# Deep Learning for Optical Music Recognition

Wen Li
wen3@ualberta.ca

## 1. Introduction

With the development of society and the improvement of living standards, quality education has been emphasized. An increasing number of people prefer to learn a kind of musical instrument which edifies their sentiment. However, those rookies who are eager to practise more by themselves in their leisure time would find it hard to understand the meaning of musical notation, thereby giving up halfway.

Even though there are huge collections of music notation that have been published on many websites or by printed books in libraries, many people will not be able to identify the character effortlessly until they learn over the course in school, especially when there are a number of formats of musical notation. Like characters, music notation has handwritten and printed forms. The files differ strongly regarding their quality when they are converted to flat files on the web page or paper. Some have better resolution than others, some show strong distortions near the edges of the page or slight rotations throughout [1].

Machine learning technique has been considered as an effective method of building a system that allows the computer to learn how to solve the problems themselves. In this work, the computer would automatically classify the notes into certain types. Deep learning acts as a subset of machine learning that excels at processing image data. In this way, we compared different deep learning neural networks based on the accuracy of model performance.

## 2. Background

### 2.1 Basics of music notation

The music notation is the simplest way to represent tones of any given piece of music. But there are a variety of standards of naming the tones owing to different usage specifications. In this work, we mainly focus on German literature where the Middle C (261.6 Hz) is designated by the name $c^1$ in Figure 1 [2].

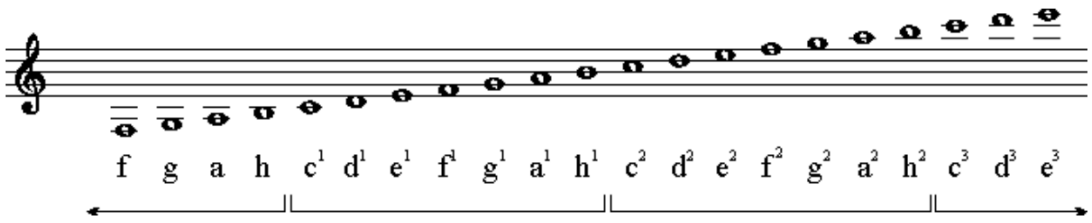



FIG 1 German music notation

Specifically, the five horizontal lines through the notation are called staff which is the basis of different frequencies of tones [3]. The frequency of a tone named pitch means how high or low it sounds. The higher position of a tone, the higher it sounds. Besides that, there is also a duration concept which decides how long a note will sound within a bar (the central time unit). In Figure 2, whole notes take up an entire bar, half notes sound half as long as whole notes and so on [4]. Similarly, the rest notes can be analogized to general notes but with different representations as shown in Figure 3.

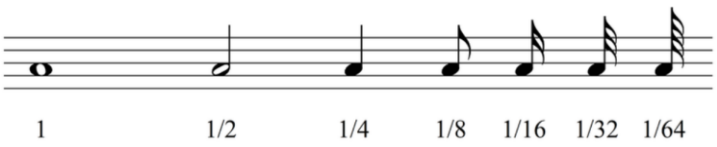



FIG 2 Note durations

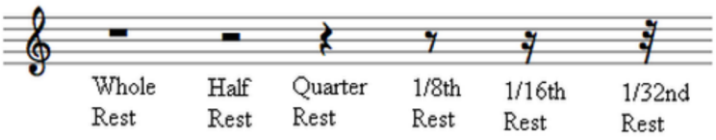



FIG 3 Rest durations

## 2.2 Data samples

The available data set comes from [1]. It comprises 3824 examples in total, each showing a single note which is more convenient to train and predict than multiple notes at a time [5]. Every image has the same size of 30 x 50 pixels which is shown in Figure 4 [1].

FIG 4 A single note with 30x50 pixels

As a result, all the images consist of 1,500 pixels and every image was also assigned with a unique filename, representing different types of notes for further data processing.

## 3. Proposed system

### 3.1.1 Fully connected neural networks (FCNNs)

The fully connected neural networks is a basic type of artificial neural network where all nodes from one layer are connected to all nodes in the next layer as shown in Figure 5.

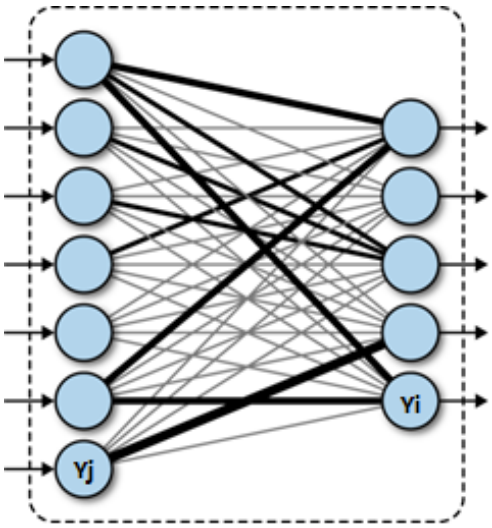



FIG 5 Fully connected neural networks

Since we have multiple layers, we need recursive algorithms that compute the activation function of all nodes automatically, namely, forward propagation and backward propagation. The output of each dimension depends on the input dimension and the output can be computed as follows:

$$z_i = {}^{N_{n-1}}_{j=1}\Sigma w_{ij} y_j^{l-1} + b_i$$

$$y_i^{(l)} = f(z_i)^{(l)}$$

where f is an activation function e,g., sigmoid, tanh, relu. We call the output of the current layer as y and the input of current layer y, which is the output the lower layer.

### 3.1.2 Convolutional neural networks (CNNs)

Convolutional neural networks (CNNs) is the most widely applied model that is made up of an input layer, hidden layers, and an output layer in deep learning. In the meanwhile, CNNs has the advantage of processing image data because the patterns it learns are fundamentally translation invariant [22]. In other words, it does not need larger training samples to learn representations. Secondly, the complex hierarchical patterns could also be learned efficiently by using CNNs.

Convolutions focus on operating 3D tensors, called a feature map which has two spatial axes

(height and width) and a depth axis also known as channels axis [6]. The input feature map provides patches for convolutions first and the output feature map would be produced by applying transformations to these patches.

In 2D CNNs, convolutions select a window from the input feature map and slide it until all the features are traversed. Every selection has its 3D patch features transformed by a dot product with a convolution kernel. The 1D vector shaped transformed patches are then assembled together to form an output feature map [6].

### 3.1.3 Recurrent neural networks (RNNs)

The recurrent neural networks is also a widely used artificial neural network that performs well for ordinal or temporal problems, such as NLP, speech recognition and image captioning [7]. As is shown in Figure 6, the nodes are connected through a cycle, making the output affect the input to the same nodes. The inputs and outputs in traditional deep neural networks are independent of each other, however, the outputs from RNNs depend on prior elements in the sequence.

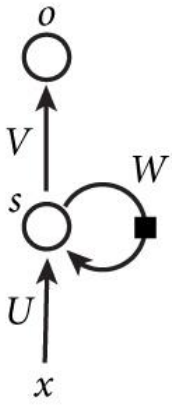



FIG 6 Recurrent neural networks

The difference between RNNs and CNNs is that RNNs has an infinite impulse response while CNNs a finite impulse response. Specifically, in RNNs, the directed cyclic graph will not be unrolled under the infinite impulse response.

### 3.1.4 Long short term memory (LSTM)

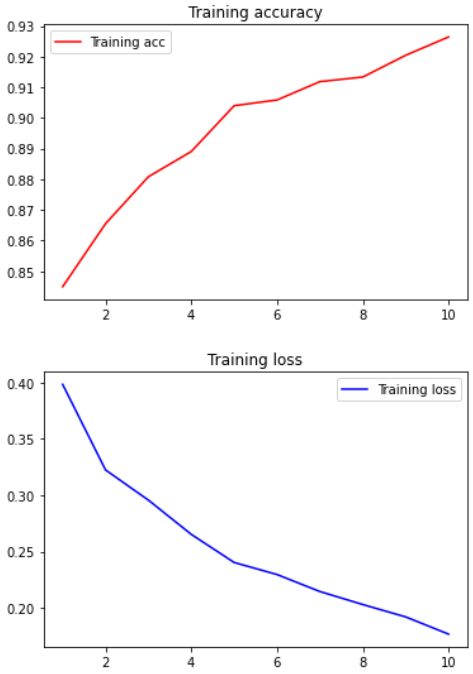
Long short term memory networks is a special kind of RNNs. It was designed to avoid the long term dependency problem [8]. Instead of having a single neural network layer, LSTM has more layers that can decide what information we are going to use or not.

## 4. Result

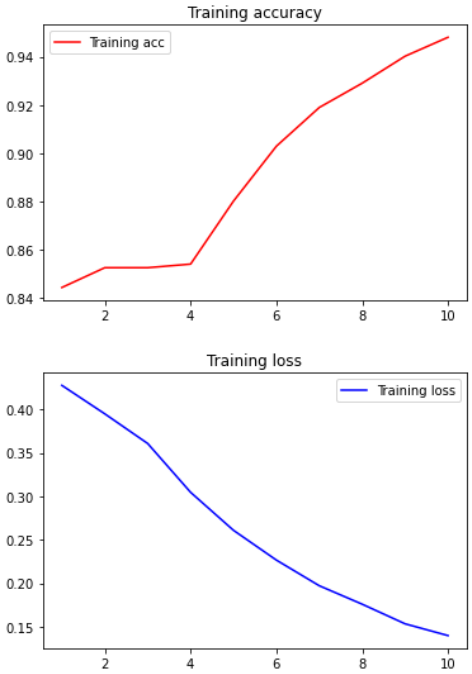
For all notes, they do not represent only one category, but a combination of multiple categories. For example, a note is a quarter note with pitch of a1. There are 3 categories within the note: 'note','quarter' and 'a1'. So, the classifications should be used layer by layer. We use a classification based on whether a note is a rest note and further categories would be easy to extend. The model performance is achieved by looking at the training loss and accuracy as well as the test loss and accuracy.

### 4.1 Note or rest

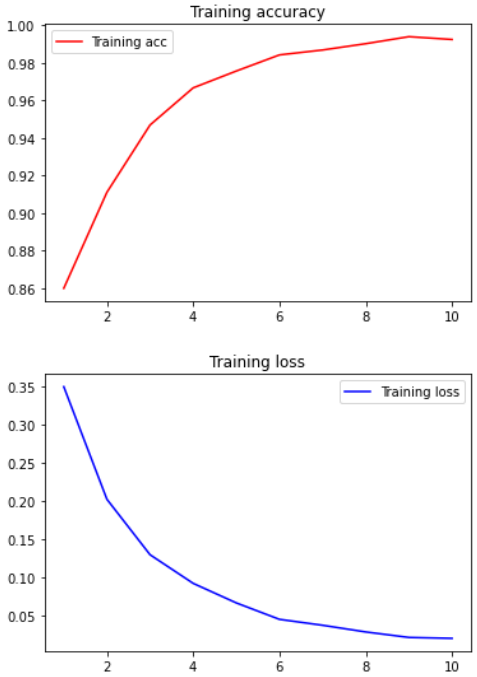
#### 4.1.1 FCNNs




#### 4.1.2 CNNs



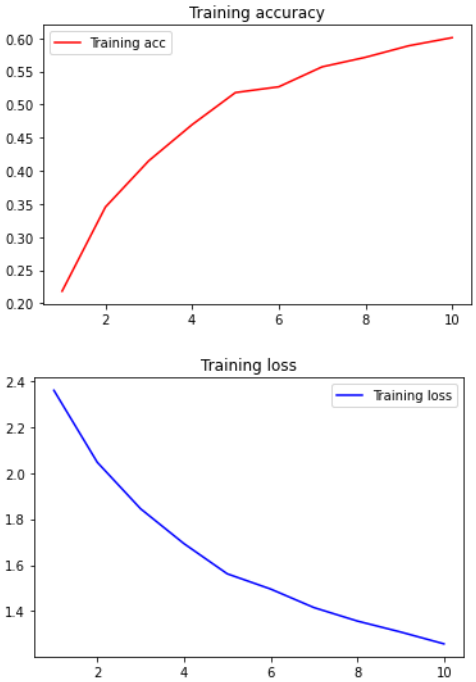

### 4.1.3 RNNs



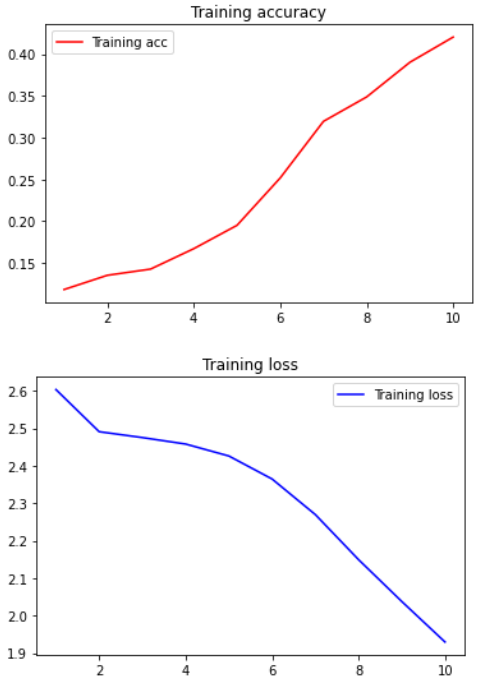

### 4.1.4 LSTM


## 4.2 Note pitch

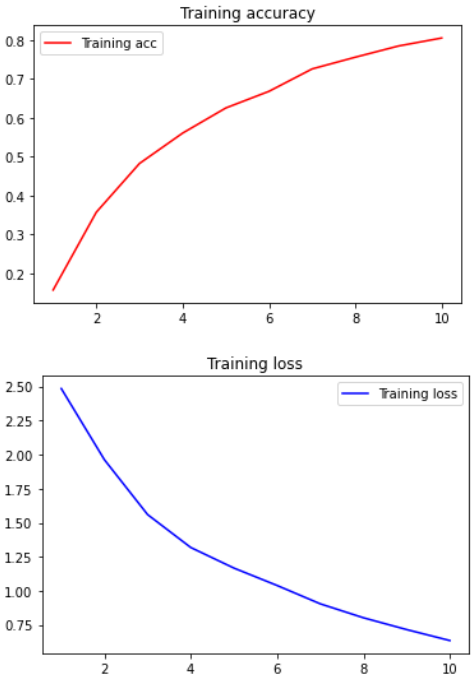
### 4.2.1 FCNNs



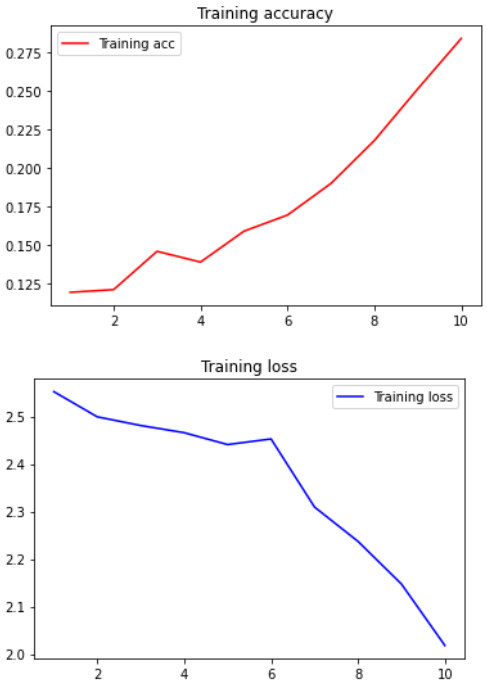

### 4.2.2 CNNs




### 4.2.3 RNNs

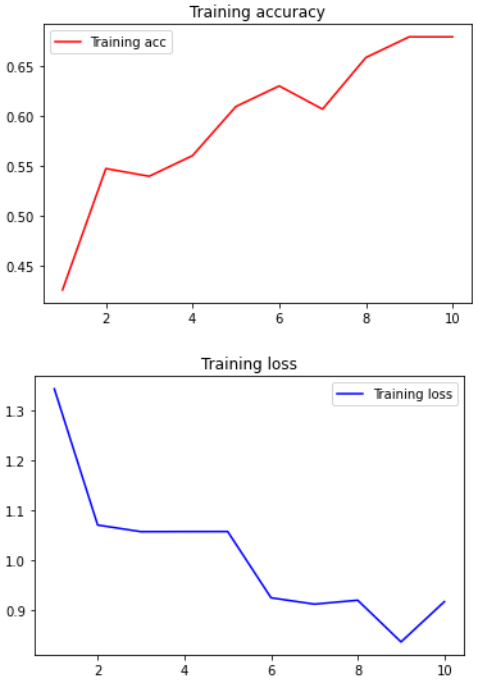



### 4.2.4 LSTM

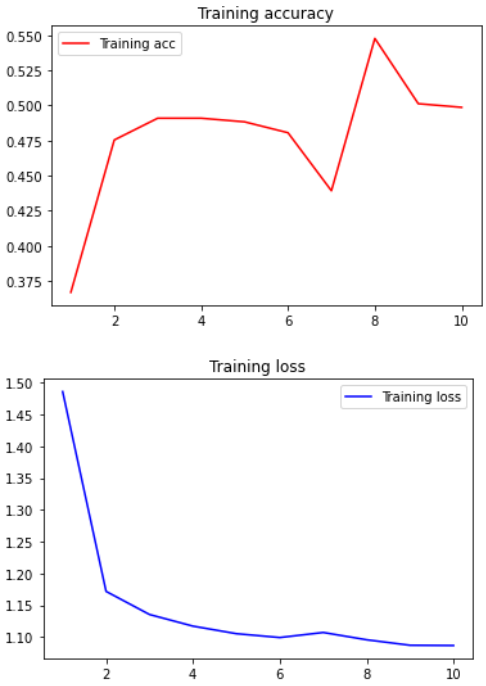



## 4.3 Note duration

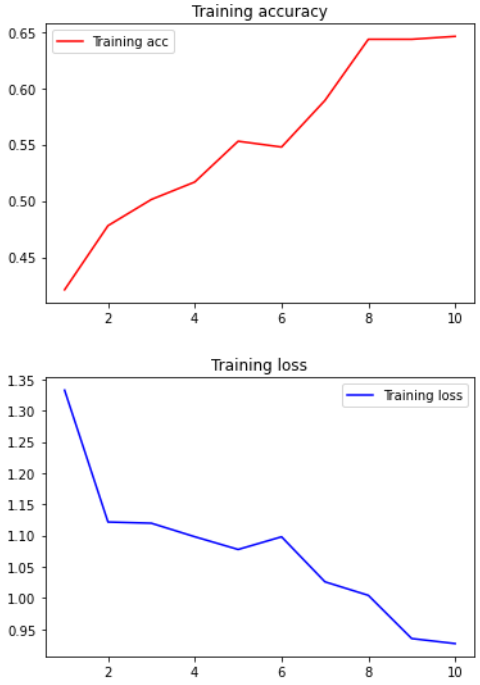
### 4.3.1 FCNNs



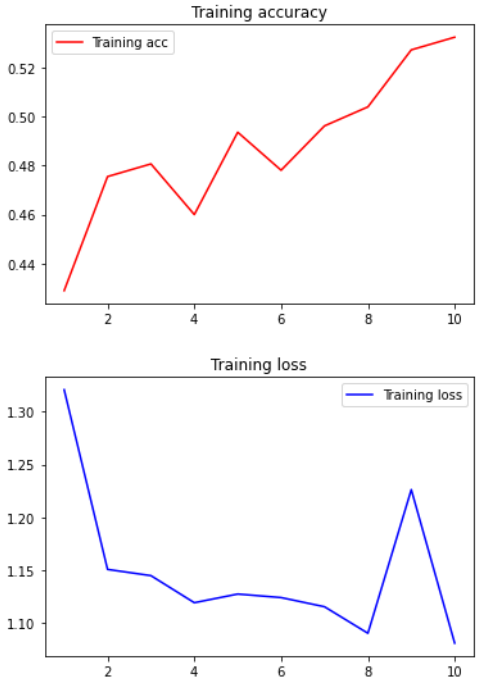

### 4.3.2 CNNs




### 4.3.3 RNNs

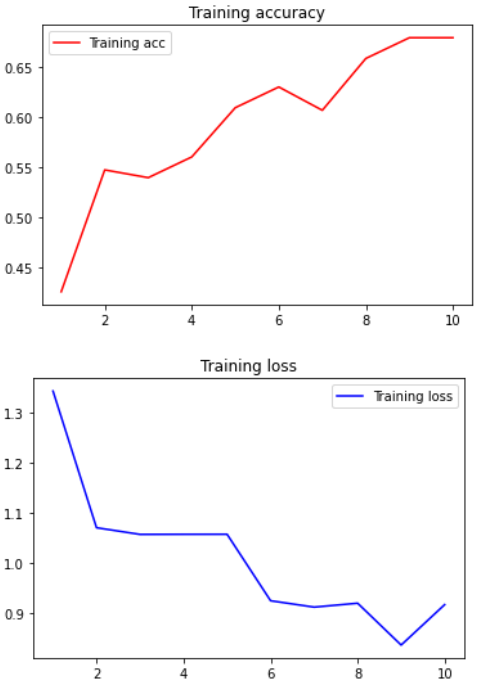



### 4.3.4 LSTM

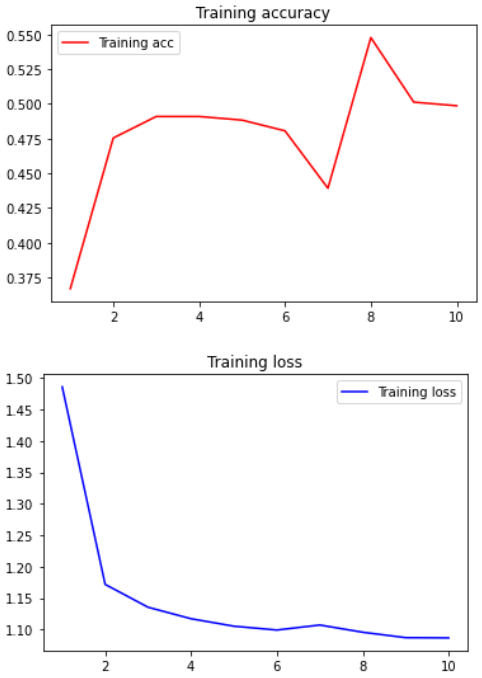



## 4.4 Rest duration

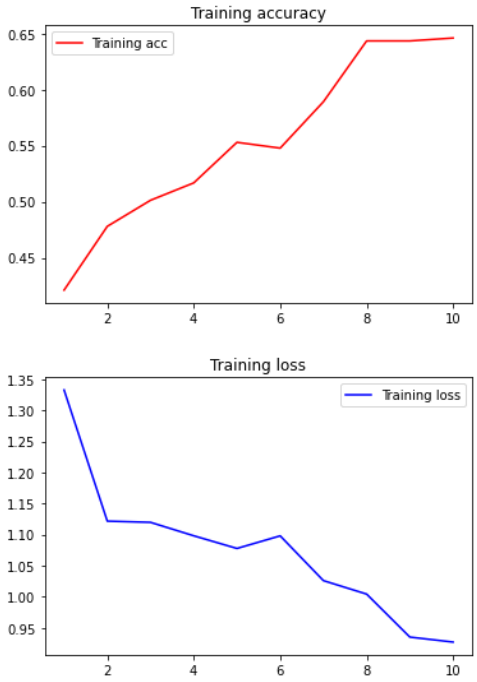
### 4.4.1 FCNNs



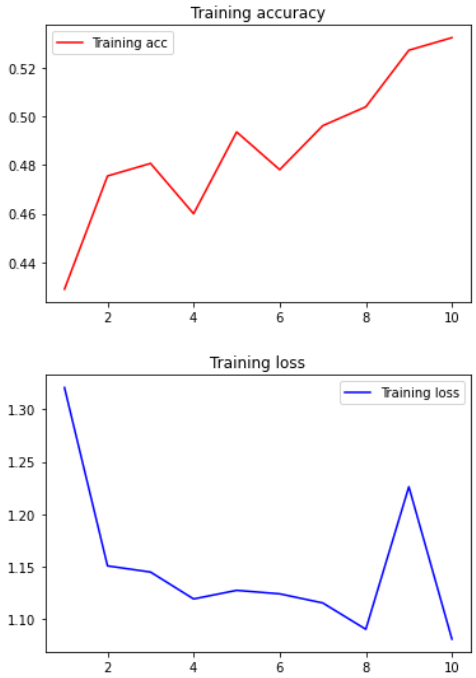

### 4.4.2 CNNs




### 4.4.3 RNNs




### 4.4.4 LSTM




## 4.5 Model performance comparison

| Model | Phase | Note/rest | | Note pitch | | Note duration | | Rest duration | |
|---|---|---|---|---|---|---|---|---|---|
| FCNNS | Training | loss | accuracy | loss | accuracy | loss | accuracy | loss | accuracy |
| | | 0.1768 | 0.9264 | 1.2578 | 0.6010 | 0.9784 | 0.5827 | 0.9172 | 0.6796 |
| | Testing | loss | accuracy | loss | accuracy | loss | accuracy | loss | accuracy |
| | | 0.2081 | 0.9172 | 1.3985 | 0.5403 | 1.0422 | 0.5515 | 1.1239 | 0.5091 |
| CNNs | Training | loss | accuracy | loss | accuracy | loss | accuracy | loss | accuracy |
| | | 0.1399 | 0.9481 | 1.9295 | 0.4203 | 1.1391 | 0.5168 | 1.0868 | 0.4987 |
| | Testing | loss | accuracy | loss | accuracy | loss | accuracy | loss | accuracy |
| | | 0.1701 | 0.9241 | 1.8258 | 0.4811 | 1.1844 | 0.4944 | 1.0312 | 0.4545 |
| **RNNs** | **Training** | **loss** | **accuracy** | **loss** | **accuracy** | **loss** | **accuracy** | **loss** | **accuracy** |
| | | **0.0197** | **0.9925** | **0.6352** | **0.8049** | **0.677** | **0.7399** | **0.9269** | **0.646** |
| | **Testing** | **loss** | **accuracy** | **loss** | **accuracy** | **loss** | **accuracy** | **loss** | **accuracy** |
| | | **0.0407** | **0.9826** | **0.7789** | **0.7604** | **0.7004** | **0.7513** | **1.1479** | **0.5455** |
| LSTM | Training | loss | accuracy | loss | accuracy | loss | accuracy | loss | accuracy |
| | | 115573 | 0.7191 | 2.018 | 0.2842 | 0.9898 | 0.5949 | 1.0812 | 0.5323 |

| | Testing | loss | accuracy | loss | accuracy | loss | accuracy | loss | accuracy |
|---|---|---|---|---|---|---|---|---|---|
| | | 4615 3 | 0.8631 | 1.939 7 | 0.2691 | 1.515 3 | 0.3782 | 1.1 914 | 0.4545 |

## 5. Conclusion

According to the cross-entropy loss and accuracy among 4 deep learning algorithms in this work, the RNNs shows the best classification result for all kinds of musical notations while the least is LSTM. That is because it can model a collection of records so that each pattern can be assumed to be dependent on previous ones. RNNs are even used with convolutional layers to extend the powerful pixel neighborhood.

## References


[1] Attwenger, P. (2015). Recognizing Musical Notation Using Artificial Neural Networks. Bachelor's thesis at the University of Vienna.

[2] Graham. (2010, December 09). Naming musical notes: Music. Retrieved December 22, 2020, from https://www.allthingsgerman.net/blog/music/naming-musical-notes/.

[3] Calvo-Zaragoza, J.; Rizo, D. End-to-End Neural Optical Music Recognition of Monophonic Scores. Appl. Sci. 2018, 8, 606.

[4] Luckner, Marcin & Homenda, Władysław. (2004). Automatic recognition of music notation using neural networks.

[5] Hui H.I Built a Music Sheet Transcriber — Here's How. https://towardsdatascience.com/i-built-a-music-sheet-transcriber-heres-how-74708fe7c04c.

[6] Chollet, F. (2018). Deep learning with Python. Shelter Island, NY: Manning Publications.

[7] Dupond, Samuel (2019). "A thorough review on the current advance of neural network structures". Annual Reviews in Control. 14: 200–230.

[8] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*, 1735-1780.