Musical Note Position and Duration Recognition Model in Optical Music Recognition Using Convolutional Neural Network

Douglas Rakasiwi Nugroho * and Amalia Zahra

Department of Computer Science, Bina Nusantara University, Jakarta, Indonesia
Email: douglas.nugroho@binus.ac.id (D.R.N.); amalia.zahra@binus.edu (A.Z.)
*Corresponding author

***Abstract*—This study aims to solve Optical Music Recognition (OMR) problems using a non-End-to-End (non-E2E) approach. Therefore, separate models for Position Recognition (PR) and Duration Recognition (DR) are constructed, with both employing Convolutional Neural Networks (CNN). In terms of constructing a non-E2E architecture to solve OMR problems, this study obtains superior evaluation results compared to previous research, with the PR and DR models achieving accuracies of 97.88% and 99.23%, respectively. In addition, this study employs template matching in conjunction with several supplementary tasks to identify the positions of musical notes and generate the corresponding note sequences in the intended reading format. Our Optical Music Recognition (OMR) system can accomplish comparable results to the E2E architecture by utilizing these techniques.**

***Keywords*—non-end-to-end optical music recognition, convolutional neural network, position recognition model, duration recognition model, template matching**

I. Introduction

Almost every aspect of existence is supported by technology at this time in human history. Several typical human activities will be facilitated by technology, making daily life simpler. This concept gave birth to Artificial Intelligence, a digital representation of the human brain's abilities [1]. One of the human abilities is to "record music" [2].

In industrial and academic evolution, humans have long been striving to convert printed documents into machine-readable forms. The study on this topic was first initiated around 1930 and is known as the Optical Character Recognition (OCR) technique [3]. Current OCR implementation involves converting letter characters in images of printed documents into ASCII format so that they can be recognized by computers [4].

The field of research that discusses computer understanding of musical notation is usually called Optical Music Recognition (OMR). In contrast to OCR which reads letter characters, OMR is used to automatically interpret symbols in musical notation from printed music documents (scores) into a form that can be understood by computers [5]. OMR enables the converted music notation to be applied to various other objects [6] for further development.

Several previous studies have been conducted in the field of Optical Music Recognition (OMR). Some studies utilize an E2E approach, which involves solving OMR problems within a single architecture [7, 8], while others do not [9, 10]. This distinction implies that in the non-End-to-End (non-E2E) approach, each OMR problem is addressed separately (For example, separating the position and duration recognition model). This study aims to avoid employing an E2E approach, taking into account several potential limitations [11], while proposing an improved non-E2E approach.

The recent developments in OMR have shown promising outcomes by employing deep learning for symbol identification. The rationale behind using deep learning in OMR is to enable automatic learning and reduce the need for human intervention since the feature extraction is automatically done by the algorithm [12]. Therefore, in this study, the process of identifying position and duration will also employ deep learning techniques, specifically the Convolutional Neural Network (CNN), as it has shown positive results in object recognition [13].

This study begins by developing models for recognizing the positions and durations of musical notes. The evaluation of both models is then conducted, referring to the methods described in [14]. Subsequently, a musical note detection scenario is created to resemble how humans read musical notes, utilizing the template matching module from the cv2 library with specific adjustments. This scenario aims to detect the regions in the music sheet image that correspond to musical notes. The detected notes are then prioritized and arranged from top to bottom and left to right, as a person would read them. The OMR system then performs position and duration recognition for each image

Manuscript received June 2, 2023; revised June 27, 2023; accepted October 3, 2023; published February 6, 2024.

in the sequence using the previously developed models. Finally, the findings from applying the models are discussed at the end of the document. Therefore, this study aims to utilize a non-E2E approach to construct an OMR system.

II. Related Works

The commonly known OMR research was conducted by Calvo-Zaragoza and Rizo [7], resulting in an OMR system with an E2E approach. In their study, the developed architecture was able to accept single musical staff line images as input and output a sequential classification of symbols appearing in the image as output. Their study was then expanded to work with full-page documents (more than one staff line) [8]. The research has made significant contributions to OMR, particularly in providing an understanding of the tasks required for OMR. However, in our study, a non-E2E approach is employed, and different evaluation metrics are utilized.

The research on the non-E2E approach is not specifically stated as a non-E2E OMR. Research that separates the architecture for each OMR problem is considered as non-E2E OMR. One of the studies was carried out by Wel and Ullrich [9]. They employed OMR with Convolutional Sequence-to-Sequence as a solution to issues in OMR research [9]. They separated the models for pitch and duration recognition. Their investigation yielded results of 81% pitch accuracy, 94% duration accuracy, and 80% note accuracy (accuracy when pitch and duration are correct).

Research on non-E2E OMR was also conducted by Andrea and Paoline *et al.* [10] using the Convolutional Neural Network in recognizing musical note position. They developed a model that can recognize 13 positions of a musical note to determine the note's corresponding frequency. This study yielded training accuracy of 90%, validation accuracy of 85%, and assessment accuracy of 80%.

III. Dataset

A. *Printed Images of Music Staves (PrIMuS)* and *Camera Printed Images of Music Staves (Camera-PrIMuS)*

PrIMuS and Camera-PrIMuS are datasets created by Calvo-Zaragoza and Rizo [7]. PrIMuS contains musical note images which are representations of printed music documents. This dataset consists of 87,678 real musical inches. An incipit is a sequence of notes used to identify music. An example of PrIMuS can be seen in Fig. 1.

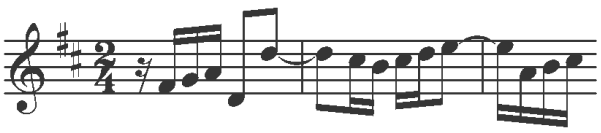

Fig. 1. Example of PrIMuS data before being cropped per musical note to form a new dataset (https://grfia.dlsi.ua.es/primus/).

Camera-PrIMuS is an extension of the PrIMuS dataset. Each image from PrIMuS is given a distortion treatment as a simulation of an image taken from a camera. An example of Camera-PrIMuS can be seen in Fig. 2.

Fig. 2. Example of Camera-PrIMuS data before being cropped per musical note to form a new dataset (https://grfia.dlsi.ua.es/primus/).

Instead of using PrIMuS and Camera-PrIMuS as they were in the original study, we extract each image of a single musical note segment from this data to create a new dataset. The range of positions and durations collected in this study can be observed in Fig. 6 and Table III.

B. *Additional Images from Noteflight*

To improve the model's performance, the dataset is intended to include all possible musical note forms. For the same position and duration of a note, there could be a distinct form symbol of the musical note. Therefore, every image of the position and duration used in this research is attempted to have a form as in Table I. This is one of the expanded aspects from previous non-E2E research, which did not take into account the distinctions in musical note form. However, specific musical note forms cannot be found in the PrIMuS and Camera-PrIMuS datasets (or it takes too much time to seek for them manually). Therefore, we cropped some images of musical notes using Noteflight, which is software for producing online music sheets for students [15].

TABLE I. Note Forms Collected for All Positions and Duration

Description	Example	Description	Example
Upper Flag		Upper Beam End	
Lower Flag		Lower Beam Start	
Upper Beam Start		Lower Beam Middle	
Upper Beam Middle		Lower Beam End	

We aim to use both PrIMuS and Camera-PrIMuS as parts of our datasets, considering their unique characteristics. Each of which represents the best state of the optical capture of the partiture and the condition when the camera distorts the optical capture of musical notes. The dataset used in this study does not use all PrIMuS and Camera-PrIMuS. As previously explained, the new dataset combines data from PrIMuS and Camera-PrIMuS with

images to complete the original dataset. The dataset is retrieved based on position and duration. Therefore, the dataset is divided into 13 classes based on position and five classes based on duration. The final dataset used in this study can be accessed at https://omr.douglasnugroho.com/downloads/dl-omr.zip.

IV. Proposed Method

The process visualized in Fig. 3 begins with obtaining the input image. The input image is then processed in some tasks which separate the image into several staff area, and a musical note detection procedure is executed. The process is then followed by musical note recognition, where OMR system predict the detected musical note positions and durations, resulting the final output: a sequence of recognized musical note positions and durations. The tasks performed during the musical note detection process include:

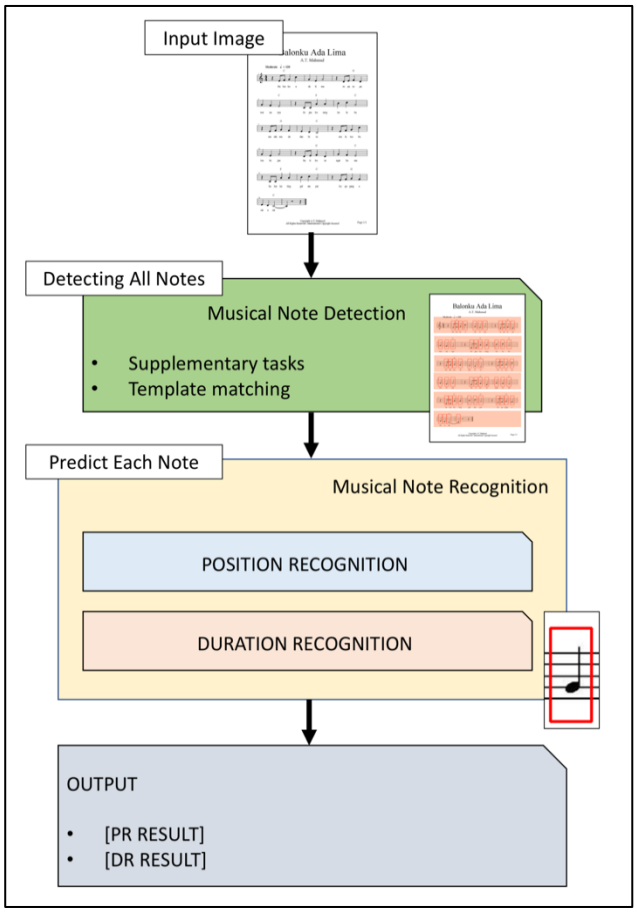

Fig. 3. Proposed OMR process: (1) Input image containing musical notation. (2) object detection applied to the input image using template matching. (3) once all musical notes are detected, cropping is performed. 4) each cropped segment is sequentially processed to predict its position and duration.

A. Affine Transformation

This method is used to straighten the tilted image [16]. The image must be straight so the staff can be extracted easily. The step to deskew starts with extracting the line with some minimum length from the image. Then, we find the angle from the line on the x-axis. After that, we do Affine Transformation to rotate the image.

B. Staff Line Extraction

The staff contains five parallel horizontal lines where the musical note is placed. This method is used to do line extraction that later will be clustered to construct each staff. First, we do the noise removal by doing erosion and dilation. This method is called "opening". After that, we do contours retrieval in the input image. A contour is a curve that joins all the continuous points having the same color or intensity. The contour that we retrieve is the staff's line. The result of the method is the line position.

C. Ward Algorithm

The Ward linkage method is a greedy algorithm used for hierarchical clustering. Hierarchical clustering is a nested partitioning of a point set into *n* clusters. Ward's method is agglomerative starting with *n* singleton clusters. Every step chooses two clusters in the current clustering which D (A, B) is minimal. The choice is optimal for the next clustering and merging until Ward's method produced clustered data into *n* cluster without initiating the *n* number [1]. This method divides all the detected line positions from the previous step into several clusters of staff in the sheet music image.

D. Template Matching

The next step is template matching to get the position of a musical note. Template matching involves locating the position of a smaller image, known as the template, within a larger image. The process involves sliding the template over the larger image, similar to 2D convolution, and comparing the patch of the larger image under the template to the template itself [17]. The template used in this study can be seen in Fig. 4.

Fig. 4. Template image to do musical note detection.

The image that we retrieve from the input image is the note dots symbol. After we get the dot position, we find the y1 and y2 positions by going up and down for a certain distance and find the x1 and x2 positions by going left and right for a certain distance. The illustration can be seen in Fig. 5.

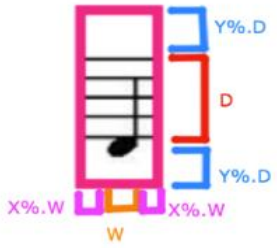

Fig. 5. Determining the detected musical note cutout image size. D = distance between the lower and higher staff line, W = dot's width, X and Y respectively are coefficients for horizontal and vertical expansion.

After acquiring information about each staff area and the placement of musical notes through template matching, we arrange all note cutaway images according to their y and x positions. This arrangement mirrors the natural manner in which human eyes interpret them on a sheet of music.

E. Convolutional Neural Network (CNN)

The proposed Position Recognition (PR) and Duration Recognition (DR) models are built using a Convolutional Neural Network (CNN). CNN consists of several layers. The convolutional layer will calculate the dot product of weight with a small region in the input volume. The pooling layer will reduce the complexity of the previous layer. The activation function layer is then used. It will map the input from the previous layer to the output enabling the CNN to learn. The data then will be flattened before going to the fully connected layer. The fully connected layer calculates the class score and is connected to all numbers and volumes. The model ends with a layer that has a SoftMax activation function to return the probabilities of each class [13].

F. Hyperparameter Tuning

The next proposed method is about the optimization of model performance. This study will examine hyperparameter tuning to optimize model performance. Two hyperparameters will undergo the tuning process, learning rate, and batch size. The tuning hyperparameters that will be examined are the learning rate and batch size. Changing the learning rate value is varied to increase or decrease with a scale of 2. Meanwhile, changes in the batch size value will vary with an increasing scale of 2. The hyperparameters studied can be seen in Table II.

TABLE II. HYPERPARAMETER VALUE USED FOR TUNING

Hyperparameter	Value		
Learning Rate	0.0001	0.001	0.01
Batch Size	16	32	64

V. EXPERIMENT

A. Dataset Preparation

Before starting the experiment, the data will be divided into 3 for training, validation, and testing of 13 classes for position and 5 classes for duration. The 13 position classes each represent the pitch that can be seen in Fig. 6. Then, 5 classes for duration can be seen in Table III.

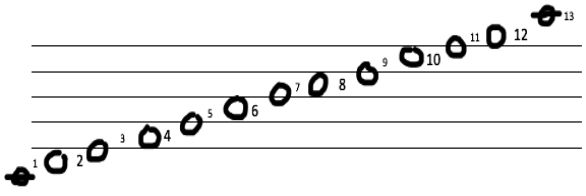

Fig. 6. 13 Position classes.

For this experiment, the data is divided into 60% for training, 20% for validation, and 20% for testing of each class so that the total images used are 3120 for training, 1040 for validation, and 1040 for testing. However, they are separated into different classes for a different purpose (position or duration). The detail can be seen in Tables IV and V.

TABLE III. FIVE DURATION CLASSES

Duration	Example
1	
1/2	
1/4	
1/8	
1/16	

TABLE IV. NUMBER OF IMAGES IN DATASET FOR 13 POSITION CLASSES

Position Classes	Number of Images (*Normal* + *Disort*)			Total
	Training	Validation	Testing	
1	120+120	40+40	40+40	400
2	120+120	40+40	40+40	400
3	120+120	40+40	40+40	400
4	120+120	40+40	40+40	400
5	120+120	40+40	40+40	400
6	120+120	40+40	40+40	400
7	120+120	40+40	40+40	400
8	120+120	40+40	40+40	400
9	120+120	40+40	40+40	400
10	120+120	40+40	40+40	400
11	120+120	40+40	40+40	400
12	120+120	40+40	40+40	400
13	120+120	40+40	40+40	400
Total	3120	1040	1040	5200

TABLE V. NUMBER OF IMAGES IN DATASET FOR 5 DURATION CLASSES

Duration Classes	Number of Images (*Normal* + *Disort*)			Total
	Training	Validation	Testing	
1	312+312	104+104	104+104	1040
1/2	312+312	104+104	104+104	1040
1/4	312+312	104+104	104+104	1040
1/8	312+312	104+104	104+104	1040
1/16	312+312	104+104	104+104	1040
Total	3120	1040	1040	5200

B. Architecture Layer for Recognition Model

This research algorithm relies on CNN with the PyTorch framework to create a robust model. The PR model has a total of 36 layers and the DR model has a total of 30 layers made up of Convolutional, Pooling, and Fully Connected Layers. These layers work together to deliver optimal performance as shown in Tables VI and VII.

TABLE VI. 36 LAYERS USED IN POSITION RECOGNITION CNN MODEL

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 8, 257, 65]	136
ReLU-2	[-1, 8, 257, 65]	0
MaxPool2d-3	[-1, 8, 128, 32]	0
Conv2d-4	[-1, 16, 129, 33]	2,064
ReLU-5	[-1, 16, 129, 33]	0
MaxPool2d-6	[-1, 16, 64, 16]	0
Conv2d-7	[-1, 32, 65, 17]	8,224
ReLU-8	[-1, 32, 65, 17]	0
MaxPool2d-9	[-1, 32, 32, 8]	0
Conv2d-10	[-1, 64, 33, 9]	32,832
ReLU-11	[-1, 64, 33, 9]	0
MaxPool2d-12	[-1, 64, 16, 4]	0
Conv2d-13	[-1, 128, 17, 5]	131,200
ReLU-14	[-1, 128, 17, 5]	0
MaxPool2d-15	[-1, 128, 8, 2]	0
Conv2d-16	[-1, 256, 9, 3]	524,544
ReLU-17	[-1, 256, 9, 3]	0
MaxPool2d-18	[-1, 256, 4, 1]	0
Conv2d-19	[-1, 512, 5, 2]	2,097,664
ReLU-20	[-1, 512, 5, 2]	0
MaxPool2d-21	[-1, 512, 2, 1]	0
Conv2d-22	[-1, 1024, 3, 2]	8,389,632
ReLU-23	[-1, 1024, 3, 2]	0
MaxPool2d-24	[-1, 1024, 1, 1]	0
Conv2d-25	[-1, 2048, 2, 2]	33,556,480
ReLU-26	[-1, 2048, 2, 2]	0
MaxPool2d-27	[-1, 2048, 1, 1]	0
Flatten-28	[-1, 2048]	0
Linear-29	[-1, 1024]	2,098,176
Linear-30	[-1, 512]	524,800
Linear-31	[-1, 256]	131,328
Linear-32	[-1, 128]	32,896
Linear-33	[-1, 64]	8,256
Linear-34	[-1, 32]	2,080
Linear-35	[-1, 16]	528
Linear-36	[-1, 13]	221

TABLE VII. 30 LAYERS USED IN DURATION RECOGNITION CNN MODEL

LAYER (TYPE)	OUTPUT SHAPE	PARAM #
CONV2D-1	[-1, 8, 257, 65]	136
RELU-2	[-1, 8, 257, 65]	0
MAXPOOL2D-3	[-1, 8, 128, 32]	0
CONV2D-4	[-1, 16, 129, 33]	2,064
RELU-5	[-1, 16, 129, 33]	0
MAXPOOL2D-6	[-1, 16, 64, 16]	0
CONV2D-7	[-1, 32, 65, 17]	8,224
ReLU-8	[-1, 32, 65, 17]	0
MaxPool2d-9	[-1, 32, 32, 8]	0
Conv2d-10	[-1, 64, 33, 9]	32,832
ReLU-11	[-1, 64, 33, 9]	0
MaxPool2d-12	[-1, 64, 16, 4]	0
Conv2d-13	[-1, 128, 17, 5]	131,200
ReLU-14	[-1, 128, 17, 5]	0
MaxPool2d-15	[-1, 128, 8, 2]	0
Conv2d-16	[-1, 256, 9, 3]	524,544
ReLU-17	[-1, 256, 9, 3]	0
MaxPool2d-18	[-1, 256, 4, 1]	0
Conv2d-19	[-1, 512, 5, 2]	2,097,664
ReLU-20	[-1, 512, 5, 2]	0
MaxPool2d-21	[-1, 512, 2, 1]	0
Conv2d-22	[-1, 1024, 3, 2]	8,389,632
ReLU-23	[-1, 1024, 3, 2]	0
MaxPool2d-24	[-1, 1024, 1, 1]	0
FLATTEN-25	[-1, 1024]	0
LINEAR-26	[-1, 256]	262,400
LINEAR-27	[-1, 128]	32,896
LINEAR-28	[-1, 64]	8,256
LINEAR-29	[-1, 32]	2,080
LINEAR-30	[-1, 13]	429

VI. RESULT

We carried out the PR and DR CNN model training with each running for 20 epochs. This model evaluates the results of hyperparameter tuning in Table II. Each tuned parameter produced records of best training and validation accuracy over the epochs as shown in Tables VIII and IX.

Therefore, it can be concluded that the best combination for PR model is when using a learning rate of 0.0001 and a batch size of 16 resulting in 96% training accuracy and 98% validation accuracy. Meanwhile, the best combination for DR model is when using a learning rate of 0.001 and a batch size of 64 resulting in the best results of 99% training accuracy and 100% validation accuracy for the duration recognition model.

TABLE VIII. EVALUATION OF DIFFERENT COMBINATIONS OF LR AND BS FOR POSITION RECOGNITION MODEL

Hyperparameter		Training Accuracy	Validation Accuracy
LR	BS		
0.01	16	0.07	0.08
0.001	16	0.07	0.08
0.0001	16	0.96	0.98
0.01	32	0.08	0.08
0.001	32	0.07	0.08
0.0001	32	0.94	0.96
0.01	64	0.07	0.08
0.001	64	0.91	0.95
0.0001	64	0.91	0.93

TABLE IX. EVALUATION OF DIFFERENT COMBINATIONS OF LR AND BS FOR DURATION RECOGNITION MODEL

Hyperparameter		Training Accuracy	Validation Accuracy
LR	BS		
0.01	16	0.20	0.20
0.001	16	0.19	0.20
0.0001	16	0.98	0.99
0.01	32	0.19	0.20
0.001	32	0.20	0.20
0.0001	32	0.93	0.96
0.01	64	0.18	0.20
0.001	64	0.99	1.00
0.0001	64	0.77	0.79

The model with the best combination of hyperparameters in the previous section is then continued by evaluating the prepared test data. The evaluation uses accuracy, precision, recall, and F1-score metrics. The results of testing for each model can be seen in Table X.

TABLE X. POSITION AND DURATION RECOGNITION MODEL EVALUATION RESULTS

Model	Accuracy	Precision	Recall	F1-score
PR	97.88%	97.92%	97.88%	97.90%
DR	99.23%	99.24%	99.23%	99.24%

The experiment continued by providing an image of a music sheet as shown in Fig. 7 as an input image to be read by the OMR system.

Next, musical note detection was performed on the music sheet image. In this stage, the previously developed template matching module was capable of detecting all the musical notes present in Fig. 7. The creation of this template matching module initiates a complete non-E2E OMR research, where the OMR system can autonomously locate the musical note objects that need to be read. The output from the template matching module will then be further processed to PR and DR models. The detected musical notes can be seen in Fig. 8. It can be observed that the template matching module is able to detect 57 musical notes in the entire music sheet image.

After obtaining 57 detected musical note cutout images, each cutout image sequentially is predicted for its position and duration. The results of position and duration recognition, as well as the true class, can be seen in Table XI. In these results, it was found that there were four mispredictions in the position of the segmented musical notes, while there were no mispredictions in the duration of the cutout image of musical notes.

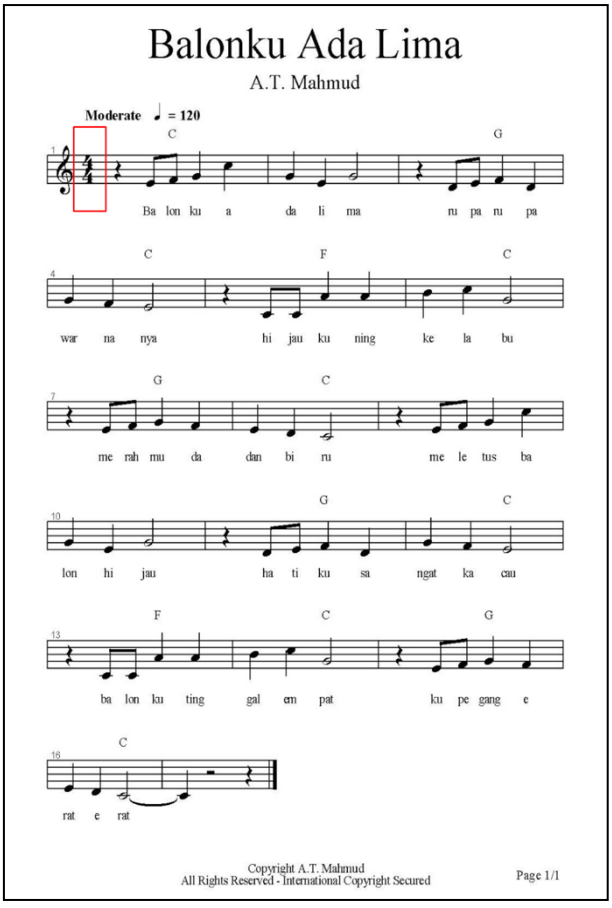

Fig. 7. Balonku Ada lima music sheet (http://yohanesocta.blogspot.com/2016/05/partitur-balonku-ada-lima-ciptaan-at.html).

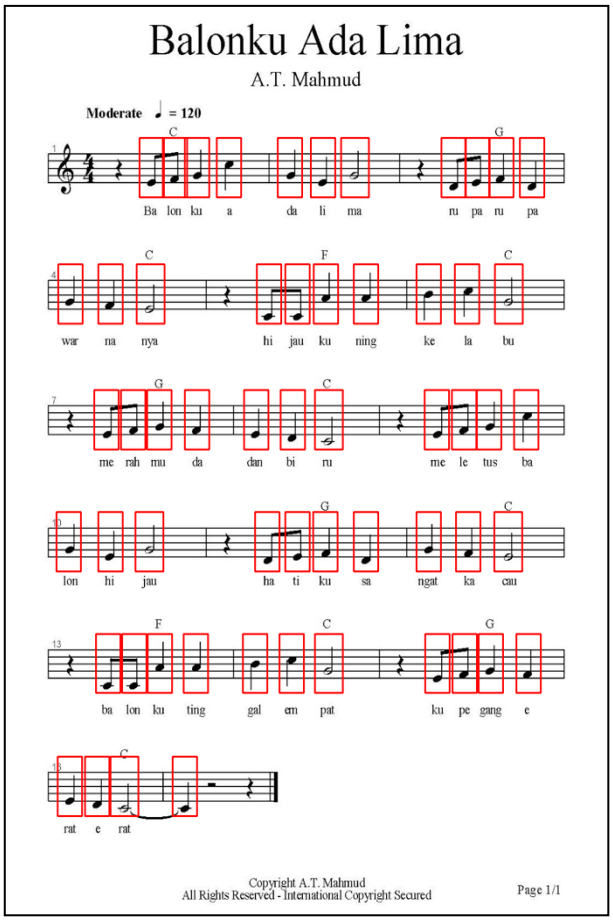

Fig. 8. Detected musical notes using template matching module.

In the first experiment, several variables affected the window size of the cutout image for each musical note. The window size of the detected musical note was defined as follows: X equals 90% of W, and Y equals 54% of D. For a more detailed visualization, refer to Fig. 9 to see the size of the image cutout for each musical note.

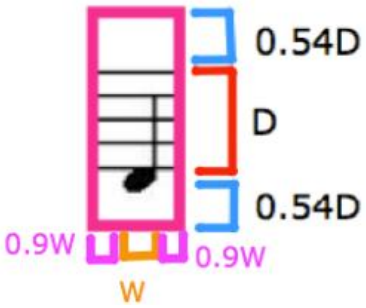

Fig. 9. The size of the detected musical note image cutout in the First experiment.

TABLE XI. PREDICTION RESULTS AND TRUE CLASSES OF EACH IMAGE CUTOUT IN THE FIRST EXPERIMENT. P = PREDICTION, AND T = TRUE CLASS

No	Pos		Dur		No	Pos		Dur	
	P	T	P	T		P	T	P	T
1	1	3	1/8	1/8	30	4	4	1/8	1/8
2	4	4	1/8	1/8	31	5	5	1/4	1/4
3	5	5	1/4	1/4	32	8	8	1/4	1/4
4	8	8	1/4	1/4	33	5	5	1/4	1/4
5	5	5	1/4	1/4	34	3	3	1/4	1/4
6	3	3	1/4	1/4	35	5	5	1/2	1/2
7	5	5	1/2	1/2	36	2	2	1/8	1/8
8	2	2	1/8	1/8	37	4	3	1/8	1/8
9	3	3	1/8	1/8	38	4	4	1/4	1/4
10	4	4	1/4	1/4	39	2	2	1/4	1/4
11	2	2	1/4	1/4	40	5	5	1/4	1/4
12	5	5	1/4	1/4	41	4	4	1/4	1/4
13	4	4	1/4	1/4	42	3	3	1/2	1/2
14	4	3	1/2	1/2	43	1	1	1/8	1/8
15	1	1	1/8	1/8	44	2	1	1/8	1/8
16	1	1	1/8	1/8	45	6	6	1/4	1/4
17	6	6	1/4	1/4	46	6	6	1/4	1/4
18	6	6	1/4	1/4	47	7	7	1/4	1/4
19	7	7	1/4	1/4	48	8	8	1/4	1/4
20	8	8	1/4	1/4	49	5	5	1/2	1/2
21	5	5	1/2	1/2	50	3	3	1/8	1/8
22	3	3	1/8	1/8	51	4	4	1/8	1/8
23	4	4	1/8	1/8	52	5	5	1/4	1/4
24	5	5	1/4	1/4	53	4	4	1/4	1/4
25	4	4	1/4	1/4	54	3	3	1/4	1/4
26	3	3	1/4	1/4	55	2	2	1/4	1/4
27	2	2	1/4	1/4	56	1	1	1/2	1/2
28	1	1	1/2	1/2	57	1	1	1/4	1/4
29	3	3	1/8	1/8					

TABLE XII. PREDICTION RESULTS AND TRUE CLASSES OF EACH IMAGE CUTOUT IN THE SECOND EXPERIMENT. P = PREDICTION, AND T = TRUE CLASS

No	Pos		Dur		No	Pos		Dur	
	P	T	P	T		P	T	P	T
1	3	3	1/8	1/8	30	4	4	1/8	1/8
2	4	4	1/8	1/8	31	5	5	1/4	1/4
3	5	5	1/4	1/4	32	8	8	1/4	1/4
4	8	8	1/4	1/4	33	5	5	1/4	1/4
5	5	5	1/4	1/4	34	3	3	1/4	1/4
6	3	3	1/4	1/4	35	6	5	1/2	1/2
7	6	5	1/2	1/2	36	3	2	1/8	1/8
8	3	2	1/8	1/8	37	4	3	1/8	1/8
9	3	3	1/8	1/8	38	4	4	1/4	1/4
10	4	4	1/4	1/4	39	3	2	1/4	1/4
11	3	2	1/4	1/4	40	5	5	1/4	1/4
12	5	5	1/4	1/4	41	4	4	1/4	1/4
13	4	4	1/4	1/4	42	3	3	1/2	1/2
14	4	3	1/2	1/2	43	2	1	1/8	1/8
15	2	1	1/8	1/8	44	2	1	1/8	1/8
16	2	1	1/8	1/8	45	6	6	1/4	1/4
17	6	6	1/4	1/4	46	6	6	1/4	1/4
18	6	6	1/4	1/4	47	7	7	1/4	1/4
19	7	7	1/4	1/4	48	8	8	1/4	1/4
20	8	8	1/4	1/4	49	6	5	1/2	1/2
21	6	5	1/2	1/2	50	3	3	1/8	1/8
22	3	3	1/8	1/8	51	4	4	1/8	1/8
23	4	4	1/8	1/8	52	5	5	1/4	1/4
24	5	5	1/4	1/4	53	4	4	1/4	1/4
25	4	4	1/4	1/4	54	3	3	1/4	1/4
26	3	3	1/4	1/4	55	2	2	1/4	1/4
27	3	2	1/4	1/4	56	1	1	1/2	1/2
28	2	1	1/2	1/2	57	1	1	1/4	1/4
29	3	3	1/8	1/8					

Next, a second experiment was conducted by modifying the size of the detected musical note cutout image window. The window size of the detected musical note is defined as follows: X equals 100% of W, and Y equals 80% of D as illustrated in Fig. 10.

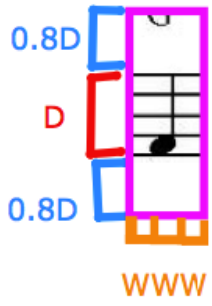

Fig. 10. The size of detected musical note image cutout in the second experiment.

In the second experiment, the sequential recognition results of the position and duration for each recognized musical note are shown in Table XII. There are differences from the first experiment. Note number 1 was successfully predicted correctly in terms of its position, but the other findings show that there were several additional musical note cutout images that were incorrectly predicted in terms of their position. However, in terms of DR, the model once again successfully predicted the duration class for all 57 musical note cutout images correctly.

It was found that the position recognition model in this study was still affected by the window size of the musical note cutout images. In contrast to the duration recognition model, the two experiments carried out always succeeded in predicting all musical notes correctly. This is presumably because, as shown in Tables VI and V, the quantity of data for each class in the dataset for the PR model has fewer variations in terms of the window size (the position of the musical note dot symbol over the image) than the dataset for the DR model.

VII. CONCLUSION

This study successfully developed two models, which are PR and DR models. The accuracy, precision, recall, and F1-score for PR model were 97.88%, 97.92%, 97.88%, and 97.90%, respectively. As for DR model, the corresponding metrics were 99.23%, 99.24%, 99.23%, and 99.24%. It can be concluded that the two models created to establish the non-E2E OMR yield better results compared to the previous referenced studies.

Furthermore, the subsequent experiments in this study led to the conclusion that the performance of the PR model was still affected by the size of the musical note cutout image window provided, while the DR model performed well and was not affected by the size of the musical note cutout image window size. Therefore, the next challenge in this research is to increase the variation of musical note dot symbol positions over the images used as the dataset for the position recognition model.

This study did not specifically evaluate the performance of musical note detection because it only utilized an

existing model, which is template matching from cv2. The significant time required for detection is a drawback of using this method. Further discussion should address the creation of musical note detection so that it is not necessary to conduct additional tasks, as in this study, in order to detect musical notes and determine their sequence of appearance. However, the use of template matching and other efforts demonstrated in this study for musical note detection serves as an idea for future development in non-E2E OMR.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

AZ initiated the research topic, provided guidance in the research, and assisted in refining the research findings. DRN contributed to the creation of the dataset, conducted experiments, documented the experimental results, and contributed to the paper writing. All authors had approved the final version.

Funding

This research is funded by the Directorate of Research, Technology, and Community Service; the Directorate General of Higher Education, Research, and Technology; and the Ministry of Education, Culture, Research, and Technology of Indonesia, in accordance with the Research Contract Fiscal Year of 2023 No 1402/LL3/AL.04/2023, June 26, 2023.

Acknowledgment

The authors wish to thank the Directorate of Research, Technology, and Community Service; the Directorate General of Higher Education, Research, and Technology; and the Ministry of Education, Culture, Research, and Technology of Indonesia.

References

[1] B. P. Sutton *et al.*, "Impact of technology on human life," *Life Sci. J.*, vol. 1, no. 7, pp. 417–423, 2019.

[2] W. Lemberg, "A survey of the history of musical notation," *TUGboat*, vol. 37, no. 3, pp. 284–304, 2016.

[3] A. Chaudhuri, K. Mandaviya, P. Badelia, and S. K. Ghosh, *Optical Character Recognition Systems for Different Languages with Soft Computing*, 2017, vol. 352.

[4] F. Mohammad, J. Anarase, M. Shingote, and P. Ghanwat, "Optical character recognition implementation using pattern matching," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 7, no. 8, pp. 1092–1095, 2019.

[5] J. Novotný and J. Pokorný, "Introduction to optical music recognition: Overview and practical challenges," in *Proc. CEUR Workshop*, vol. 1343, pp. 65–76, 2015.

[6] J. Calvo-Zaragoza, J. Hajic, and A. Pacha, "Understanding optical music recognition," *ACM Comput. Surv.*, vol. 53, no. 4, pp. 1–31, 2020.

[7] J. Calvo-Zaragoza and D. Rizo, "Camera-primus: Neural end-to-end optical music recognition on realistic monophonic scores," in *Proc. 19th Int. Soc. Music Inf. Retr. Conf. ISMIR 2018*, 2018, pp. 248–254.

[8] F. J. Castellanos, J. Calvo-Zaragoza, and J. M. Inesta, "A neural approach for full-page optical music recognition of mensural documents," in *Proc. 21st Int. Soc. Music Information Retr. Conf.*, October, 2020.

[9] E. V. D. Wel and K. Ullrich, "Optical music recognition with convolutional sequence-to-sequence models," in *Proc. 18th Int. Soc. Music Inf. Retr. Conf. ISMIR 2017*, 2017, pp. 731–737.

[10] Andrea, Paoline, and A. Zahra, "Music note position recognition in optical music recognition using convolutional neural network," *Int. J. Arts Technol.*, vol. 13, no. 1, pp. 45–60, 2021.

[11] T. Glasmachers, "Limits of end-to-end learning," in *Proc. Machine Learning Research, ACML 2017*, pp. 17–32.

[12] L. Alzubaidi *et al.*, *Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions*, 2021, vol. 8, no. 1, Springer International Publishing.

[13] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, p. 52, 2023.

[14] Ž. Vujović, "Classification model evaluation metrics," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, 2021.

[15] About—Noteflight Music Notation Software. (2023). [Online]. Available: https://www.noteflight.com/company/about

[16] OpenCV: Affine Transformations. (2023). [Online]. Available: https://docs.opencv.org/3.4/d4/d61/tutorial_warp_affine.html

[17] OpenCV: Template Matching. (2023). [Online]. Available: https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html