Jeevan Thomas Koshy

email: jeevan.thomaskoshy@gmail.com
Phone no: +91-9790469245
Github: https://github.com/jeevantk
Website: https://jeevantk.github.io/
TimeZone: +5:30
Blog: jeevantk.blogspot.in

Semi Autonomous Optical Music Recognition

Synopsis: To convert an image of a musical document to MusicXML format so that it can be used to generate music. Optical Music Resolution (OMR) is equivalent to Optical Character Recognition for Images. Researchers have mentioned OMR has a tougher problem since the variation of a symbol in vertical direction can also imply a new character making it equivalent to a two dimensional equivalent of OCR.

Importance of this Project: There are millions of Sheet Music Available around the world in paper. There are softwares like Musescore which enable you to convert Sheet Music manually into MusicXML format via their User Interface. A large majority of Musescore users use the software to generate music from sheet music. This helps students in music to learn faster as compared to learning directly from a notation. Myself personally being a music lover and a pianist have used Musescore numerous times. It typically takes at least half an hour for converting a page of sheet music into MusicXML format if you are only using your Computer. The solution that I thought for the problem was to make a Optical Music Recognition Software with the help of the Computer VIsion technologies.I believe that the impact which can be made by the project is huge.A good Optical Music Recognition system can be integrated to Musescore which can be useful to thousands if not millions of Musical Lovers.

Deliverables: Since the difficulty of this project is as comparable to that of making a OCR system if not more, I'm proposing the part of the project as my GSOC task which adds the largest amount of value in terms of time of the users using the software. This part won't require large amount of labelled training data which is not currently available. I'm dividing the problem into two subproblems:

1.Finding the coordinates of noteheads relatively to the position of the staves: The coordinates of the noteheads gives us the actual musical note that has to be played at an instant of time. Even though this may sound easy there are a lot of complications involved. The images of the documents need not be always aligned making our job difficult. n order to achieve this we need to find the position of the Clefs and the Staves before hand.

2.Classification of the detected notehead: This is a much more difficult problem than detecting the notehead .The actual shape of the note dictates the number of beats it should be played. There can a large number of variations possible for a note indicating a particular duration depending on the position , shape of the nearby notation and so on. This make classifying a given notation the task of a neural network.But the lack of sufficient amount of training data has hindered the development in this direction.

Due to this challenging nature of this problem I believe that I might only be able to complete the first part of the problem in the duration of GSOC 2017. Hence what I propose is a useful outcome which can be achieved by completing the first part of the task. If you are using a mouse and a Keyboard to enter a musical notation into a software like that of Musescore the majority of time is spent in positioning the note at the exact point. If we can detect the exact position of the notehead (mentioned as the first task) this can be done automatically and the user will only have to select the appropriate note type for every note. If there is a prediction error in the coordinate position of the note by a small amount it can be adjusted with the help of some appropriate keys(eg: arrow keys).

Hence the proposed deliverables as a part of GSOC 2017 are:

1.Accurate detection of notehead co-ordinates for printed sheet music
2.A separate GUI using which a user can classify the notehead and hence create the notehead.
3.Integration with Musescore (In case of sufficient time).
4.Automatic Classification of the detected notehead(In case of sufficient time).

Project Description

I have not found a reliable enough paper to follow for optical music recognition.Also since my goal is to just find the position of the notehead and not to classify it I believe it can be done in my own way. I am open to experimentations and have done a bit of experimentation on this topic last year. Here are the finer Implementation details that I wish to add to my implementation.

Classifying an image as a sheet music or not : This classification is indented so as to prevent processing any images which may not contain a bit of musical information as such. (For instance an image of a cat). A binary classification Convolutional neural network can be trained in order to achieve this purpose. I have done similar work during one of my internships where I classified Documents vs Non-Documents. An popular neural network pre trained model (say for classification in Imagenet) can be taken and can be tweaked to achieve this classification.The only issue being to find a lot of training data in this format. Also since this is only required to make the system work better for an external testing the implementation concerning this would be done at last as this stage won't affect the performance of the remaining part.

Identification of Text in the Document: Tesseract can be used to find out the position and exact coordinates of the present text.This text has to be removed as it can affect with the further processing. OpenCV has a tesseract class which helps to directly use tesseract in this case. I

had made a sample implementation using tesseract to remove words in a sheet music last year and it had worked beautifully. Removal of text reduces the chances of errors in the future methods that I am going to propose. Also the orientation of text typically gives the orientation of the sheet music and a reverse transformation can be done to orient the image in proper direction.

Detection of Staves in the Musical Sheet: A Hough line transformation can be done to detect all the horizontal lines in the image. We can preferably select only the parallel lines compared to lines in other direction. If alignment was not done based on the detect text using tesseract the alignment of the image can be done by performing a rotation dictated by the maximum number of of lines orientation (would be typically be dictated by that of a Stave).Hough line transform is a beautiful linear time algorithm (linear on the number of edge points/total number of pixels) and can find the best line even if there are points which do not lie on a line. Texts if present can be detected as a false line (based on my experimentations). That was one of the main reasons why I insisted on removing text first.

Detection of Clefs, Time Signature and Scales:

There exists a lot of rules in sheet music which makes the Job of OMR easier compared to that of OCR is certain aspects. For instance every Stave will be started by either a bass clef or a Treble clef. Hence this can also be used an additional condition of accurate detection of staves. I had made a test implementation using SVM classifier using HoG as the features. It showed decent results even with a very small training data I personally created.Hough line transform and Clef Detection can be used together in order to accurately identify staves and clefs(use AND condition) .

It does not take much time to manually input the Time Signature and Scales and for now this shall be done by the user using the created GUI. On a later stage this can be further automated using the CNN that is made to do the classification part.

Detection of Barlines:

Bar Lines can are the vertical lines that exist frequently throughout a stave.The region enclosed by two bar lines defines a bar. The total number of beats inside each bar would be a constant and would be equal to the time signature of the song.This gives us an additional constraint on recognizing each notes which can be exploited later. Barlines are basically vertical lines which can be detected using using a Hough line transform. A Sobel-X filter can be used to amplify the edges along the vertical direction making it easier to detect the vertical lines.

Even though it sounds easy a lot of notes has a vertical tail associated with it which can cause a real issue in differentiating between a bar line and a note. This can be left to either the discretion of the user or can be distinguished based on the detection used for notehead.

Removal of Staves:

Staves can cause an issue in further processing and is better to be removed (assuming we store the coordinates corresponding to the staff lines). The following package gives a decent implementation of staff line removal and I indent to use it for removing the staff

lines.Alternatively removing all the elements of Hough line will also give an approximately same result.

Accurate detection of Notehead Coordinates:

This is the most important step in the entire implementation. The amount of time that can be saved for the user will directly depend on how accurately we can detect the coordinate correctly and hence identify the pitch correctly. Detecting the horizontal position is relatively easier compared to that of detecting the vertical position. The major characters left after removing staff lines will denote individual notes (or localised symbols) and can be easily be identified using come clustering algorithm (eg: K Means).For identifying the vertical position I'm proposing an approach which is a combination of a number of methods. Each method will vote for a particular coordinate in Y-direction. The coordinate getting the majority of the votes will be selected as the Y-coordinate for the notehead.

Taking closed contours, filling them with floodfill and using distance transform can give high intensity values for noteheads.Hough circles can also give quite good estimation of noteheads. A template matching can also be used for detecting notehead position.Even though these methods might not be that good in detecting the note heads individually when combined together these methods can give a quite good accuracy(The principle of Cascading of weak Classifier to get a strong classifier.)

Creation of Graphical User Interface:

I intend to use Qt for making the GUI for the created software. Musescore also uses Qt for their development and hence it would be easier to integrate with their software later on if I use Qt. I intend to make a relatively simple Interface or collaborate with Musescore to use their Interface.Whenever a Sheet Music is inserted it will create an a blank Sheet Music file with the exact number of staves as in the original file. The user would have to select the Time Signature and the scale for each staves. For each of the detected notes the user will have to choose between a number of options for the correct beat . This saves a lot of time in comparison to the normal version of Musescore where a person has to select the beat and click at the exact coordinate to get the correct pitch.The user would have the power to override any nothead predictions made by the system.

Another added advantage with this method lies in the fact that we would be able to generate sufficient labelled data required for fully autonomous classification of the note head by cropping out and saving a region of the image where the notehead was detected.

Integration with Musescore:

Musescore would directly be able to input MusicXML files into their system so as to generate beautiful music.I am familiar with Musescore code and with the support of the Musescore community integrating this would be straightforward. Musescore has a light OMR framework already written where a OMR system can be plugged in .

Automatic Classification of detected noteheads:

I think a lot of data is required for this process which can be generated by the above proposed method. If time permits I will try out conventional Computer Vision techniques with the limited amount of available data. For printed Sheet Music we might be able to get decent results with such a system.

Schedule

Our Institute has a three month summer vacation almost aligning itself with the GSOC period and hence I am totally free during the vacation.Our summer vacation commences on May 5th and ends on 31st July.GSOC schedule is from May 30th - August 21st. Even though I would be able to spend much time after the classes start I'm willing to start as soon as my summer vacation starts and finish the project before my institute reopens.

Timeline	Task	GSOC Timeline
May 5th-May 30th	● Start Coding early since my classes will commence by July 31st. ● Update the code that I have already written last year including the parts on Text removal, stave detection, clef identification and detection of barlines. ● Add proper Documentation to the written code. ● Contact various organizations including Musescore for obtaining required datasets for the latter part	Community Bonding
June 1st-June 30th	● Implement bar line detection , removal of staves and accurate detection of noteheads.	Coding Phase before first evaluation
July 1th-July 24	● Implement the User Interface and experiment on the software to fix bugs. ● Perform unit tests .	Coding Phase before Second Evaluations
July 28th -August 21th	● Integration with Musescore ● Automatic Classification of detected Noteheads if sufficient data was obtained. ● Buffer time for accounting for delay due to unforeseen circumstances.	Last coding Phase

Related Work:

While there exists quite good commercial implementations of OMR the only popular open source implementation is that of Audiveris . Audiveris was developed long before and

Computer Vision Techniques have changed a lot since then.The development of Audiveris had been stagnant for a long time and I was never satisfied with the results generated by the software. I am also not aware of a single open source software which does semi-automated notation entry.

I began working on this project last year and had proposed it directly for Musescore last year for GSOC 2016. Here is the link to my previous proposal and the accompanying code the wrote to ascertain the validity of certain components. I was unable to get into GSOC 2016 and since I got another internship after that I had to stop working on the project.During my last proposal I thought I would be able to complete the entire project and hence proposed the whole framework. Currently I believe that the entire project was a bit too much for a summer and have hence proposed a lighter version of the project. Most of the implementation details mentioned was self proposed and was not taken from any paper. However I have referred to these papers as well [1] ,[2],[3]

Biographical Information:

I am a third year undergraduate student in the Department of Mechanical Engineering at IIT Madras. I developed an interest in Computer Vision due to the amazing Computer Vision Group here at IIT Madras.I was a part of the Group from my first year and is currently the Club Strategist. I have worked on various projects in Computer Vision and have Mentored a lot more projects done by Juniors.I have worked on extensively on Hand Gesture Recognition, Optical Music Recognition , Image Stitching ,3D reconstruction and Deep Learning.

I choose robotics as my Minor Stream and hence have done some courses related to Computer Vision and Machine learning. My relevant coursework includes Machine learning(from Coursera), Artificial Intelligence (CS6380) ,Computer Vision (CS6350) , Reinforcement Learning (CS6700) and Data Structures and Algorithms (EE4371). All courses apart from the Machine learning course were done from the IIT Madras.

I have interned with two startups before . In my second year I interned with a startup HyperVerge . This startup was founded by our former students of IIT Madras at the Computer Vision Group. Most of my time there was spend in learning and I trained a Convolutional Neural Network for classifying Documents vs Non - Documents. Last summer vacation I interned with a startup Detect Technologies. I was the lead Computer Vision Developer in this startup and I worked on processing drone Videos. I developed a Image stitching algorithm to stitch drone videos. Most of the ideas were taken from this paper by David Lowe. I was able to extend the stitching which was limited to rotational movements of the camera to translational motion as well (for objects at fixed depth).An output of this stitching is shown in my github page. I further worked on these problems and was able to come up with a novel way of 3D reconstruction . I was given a job offer in the startup for my contributions.

I would be available for full time for this summer vacation . Our Institute has a 3 month vacation which almost aligns with the GSOC schedule.My classes for the next semester will start on 31st of July. I won't be having a lot of courses next semester since I am going to be in my final year.Nevertheless I plan to complete the most of the work before that and hence I don't

think that would be an issue. I am a music lover and plays piano in a nearby church which would require around 3 -4 hours of my time on Sundays. Apart from that I don't have any other commitments for this summer vacation and would be able to do justice to the project if I get selected.I am quite flexible with my sleep timings and would be able to attend all the IRC meetings .

I am really passionate about Optical Music Recognition and I will do my best on the project if I get selected. OMR can the potential to affect a lot of people and I am really enthusiastic about the project. I have been learning piano for the past 6 years and am comfortable with all the knowledge required on behalf of reading a Sheet Music. I also wish to undertake research in this field for my higher studies.

I am applying for two projects this time in GSOC both under PSU. The other project is on Face Alignment. I like both of these project topics but would give more weightage to the Optical Music Recognition Project since it would have a larger impact if completed.Thanks for taking the time to read my proposal. Hoping to work with Portland State University for GSOC 2017.

Extra Information:

- **Resume:** link
- **University Information:**
  - **University Name:** Indian Institute of Technology (IITM)
  - **Major:** Mechanical Engineering
  - **Minor:** Robotics
  - **Current Year:** Junior Year
  - **Expected Graduation Date:** June ,2018
  - **Degree:** Bachelor of Technology
- **Other Contact information:**
  - **Alternative Email:** me14b030@smail.iitm.ac.in
  - **Alternative Phone No:** +91-8547368410
- **Address:** Room No: 363,
  Alakananda Hostel,
  IIT Madras,
  Chennai,
  India.