa, E. (2020). *Pix2Pitch: generating music from paintings by using conditionals GANs* (Doctoral dissertation, ETSI_Informatica).

Hadjeres, G., & Crestel, L. (2021). The piano inpainting application. *arXiv preprint arXiv:2107.05944*.

Hsiao, W. Y., Liu, J. Y., Yeh, Y. C., & Yang, Y. H. (2021). Compound Word Transformer: Learning to compose full-song music over dynamic directed hypergraphs. *arXiv preprint arXiv:2101.02402*.

Zeng, M., Tan, X., Wang, R., Ju, Z., Qin, T., & Liu, T. Y. (2021). Musicbert: Symbolic music understanding with large-scale pre-training. *arXiv preprint arXiv:2106.05630*.

audiolabs-erlangen. (2022). Cross-Era Dataset. Retrieved from: https://www.audiolabs-erlangen.de/resources/MIR/ Assessed: 30.04.2022

kaggle, antoinegruson. (2022). WikiArt | All images (120k+) Retrieved from: https://www.kaggle.com/datasets/antoinegruson/-wikiart-all-images-120k-link/download Assessed: 30.04.2022

pyimagesearch. (2022). The bag of (visual) words model. Retrieved from: https://customers.pyimagesearch.com/the-bag-of-visual-words-model/ Assessed: 30.04.2022

Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., & Wu, Y. (2022). CoCa: Contrastive Captioners are Image-Text Foundation Models. *arXiv preprint arXiv:2205.01917*.

Appendix A: List of Symbols / Abbreviations

- 1-D -- 1-dimensional
- 2-D -- 2-dimensional
- AE -- Autoencoder
- AGI -- Artificial General Intelligence
- AI -- Artificial Intelligence
- BoVW -- Bag-of-Visual-Words
- BoW -- Bag-of-Words
- BLEU -- Bilingual Evaluation Understudy
- BPM -- Beats Per Minute
- CAN -- Creative Adversarial Network
- CCS -- Computationally Creative System
- CENS -- Chroma Energy Normalized
- cGAN -- Conditional Generative Adversarial Network
- CNN -- Convolutional Neural Network
- Conv -- Convolution
- CPU -- Central Processing Unit
- CSV -- Comma-Separated Values
- DBM -- Deep Boltzmann Machine
- DBN -- Deep Belief Networks
- EDA -- Exploratory Data Analysis
- GAN -- Generative Adversarial Network
- GPU -- Graphics Processing Unit
- HSV -- Hue, Saturation, Value

- JPG -- Joint Photographic Experts Group
- JPEG -- Joint Photographic Experts Group
- LSTM -- Long-Short Term Memory
- MAESTRO -- MIDI and Audio Edited for Synchronous TRacks and Organization
- MSE -- Mean Squared Error
- METEOR -- Metric for Evaluation of Translation with Explicit ORdering
- MFCC -- Mel-frequency cepstral coefficient
- MGS -- Music Generation System
- MIDI -- Musical Instrument Digital Interface
- MuMIDI -- MUlti-track MIDI representation
- ML -- Machine Learning
- MLP -- Multilayer Perceptron
- NLP -- Natural Language Processing
- OS -- Operating System
- PNG -- Portable Network Graphics
- RAM -- Random-Access Machine
- REMI -- Revamped MIDI
- RGB -- Red, Green, Blue
- RNN -- Recurrent Neural Network
- RL -- Reinforcement Learning
- ROGUE -- Recall-Oriented Understudy for Gisting Evaluation
- ROI -- Region of Interest
- RBM -- Restricted Boltzmann Machine

- SGD -- Stochastic Gradient Descent
- SLIC -- Simple Linear Iterative Clustering
- SOTA -- State-of-the-Art
- SVM -- Support Vector Machine
- URL -- Uniform Resource Locator
- VAE -- Variational Autoencoder
- ViT -- Vision Transformer
- WAV -- Waveform Audio File Format
- WAVE -- Waveform Audio File Format