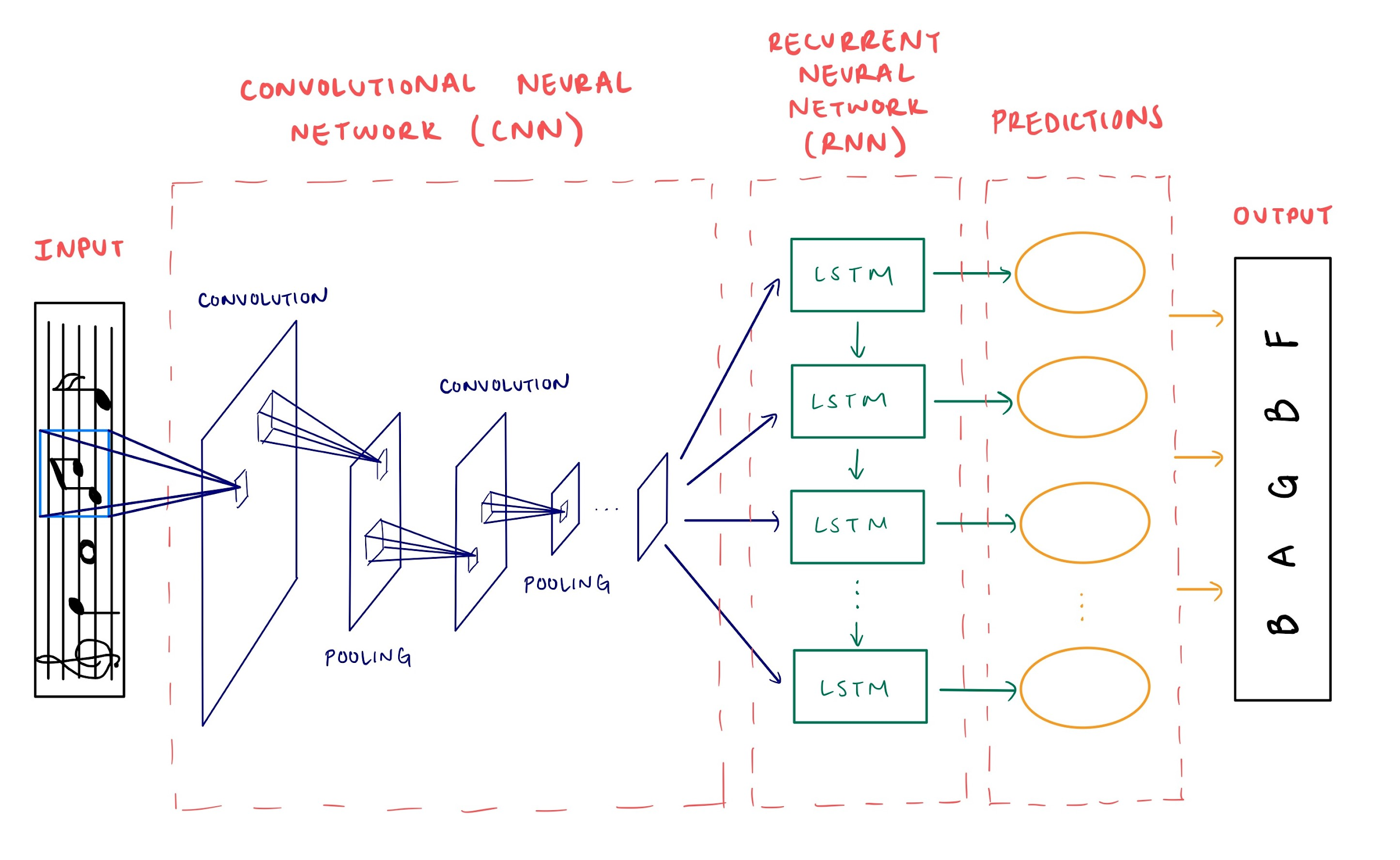
**1. Introduction**

Have you ever wanted to play the piano but didn’t have the time to learn to read musical notes? Reading sheet music can be a major learning obstacle for beginners. We want to create a music sheet transcriber that makes sheet music easier to understand for those who cannot sight read but simply want to play their favourite songs. A successful implementation of our model could also help to solve related problems of digitizing sheet music records, or evaluating music theory homework for students. We aim to use Machine Learning classification algorithms to create a model that takes an image of sheet music and outputs the corresponding notes as alphabets. Machine Learning is an ideal tool for this project because it is a subset of the image classification problem which is a solved problem that has an efficient Machine Learning solution. Hardcoding note recognition is a near impossible task due to the myriad of different styles and formatting for the music sheets and notes.

**2. Illustration / Figure**



**3. Background & Related Work**

Our project lies in the field of Optical Music Recognition (OMR) which aims to computationally read music notation from images. A study published in 2012 by Rebelo et al. [1] describes traditionally used algorithms which are used in a multi-staged approach involving image preprocessing, symbol classification and musical notation reconstruction. Common preliminary steps include staff line removal, locating and isolating individual musical symbols, and finally interpreting the musical semantics from the graphical objects detected as well as their positional information. This approach uses complex code, making it difficult to generalize the models to any piece of sheet music.

A model proposed in a paper by Calvo-Zaragoza et al. tries to solve the OMR problem for printed monophonic(one melody) scores with a holistic end-to-end approach using supervised learning and a Convolutional Recurrent Neural Network (CRNN) [2]. The model takes in a PNG image of an incipit (first few notes in a piece of music) as input to a Convolutional Neural Network (CNN) for image processing and feature extraction. The output channels from the last layer of the CNN are concatenated into a single image and used as input for a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) units. The RNN then analyzes columns of the image to produce the predicted sequence of musical notes as output. The model achieved a sequence error rate of 12.5%, where sequence error refers to the ratio of predicted sequences that had at least one error.

**4. Data Processing**

We will be using the Printed Images of Music Staves(PrIMuS) dataset [3], comprising approximately 80,000 incipits.

Each piece of data has 5 representations:

1. Plaine and Easie code [4] (a standard library used to denote musical symbols with textual symbols)
2. PNG Graphical rendering
3. Music Encoding Initiative (MEI) format
4. Simplified semantic encoding (list of symbols with musical meaning)
5. Agnostic encoding (list of symbols with position given, without musical meaning)

2) will be used as input into the model, and 4) will be used for the ground truth labels.

When we load 2), the pictures will be represented by a height x width tensor (RGB is irrelevant since pictures are in black and white)

**5. Architecture**

We plan to use a CRNN for our project. The CNN will be used for image processing and its output will be fed into a RNN that will handle the sequential aspect of the problem. The CNN will consist of convolutional layers as well as pooling layers to consolidate information. For the RNN, we will likely use LSTM units as the sequence predictions may require learning long-term dependencies. The CRNN will also use Rectified Linear Unit (ReLU) and softmax activation functions as well as stochastic gradient descent (SGD) for the weight training process.

**6. Baseline Model**

The field of OMR has been an important research topic, progressing slowly for nearly two decades due to poor scalability and resource consumption of the manual solutions. It has since been demonstrated that in order to analyze a music score, an image analyzing component and a sequence identifying component both need to be present in the model [2]. As such we have decided to have our baseline model be a simple Convolutional Recurrent Neural Network. The CNN part will have a few convolutional and pooling layers to compare our symbol classification against while the RNN part will be used for sequence prediction and won’t have LSTM units. This baseline model can help quickly understand if there are any problems with our data, or implementation bugs, without adding in a lot of complexity.

**7. Ethical Considerations**

Under Canadian Law, a musical composition is protected by copyright if its author is still living or died less than 50 years ago. Using our model to redistribute the transcribed music online may violate this law. If one obtained the creator’s permissions and returned a portion of royalties [5], one could legally use our model to transcribe a copyrighted music sheet and distribute it online for a fee or profit through advertising on the digital platform. Therefore, it may be important to limit our model to personal use or only transcribe unprotected music sheets. In terms of training the model, it is legally acceptable to use copyrighted music to train a machine learning model [6]. However, there are ethical concerns around obtaining copyrighted music sheets through illegal downloads. Therefore, we should collect our training data through legal means, or use sheet music which creators have made freely available.

**8. Project Plan**

We will meet once a week, every Wednesday 8:30 PM and communicate with each other via Discord and Messenger. We will share our documents and code on Google Drive and Google Colab. To avoid overwriting each other’s code on Google Colab, each member will make a copy of the shared file and work on their own file first, and we will manually merge the local files when the code is completed.

**8.1 Proposal Breakdown**

Below is each member’s assignment of the project proposal.

|  | Tasks | Deadline |
| --- | --- | --- |
| Marcos | Baseline model | Friday, February 12 |
| Mirza | 1 Related Work summary, Illustration | Friday, February 12 |
| Saskia | Data Processing planning, 1 Related Work summary | Friday, February 12 |
| Nancy | Ethical Considerations | Friday, February 12 |
| All | Introduction, Architecture, Project Breakdown, Risk Register | Friday, February 12 |

**8.2 Project Breakdown**

Below is the breakdown of our project. The internal deadlines and the assigned members may be subjected to change.

| Tasks | Explanation | Internal Deadline | Assigned to |
| --- | --- | --- | --- |
| Data Processing | Collect data and labels based on Section 4: Data Processing of the proposal | February 19 | All |
| Load Data | Write function to load the data into the code and transform them to suite our model | February 26 | Saskia |
| CNN Model | Code CNN model | February 26 | Nancy, Saskia |
| RNN Model | Code RNN model | March 5th | Marcos, Mirza |
| Test the code | Combine the CNN and RNN models together and test and refine | March 19th | All |
| Training the model | Train our model and decide hyperparameters based on validation data set  Finish up the project | March 26th | All |

**9. Risk Register**

Three major risks of the project and their solutions are outlined below.

1. Missed internal deadlines: If a member cannot meet internal deadlines, they should let the team know as early as possible so that necessary adjustments can be made i.e, delaying internal deadlines, redistributing tasks, and modifying the project scope and/or model architecture.
2. Don’t understand RNN: Since RNNs are covered in the second half of the course, we may have difficulties understanding the RNN architecture, which may cause us to fall behind and have less time to properly construct the RNN model. We will address this risk by keeping each other accountable to learn about and understand RNNs earlier, so that even if the coding is only done later, we have enough time to receive help from the TAs, and have a solid idea of how our model works.
3. Training takes too long: Since our training data consists of images of lines of sheet music that contain multiple music notes, it may take a long time to train our model. If this situation occurs, and we are absolutely unable to finish training before the final deliverable deadline, we will have to reduce the complexity of our model and/or redesign the architecture. Potential solutions are reducing the number of layers, using only CNN or RNN, and using less training data.

**10. Link to Github or Colab Notebook**

<https://colab.research.google.com/drive/1W3Zuq-EvnlJ_z7Luyd073jVHSUC1Ofce?usp=sharing>

**11. References**

[1] A. Rebelo, I. Fujinaga, F. Paszkiewicz. *et al.* “Optical music recognition: state-of-the-art and open issues” *Int J MultiMed Info Retr,* 1, 2012. [Online]. Available: <https://doi.org/10.1007/s13735-012-0004-6>. [Accessed: 12-Feb-2021]

[2] J. Calvo-Zaragoza and D. Rizo, “End-to-End Neural Optical Music Recognition of Monophonic Scores,” *Applied Sciences*, vol. 8, no. 4, p. 606, Apr. 2018 [Online]. Available: <http://dx.doi.org/10.3390/app8040606>. [Accessed: 12-Feb-2021].

[3] *PrIMuS dataset*, 2018 [Online] Available at: <https://grfia.dlsi.ua.es/primus/> [Accessed 12 February 2021].

[4] IAML. 2021. *Plaine & Easie Code*. [Online] Available at: <https://www.iaml.info/plaine-easie-code>[Accessed 12 February 2021].

[5] Government of Canada, “What is copyright?,” *Canadian Intellectual Property Office*, 07-Sep-2016. [Online]. Available: <https://www.ic.gc.ca/eic/site/cipointernet-internetopic.nsf/eng/wr03719.html>. [Accessed: 12-Feb-2021].

[6] M. Stewart, “The Most Important Supreme Court Decision For Data Science and Machine Learning,” *Medium*, 29-Jul-2020. [Online]. Available: <https://towardsdatascience.com/the-most-important-supreme-court-decision-for-data-science-and-machine-learning-44cfc1c1bcaf>. [Accessed: 12-Feb-2021].