2015 International Conference on Computer Communication and Informatics (*ICCCI* -2015), Jan. 08 – 10, 2015, Coimbatore, INDIA

Optical Music Notes Recognition for Printed Piano Music Score Sheet

Prof. Apurva A Mehta
Asst. Prof., Dept. of Computer Engineering
Dharmsinh Desai University
Nadiad, India
mapurva@ymail.com

Prof. Malay S Bhatt
Assoc. Prof., Dept. of Computer Engineering
Dharmsinh Desai University
Nadiad, India
malaybhatt202@yahoo.com

Abstract— **Entertainment, Therapy and Education are the fields where music is always found in couple with homo-sapiens. Music is presented in various formats to us like aural, visual and one more - written form of music that is known very less to us. In a way music dominates our life. System discussed in this paper inputs music score written for piano music using modern staff notations as image. Segmentation is carried out using hierarchical decomposition using thresholding along with stave lines of score sheet. Segmented symbols are recognized through an established artificial neural network based on boosting approach. Recognized symbols are represented in an admissible way. System is capable enough of addressing very complex cases and validation is done over 53 songs available at various global music scores resources. Segmentation algorithms achieve accuracy of 99.12% and segmented symbols are recognized with prompt accuracy of 92.38% through the help of PCA and AdaBoost.**

Keywords- grand stave; hierarchical decomposition; PCA using SVD; AdaBoost; measures; modern staff notation; piano score; staff system

I. INTRODUCTION

Optical Music Recognition (OMR) is a process that recognizes music from any form of score sheet and makes sheet readable and editable for computer [18].OMR is somewhat related to Optical Character Recognition (OCR) and known as Music OCR[9]. OMR is much more complex compared to its counterpart OCR due to OMR's two dimensional structural representation as shown in Figure. 1. Here in OMR, horizontal scale provides the order in which music notes are arranged and vertical scale identifies actual frequency to play. In music sense horizontal direction gives the lyrics while vertical direction gives the pitch. Symbols can be found outside of staff boundary at or around ledger lines as well [18].

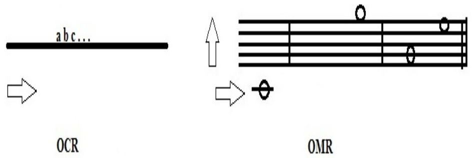


Figure. 1. OCR and OMR representation

OMR is a field principally categorized in handwritten score OMR and printed score OMR. For both category evolution is tremendous like *Ancient Greece*, *Ethiopia*, *Persian and Arab worlds*, *Modern staff notations*, *Early Europe*, etc… Evolution of various forms is based on time, place and parameters like culture. These forms have somewhat variations in symbols representation for different set of musical instrument available.

OMR is generally performed in four stages. Input image is preprocessed in the first phase. Next phase handles segmentation details of OMR sheet. Segmented symbols are recognized during third phase and in the last phase based on spatial relationship exist between symbols and then recognized music notes are represented in a file [2].

Input for any OMR system is an image representing score sheet. Un-tilting an input image is essential only if it is taken/ captured using media like mobile phone [3] or physical copy is scanned through scanner. Hough transform [4], connected component analysis [6] are feasible options to rotate the image to proper dimension. B/W music score sheet is sufficient for OMR so binarization can be achieved through Otsu's method [3][10], iterative method of Ridler and Calvard [8] and local Niblack's [10]. Preprocessing of music score sheet using 'open' morphological operation removes stave lines [3][4][5][8][10]. Once stave lines are removed, music score sheet is left with only symbols of interest. So, it will make the symbols segmentation, recognition and representation process much smoother. This will corrupt symbols resting on stave lines. Once stave lines are removed and note is recognized, it is of much interest to determine its pitch. Pitch of a particular note can be determined by observing location of a note relative to its position with respect to stave line or stave space. This requires knowledge of stave lines so, [1][6][7][9][11][12][13] prefers to carry out segmentation process without removing stave lines.

Template matching [4] and hierarchical decomposition and/or mathematical morphology with connected component analysis [12] and projections [5][1] techniques are widely used techniques for sheet segmentation. Segmentation work is most important part of entire OMR process, as we can recognize a note if and only if it is segmented correctly.

Technique used during segmentation can be used in recognition phase. Template matching and Mathematical Morphological [3][4][12] operations like opening, closing, dilation, erosion are such techniques. Other methods based on classification trees [11], fuzzy [13] and graphs [7] have been proposed. Use of clustering with kNN classifier is also a popular method for note identification. Artificial neural network [1][6][8][9] can also be considered as a good choice for recognition. Networks such as DBNN, ADAM, Probabilistic neural network and Feed forward neural network are used in various OMR systems. Tremendous piece of work is done by creating a grammar (i.e. MusBNF) that is capable of handling most of the crucial case of music notes in well structured manner[10].

Representation phase stores the segmented symbols into data structure with its two dimensional property value, as the same symbol with different pitch(position) leaves different interpretation. Several researchers have solved this problem by representing musical structure as grammar. Other techniques to construct the musical notation are based on fusion of musical rules and heuristics. Few researchers have tried to use abductive constraint logic programming and sorted lists that connect all inter related symbols, in the past. Mainly this is a post processing stage. During the post processing stage the pitch is determined by finding out the note head location with respect to staff line position [6]. Graph based approach [7] is also suitable for representation. Rules based approach [9] can suffice for identifying second dimension of scores recognized using previous stage.

II. HIERARCHICHAL DECOMPOSITION

Out of all available segmentation techniques hierarchical decomposition [2][7][9] is one that appears dominant for OMR segmentation. Hierarchical decomposition is about factoring a complex system into logical parts prosperous to understand, explore and reinvent. As in OMR *pages* of scores divide into *grand stave*. Each *grand stave* separates itself based on *measures* boundary. Each *measure* is composed of two stave: *Upper staff* segment and *lower staff* segment. Upper staff segment constitutes piano keys for right hand whereas lower staff segment constitutes piano keys for left hand. Both staff segments are made of various key symbols representing piano music in standard format. Individual staff segments are divided into individual symbol. Thus hierarchical decomposition shown in Figure. 2 is suitable for optical music score segmentation and found in OMR structure itself.

III. PROPOSED WORK

A Input and preprocessing

Initial input to the system is an RGB image. However the color detail is never of interest to OMR systems. In effect system can optimize the OMR process by reducing color detail as much as possible. System converts given an RGB input image into grayscale image. Resultant grayscale image is get converted into binary image, then we complement given binary image. In (1) I is the binary image and X is complemented binary image.

$$X = 255 - I \tag{1}$$

A. Segmentation

Mathematical analysis of piano score sheet helps in determining threshold value while performing hierarchical decomposition. Grand stave separation is the first step in segmenting score sheet. Take run length count for each column row wise (i.e. vertical projection profile) of a complemented binary image as shown in Figure. 3. Plot is rotated by (-90) degree. X-axis of plot represents row indices (i.e. span of sheet) and Y-axis shows the run length count of column for each row. Maximum count value is known as *gMax*. Out of all such profiles extract horizontal segments which are composed of pixels having continuous positive non-zero intensity in vertical direction. Here …01100110011001100… such pattern helps in extracting potential grand stave from score. Select vertical profile satisfying threshold *T1*.

$$T1 = 0.95 \times gMax \tag{2}$$

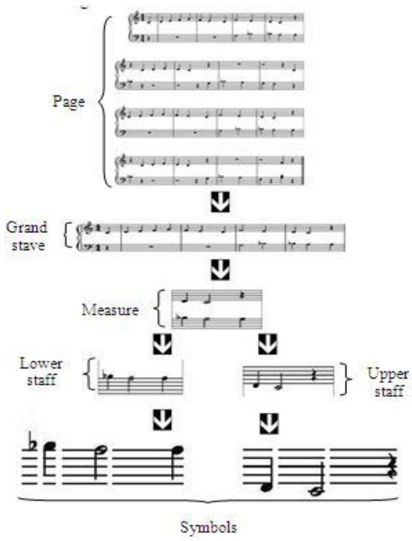


Figure. 2. Hierarchical decomposition

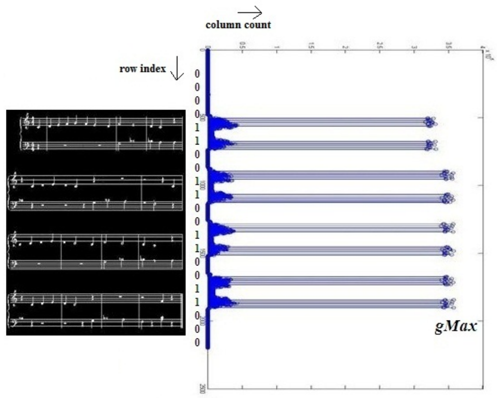
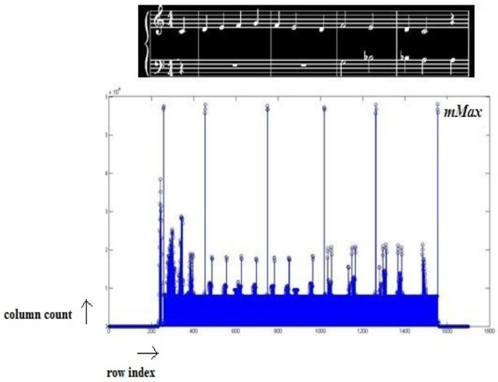


Figure. 3. Grand stave segmentation

It is essential to calculate vertical span of extracted projections profiles because sometimes song title or author credential appears at the top or bottom part of sheet that should not be identified as grand stave so if vertical span is regular/ standard then and only then extracted segment is accepted as grand stave.

After grand stave are separated then it is essential to separate out measures that suggests limits on time scale for the notes to be played. Measure's time limit walks based on time signature provided in score sheet.



Again here we take run length count for each column row wise (i.e. vertical projection profile) of a given grand stave as shown in Figure. 4. X-axis of plot represents row indices (i.e. span) of grand stave and Y-axis shows the run length count of column for each row. Take the highest run length say it *mMax*. Find other such value based on a set threshold T2 from entire vertical projection. Consider the fact that threshold value should be selected in such a way that it removes the symbol specifying grand stave (i.e. '{') from the region of interest.

$$T2 = 0.95 \times mMax \tag{3}$$

Selected vertical projection represents each measure available within grand stave. Separate each measure based on their retrieved position. Separation of measure is also important as this is piano score sheet we are segmenting, we need to play notes of both hands in conjunction without losing their spatial relation.

Then, measure holds staff system. Staff system extraction process is shown in Figure. 5. Upper half represents upper staff system and lower half represents lower staff system. Identify start point s_0 and end point s_1 of a given measure segment in Y direction. Identify mean m_y of a given measure segment in vertical direction using (4). Upper staff segment can be identified by dividing measure segment from start to mean value as in (5) and lower staff segment can be identified using mean+1 value to end s_1 as in (6).

$$\text{Mean } m_y = \frac{l}{2} \tag{4}$$

$$\text{Upper staff segment} = s_0 \text{ to } m_y \tag{5}$$

$$\text{Lower staff segment} = m_y + 1 \text{ to } s_1 \tag{6}$$

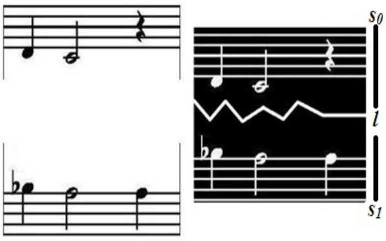


Figure. 5. Staff system segmentation

Last stage while segmenting music score sheet is to extract all individual symbols from both staff segments. System performs segmentation along with stave lines as shown in Figure. 6. It is advisable to ignore stave lines while extracting symbols to make segmentation algorithms more amiable.

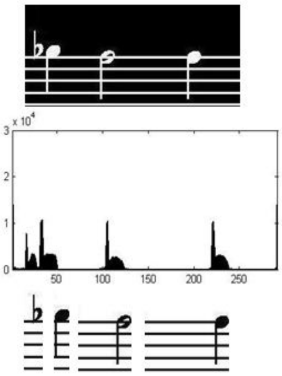


Figure. 6. Symbol segmentation

Identify staff line coordinate for a given staff segment by selecting proper threshold value T3 for identification. T3 is computed using following equation after identifying width of grand stave, *gWidth.*

$$T3 = 255 \times gWidth \tag{7}$$

Complemented binary image is represented in an 8-bit image plane suggests to take intensity value 255 in (7). Check the validity of staff line coordinates by few confidence parameters. Usually staff line count is five with an exception when ledger line(s) is (are) also part of staff segment. Sometimes stems of a note lies atop of staff line or between two staff lines, this may lead to wrong identification of staff line. Perform sum of run length for all pixels other than staff line coordinates. Based on threshold T4 identify all regions of interest. T4 is selected based on the knowledge of the smallest individual symbol available in entire symbol set. Articulation point is the smallest available symbols in modern staff notations.

$$T4 = 255 \times 2 \tag{8}$$

Minimize the area of interest for all individual symbols. Extract symbols based on their coordinates. Minimization of area leads to maximization of area between symbols next to each other.

Segmentation algorithm works neatly and it handles complex cases. Still it is no complete. Segmented symbol is surrounded by quite a big background surface (i.e. Stave lines). That is why so many authors have eliminated stave lines from score sheet while preprocessing an image. The dominating area known as region of interest (ROI) is very thin. So, to make sure that feature extraction algorithm won't extract wrong feature from image sample we need to high light region of interest in sample image. As better feature extraction helps recognizer in recognizing symbol promptly. So, it is better and advantageous to crop segmented symbol. Following Figure. 7 shows the original symbols segmented.

Figure. 8 shows the symbols after applying following technique to crop the symbols

Read the originally segmented symbols and measure the width of each such symbol. Find stave lines' location. Based on location, ignore stave lines. Return sum of image's cells pixel value column wise. Find frequently appearing value in sum arrays, this would represent background. Search for the region in image where sum value is higher than frequently appearing value. Measure start and end point of object boundary in horizontal direction. Extract object from original image based on start and end point retrieved. Such extracted objects are the cropped symbols.

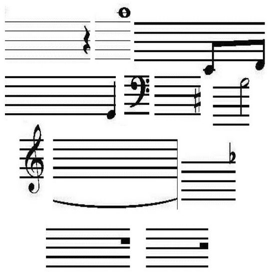


Figure. 7. Original Segmented Symbols

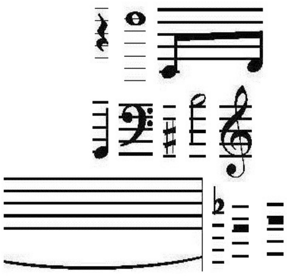


Figure. 8. Cropped Segmented Symbols

B. Recognition

Next step in OMR process after symbol segmentation is to recognize the segmented symbols. To recognize symbol system follow the approach of artificial neural network. This approach requires training an artificial neural network, which can recognize given symbol based on its learning capability. For training an ANN, input images with labels are used. All cropped images are of different dimensions so it's important to rescale them in unique dimension. This is required as if dimensions are not unique then features set may have different number of features and that is not good while extracting features using PCA with SVD. $m \times n$ sized image is converted into 15×15 size, which in turn gets converted as row vector into 1×225 sized image. Then, resized row image is passed on to ANN for training and later for testing purpose. In the row image each row indicates the sample and all columns values indicate features of sample. Here, we have some high dimensional images as data. Data are represented in form of vectors. Vectors represent various facts of data. These facts allow us in exploring data more vividly. We have images with 1 X 225 dimensions. That means each image is represented by 225 features. Data change in structured formation and pattern is hidden within these 225 features. It is possible that to describe important facts about data we may not require all 225 features for each image sample, due to presence of noise or redundancy in feature set values. If we want to find patterns along with compressing data like representing data using only few of the important feature components, we can follow technique like PCA. Purposeful change in features of data is conveyed by such technique. Even it helps in representing structure in minimal way. Use SVD to fetch us Eigen values and Eigen vectors that we need for PCA. We select only dominant eigenvectors. Here, specifically dominant means eigenvectors representing 95% of original data. Leverage scores are represented using this dominant principal component. Thus, that represents original data in new space. These features along with its label pass on to ANN to train a network for future pattern recognition.

We use ensemble method to train and recognize feature set. Ensemble method takes a help from other machine learning algorithm as a component. It basically combines many predictors and it believes in the idea that we can do a better prediction/ classification by following notion of many predictors. Our idea is to use boosting method with single predictor for many numbers of iterations and during this phase train network with specified input data. In this ensemble technique learners learn sequentially. Initial learners try to fit the data and later model analyze the data for errors and try to fit those errors in data. All models have weight and later overall weighted average is used to predict any behavior. In our implementation we use classification tree as learners in our adaptive boosting procedure. Classification tree creates single tree template for each weak learner. Performance of ensemble learners depends not only on parameters of ensemble algorithm but it performs well or poor based on parameters of learner. Thus, we use the MATLAB's facility of ClassificationTree Template that creates a suitable tree template for our learner to pass on. MinLeaf is one such attribute that makes sure that each leaf node has at least MinLeaf number of observation per tree leaf. In the case of boosting procedure default value is half the number of training samples, in our case it means default value is 1300. This value does not help us in achieving good performance. It's intuitive to know that if we keep very small value we shall definitely achieve good result. Against the good performance we have to bargain with time. So, empirically we fix this value to 100. This value achieves balance between performance and time span. This ClassificationTree template is passing as an argument to function fitensemble of MATLAB with few other arguments. Initial two arguments of fitensemble are input and class label of input. Third parameter of function is method we would like to use for classification. It is AdaBoost in our case and, specifically AdaBoostM2 as we have more than 2 classes in our data set. Algorithm iterates for 1000 number of ensemble learning cycle. At every training cycle, fitensemble does the same task. It loops over all learner templates in learners and trains one weak learner for every template. Next argument is the ClassificationTree template we just created to use. We perform this simulation with the learning rate of 0.75.

Training prepares neural ANN, later it is used to test a new sample and based on score value recognized class is stored in *Predictor* array.

C Representation

Recognition of symbol starts, a process of transforming a mere symbol into musical notes. Recognition constructs the first dimension of symbol. Yet, second dimension remains to be constructed. This is carried out using representation process. Process starts with reading *Predictor* array cell by cell and row by row. Value at current cell represent that class of sample in view is clef or accidental or rest or slur then move to next cell in *Predictor* array as for symbol in view, second dimension is not required to be constructed. Given symbol is notes of various lengths (i.e. whole note, half note…) search for circle(s) in symbol and locate the center(s) of each such circle. When more than one circle is available based on center coordinate arrange them from left to right, so that note with correct sequence pitch value is represented in data structure using ASCII value of alphabets (i.e. 99 means C). List out staff coordinates of symbol. Find out nearest staff coordinates for recognized center of circle and calculate pitch value. Save pitch value along with predictor value. Following Figure. 9 gives trace of entire OMR process spanned across four stages.

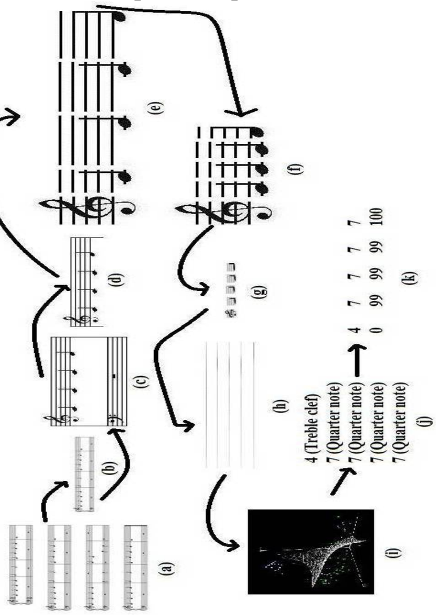


Figure. 9. OMR Process

IV. RESULTS DISCUSSION

This part of paper discusses results appeared during segmentation and recognition phase of OMR.

TABLE I SEGMENTATION RESULTS

Song Name	A	B	C	D
boom-boom-aint-it-great-to-be-crazy-easy-piano-solo	72	71	1	0
ten-little-indians-piano-solo	73	73	0	0
zum-gali-gali-piano	84	84	0	0
all-through-the-night-piano-solo	66	64	1	1
au-clair-de-la-lune-piano-solo	71	71	0	0
Bingo	81	81	0	0
trumpet-tune	82	82	0	0
peter-peter-piano	57	57	0	0
pretty-little-horses-piano	75	74	0	1
take-me-out-to-the-ball-game-piano	161	159	2	0
mexican-hat-dance-piano-solo	96	96	0	0
happy-birthday-piano	55	55	0	0
praise-god-from-whom-all-blessings-flow-piano	67	67	0	0
little_green_frog_piano	68	68	0	0
it-is-well-with-my-soul-piano	134	133	0	1
up-on-the-housetop-piano-solo	73	73	0	0
rootbeer-rootbeer-piano	138	138	0	0
good-king-wenceslas-piano-solo	84	84	0	0
jolly-old-saint-nicholas-piano-solo	75	75	0	0
camptown-races-piano	126	126	0	0
jesus-loves-me-easy-piano-solo	92	90	1	1
pretty-little-dutch-girl-piano	75	74	1	0
sakura-sakura-piano-solo	82	81	1	0
six-little-ducks-piano-solo	107	107	0	0
scotland-the-brave-piano	102	102	0	0
pop-goes-the-weasel-piano-solo	94	94	0	0
oh_susanna_piano	141	141	0	0
sixpence-piano	63	62	0	1
pick-a-bale-of-cotton-piano	80	80	0	0
my-ma-gave-me-a-nickel-piano-solo	103	102	0	1
mulberry-bush-piano	69	69	0	0
peace-like-a-river-piano	94	94	0	0
jingle-bells-easy-piano	75	75	0	0
whats-that-thing-piano	131	131	0	0
clementine-piano	120	117	2	1
song-of-the-bluebird-piano	66	66	0	0
twinkle-twinkle-little-star-piano-solo	81	81	0	0
the-b-i-b-l-e-piano-solo	73	71	1	1
its-raining-piano	71	71	0	0
mary-had-a-little-lamb-piano	68	68	0	0
hallelu-hallelu-piano	98	98	0	0
if_youre_happy_piano	63	63	0	0
hickory-dickory-dock-piano-solo	84	84	0	0
this-little-light-of-mine-piano-solo	85	85	0	0
michael-finnegan-for-piano-solo	77	77	0	0
a-tisket-a-tasket-piano	104	104	0	0
o-come-emmanuel-piano	136	136	0	0
lil-liza-jane-piano-solo	108	108	0	0
hes-got-the-whole-world-piano	60	57	1	2
michael_row_the_boat_ashore_piano	72	72	0	0
Laputa - Castle in the Sky – Gondoa	623	612	9	2
Laputa - Castle in the Sky - Shitsui	217	214	1	2
Laputa - Castle in the Sky - Laputa Theme	235	222	6	7
Total	**5487**	**5439**	**27**	**21**

Segmentation action designed is tested on 53 songs and implemented in MATLAB with help of Image Processing Toolbox. We use dataset available on various resources [14][15][16][17] to validate segmentation process.

Details are as follows. In Table I,

A = No. of Symbols Available, B = No. of Symbols Segmented Correctly, C = No. of Symbols Segmented Partly and D = No. of Symbols Missed.

Table I presents segmentation results. It highlights the fact that system segments symbols with ***99.12%*** accuracy. ***0.49%*** symbols are segmented partly and ***0.38%*** symbols are missed during segmentation process. This accuracy is achieved over more than 24 different symbols belongs to more than 5 different category of modern staff notations.

At all system works on 3588 samples. We use 75% of original samples for training the network and rest 25% samples for verifying network's learning capability.

Network is trained on 2692 samples distributed over 12 classes. We prepared two networks. One is trained on samples directly. While in another we extracted features from each sample by PCA using SVD. We measure the performance using different parameters. Such performance for such parameters with respect to both approaches is compared in following Table II and Table III.

TABLE II Confusion matrix for each class [PCA with SVD]

Class	FN	FP	TP	TN
Accidental Flat	0	11.11	88.89	100
Accidental Sharp	1.92	0	100	98.08
Clef Bass	0.24	0	100	99.76
Clef Treble	0	1.96	98.04	100
Note – Eighth	0	34.69	65.31	100
Note – Half	2.45	12.93	87.07	97.55
Note – Quarter	4.23	4.56	95.44	95.77
Note – Whole	0	5.88	94.12	100
Rest – Quarter	0	0	100	100
Rest – Whole	0.35	25	75	99.65
Rest – Half	0.69	13.64	86.36	99.31
Slur	0	37.5	62.5	100

TABLE III Confusion matrix for each class [Without PCA]

Class	FN	FP	TP	TN
Accidental Flat	0.23	33.33	66.67	99.77
Accidental Sharp	2.04	36.36	63.64	97.96
Clef Bass	0.24	0	100	99.76
Clef Treble	0.12	2	98	99.88
Note – Eighth	0.12	20.51	79.49	99.88
Note – Half	4.04	40	60	95.96
Note – Quarter	9.98	6.74	93.26	90.02
Note – Whole	0.34	7.14	92.86	99.66
Rest – Quarter	0	0.77	99.23	100
Rest – Whole	1.96	60	40	98.04
Rest – Half	1.72	54.55	45.45	98.28
Slur	0	58.33	41.67	100

Accuracy achieved by trained ANN is remarkable due to various facts. Accidental-flat is quite similar to half note and vice-versa. Sharp note also resembles whole notes and vice-versa. Difference between half and quarter note is only that circle in half note is empty and in quarter note it is filled. Half and Whole rest are distinguished by such a light difference that at first sight an experienced eye might make a mistake. Even notes like half, quarter and eights can appear with rotation of 90 degree. These are the reason that if we train ANN without extracting independent features using PCA network, it is not capable of recognizing with significant accuracy.

V. CONCLUSION

Segmentation process discussed here performs hierarchical decomposition with the help of thresholding based on mathematical analysis of printed piano music score sheet for segmentation of modern staff notation. Entire process is performed along with stave lines so system is not prune to any type of noise or problem of symbol deviation. Segmentation action is validated on 53 songs from various datasets available online. Process comes across 5487 symbols out of which 5439 symbols are segmented correctly. Thus system achieves prompt accuracy of ***99.12%*** for its segmentation work.

Recognition process is implemented using boosting based neural network classifier and compared to without using eature extraction method and along with the feature extraction method (i.e. PCA using SVD). System recognizes symbols with prompt accuracy of ***92.38%*** through the help of PCA and AdaBoost. Recognition accuracy goes down to ***84.19%*** if PCA is not used in couple of AdaBoost to extract features with dimensionality reduction. System works with 12 classes devised on various category of modern staff music notation. Pitch identification is carried out for polyphony music so it can also work for monophony music.

VI. FUTURE WORK

From reading an image file to rebuilding music note's two dimension structure work is completed. More dynamics can be added to segmentations algorithm to make the system work for more diverse score sheets and instruments.

REFERENCES

[1] Tatiana Tambouratzis, "Identifications of key music symbols for optical music recognition and on-screen presentation", IEEE, 2011.
[2] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R.S. Marcal, Carlos Guedes and Jaime S. Cardoso, "Optical music recognition state of the art and open issues", Springer, 2012.
[3] Dingyi Li, Bing Han and Jia Ji, "Music score reader based on morphological image processing", Stanford, 2012.
[4] Steve Dai, Ye Tian and Chun-Wei Lee, "Optical music recognition and playback on mobile devices", Stanford, 2012.
[5] Albert Ng and Asif khan, "Automated instructional aid for reading sheet music", Stanford, 2012.
[6] Toyokazu hori, Shinichiro wada, Howzan Tai and S. Y.Kung , "Automatic music score recognition/play system based on decision based neural network", IEEE, 1999.
[7] Christopher Raphael and Jingya Wang, "New approaches to optical music recognition", ISMIR 2011.
[8] Tomas Beran and Tomas Macek, "Recognition of printed music score", Springer, 1999.
[9] P. Bellini, I. Bruce and P. Nesi, "Optical Music Sheet Segmentation", IEEE, 2001.
[10] Mariusz Szwoch, "Guido: A Musical Score Recognition System", IEEE, 2007.
[11] Wladyslaw Homonda and Marcin Luckner, "Automatic knowledge acquisition: Recognizing music notation with methods of centroids and classification trees", IEEE, 2006.
[12] Bharath Modayur, "Music score recognition-A selective attention approach using mathematical morphology", University of Washington, 1996.
[13] Florence Rossant and Isabelle Bloch, "A fuzzy model for optical recognition of musical scores", Elsevier, 2004.
[14] International Music Score Library Project (IMSLP), Online Image dataset (http://imslp.org/).
[15] Jonathan Harnum, "Basic music theory", SolUt Press, 2001
[16] Piano Sheet Music Archieve (http://www.sheetmusicarchive.net/)
[17] Printed Piano Music Sheet (http://makingmusicfun.net)
[18] Apurva Mehta and Malay Bhatt, "Practical Issues in the Field of Optical Music Recognition", IJARCSMS,2014, Vol 2