Stacking ensemble learning for optical music recognition

Francisco Calvin Arnel Ferano, Amalia Zahra, Gede Putra Kusuma
Department of Computer Science, BINUS Graduate Program-Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Article Info

Article history:

Received Nov 2, 2022
Revised Dec 12, 2022
Accepted Jan 29, 2023

Keywords:

Common western musical notation
Convolutional neural network
Ensemble learning
Machine learning
Optical music recognition

ABSTRACT

The development of music culture has resulted in a problem called optical music recognition (OMR). OMR is a task in computer vision that explores the algorithms and models to recognize musical notation. This study proposed the stacking ensemble learning model to complete the OMR task using the common western musical notation (CWMN) musical notation. The ensemble learning model used four deep convolutional neural networks (DCNNs) models, namely ResNeXt50, Inception-V3, RegNetY-400MF, and EfficientNet-V2-S as the base classifier. This study also analysed the most appropriate technique to be used as the ensemble learning model's meta-classifier. Therefore, several machine learning techniques are determined to be evaluated, namely support vector machine (SVM), logistic regression (LR), random forest (RF), K-nearest neighbor (KNN), decision tree (DT), and Naïve Bayes (NB). Six publicly available OMR datasets are combined, down sampled, and used to test the proposed model. The dataset consists of the HOMUS_V2, Rebelo1, Rebelo2, Fornes, OpenOMR, and PrintedMusicSymbols datasets. The proposed ensemble learning model managed to outperform the model built in the previous study and succeeded in achieving outstanding accuracy and F1-scores with the best value of 97.51% and 97.52%, respectively; both of which were achieved by the LR meta-classifier.

Corresponding Author:

Francisco Calvin Arnel Ferano
Department of Computer Science, BINUS Graduate Program-Master of Computer Science
Bina Nusantara University
11480 Jakarta, Indonesia
Email: francisco.ferano@binus.ac.id

1. INTRODUCTION

Music is often described as structured notes in time and musical notation is a music representation that visually communicates that definition of music [1]. Music is an art of human culture that is passed down from generation to generation. Generational changes also make changes and developments in the music culture itself. The development of this musical culture eventually brought it to the stage it is today, where musical notation can be described using a very common notation, namely the common western music notation (CWMN). This music notation has become an international representation and at the same time the most common in representing music in writing. This musical notation eventually became a problem in the field of computer vision which had the basic idea of making computers able to recognize musical symbols in this musical notation, just like humans. This idea finally led researchers to various problems that must be solved, so that the computer can recognize and detect musical symbols well. This problem is known as optical music recognition (OMR). OMR is a field of study that studies and develops computer algorithms and models to recognize musical notation in a document [2], with CWMN as the most common musical notation used in the study. CWMN is composed of musical symbols representing music. Figure 1 shows several symbols used in

CWMN to represent music elements. Other than those symbols, there are also bar lines, staff lines, trills, and other CWMN symbols.

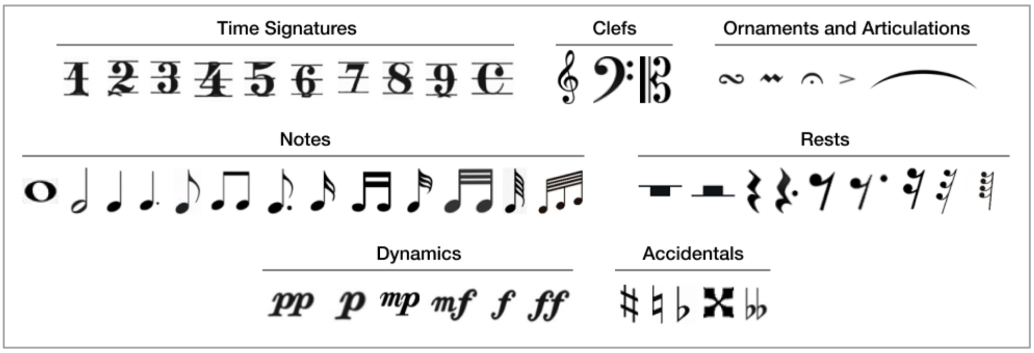

Figure 1. Several CWMN symbols

OMR has several benefits in everyday life. Music notation can be converted into several forms, such as musical instrument digital interface (MIDI) for playing the music and MusicXML for storing music notation in the form of sheet music documents. The benefits contributed by this OMR research greatly support musicians in practicing, exploring musical plays, and writing songs in CWMN musical notation. All the benefits provided by this OMR study are highly dependent on the model's performance in detecting and recognizing symbols in a musical score, which is determined by the method and the results of the classification of musical symbols. Therefore, it is imperative to determine the appropriate and accurate classification method to be used in OMR experiments. Determining a good classification method for OMR eventually becomes a problem in the field of study.

Researchers have conducted many studies and proposed various models capable of detecting and recognizing CWMN symbols. Mejía *et al.* [3] experimented with classifying music sheet images using several baseline architecture convolutional neural networks, namely VGG16, MobileNet, ResNet50, Inception V3, and Inception-ResNet-V2. The experiment shows good performances of the five models, with MobileNet achieving the highest accuracy. Other studies using datasets containing images of music scores or staves pieces have been carried out using several models or other techniques, such as a deep convolutional neural network (DCNN) using the darknet53 basic network on YOLO [4], deep watershed detector (DWD) [5], parallel bat algorithm [6], U-Net [7], [8], and several variations of the model using convolutional recurrent neural network (CRNN) [9]–[13]. These models have shown good results in carrying out the OMR task. In addition to these studies, some studies perform classification tasks at the symbol level of CWMN. These studies are conducted using datasets that contain images of cropped musical notation symbols, where each image will only contain one symbol. Some studies used several variations of the feature extractor followed by the K-nearest neighbor (KNN) classifier [14], [15]. There is also a study that applies a texture-based feature descriptor (daisy descriptor) that is optimized using quantum concept inspired gray wolf optimization and is continued by comparing the performance of several classifiers, namely multi-layer perceptron (MLP), KNN, Naive Bayes (NB), random forest (RF) and sequential minimal optimization (SMO) [16]. Although these studies have produced good and very good results in some of the studies, these results can still be improved.

In performing classification tasks, the ensemble learning method has been widely used in other areas of classification problems and has produced excellent performance results [17]–[23], but as far as is known at the time of writing this study, there is only one study that applied this method to OMR [24]. Ensemble classification is a learning method with a mining approach that utilizes various classifiers that distinguish class labels for unlabeled things from accumulation [25]. The main idea of creating learning ensembles is to improve prediction performance by constructing multiple models or multiple predictions [26]. Paul *et al.* [24], proposed an OMR ensemble learning using three pre-trained deep learning models, namely ResNet50, DenseNet161, and GoogLeNet as the base-classifier models. The base-classifier segment of the model is then followed by a support vector machine (SVM) meta-classifier. However, in this study, several DCNN models which are newer and superior to the three models were selected to be designed as ensemble models. Those models are ResNeXt50, Inception-V3, RegNetY-400MF, and EfficientNet-V2-S.

As the improved model of ResNet, ResNeXt can outperform ResNet in experiments [27], [28] and can perform classification well [29], [30]. It has only a few hyperparameters to set and its cardinality can improve classification accuracy. Inception-V3 was introduced in [31] and proven to be able to do classification tasks very well [32]–[34]. RegNet [35] and EfficientNet-V2 [36] can be said to be still fairly new models since they were first introduced in 2020 and 2021 respectively. Both models have been used several times in the study of classification and have also shown that both models can perform well on the given task [37]–[42]. Therefore, based on those studies, these four models are chosen in this research to be contained as the ensemble base-classifier models for the OMR ensemble model.

In addition, because there are no other studies that carry out ensembles on the OMR task, it is necessary to experiment and analyze the meta-classifier method that can perform well in the OMR ensemble model, which has not been done so far. Several machine learning techniques are chosen and analysed in this study, namely, SVM, logistic regression (LR), RF, KNN, decision tree (DT), and NB. There are also six musical notation symbols datasets used in this study, namely the handwritten online music symbols (HOMUS) version 2 [43], Rebelo1 [44], Rebelo2 [44], Fornes [45], OpenOMR [46], and PrintedMusicSymbols [47] datasets. These datasets are a collection of OMR datasets which do not contain music score images, but cropped CWMN symbols, so that these symbols are no longer placed on staff lines. These datasets are then combined, hence producing a unified dataset. Using the proposed model and the determined datasets, this study will only focus on boosting the performance result produced by the model without considering other assessment variables, such as time and resources required. All experiments built were run using the Python programming language on Google Colab Pro. The Python programming language version used is Python 3.8.15. The experiment was run using the available GPU on Google Colab Pro, which is one among the NVDIA Tesla K80, P100, and T4 randomly selected by the server. The results of the performance of each DCNN and the results of the ensemble of each meta-classifier in each dataset have been presented in this study.

2. METHOD

2.1. Overview

In this study, an ensemble learning design has been proposed to perform the OMR task. The designed learning ensemble consists of four DCNN architectures in the base-classifier segment and one machine learning classifier in the meta-classifier segment. Several DCNN architectures that are used as the base-classifier models are ResNeXt50, Inception-V3, RegNetY-400MF, and EfficientNet-V2-S. These models in the base-classifier segment will carry out the classification task independently. This classification process produces a collection of predictions produced by the four models of the base-classifier segment. This collection of predictions are given to the meta-classifier as the input data. The meta-classifier will once again perform the classification process and produce final prediction for each data in the dataset. Hence, ensemble prediction is produced. In this meta-classifier segment, a machine learning technique will be used as the meta-classifier of the proposed ensemble learning method. To determine the best machine learning technique to be applied to the ensemble learning model for the OMR task, several machine learning techniques, namely SVM, LR, RF, KNN, DT, and NB, are chosen to be analysed in this study. The designed ensemble learning model is tested against a unified OMR dataset which contains six OMR datasets that are publicly available, namely HOMUS_V2, Rebelo1, Rebelo2, Fornes, OpenOMR, and PrintedMusicSymbols. These datasets are combined, forming a unified dataset. As a result, in this study, six experiments have been carried out with each machine learning technique as the meta-classifier of the ensemble model. Figure 2 shows the workflow overview of the conducted experiments in this study.

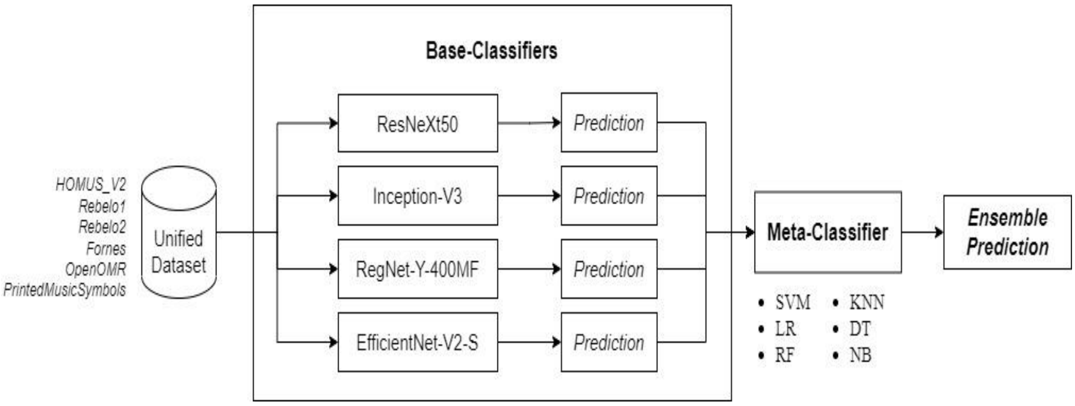

Figure 2. Proposed method's workflow overview

2.2. Dataset

The dataset used in this study is a dataset that contains images of CWN music notation symbols. Both printed and handwritten symbols are used. Pacha and Eidenberger [48] managed to build a tool that has made a huge contribution to the OMR field. The tool is a python package called the omrdatasettools. This tool can retrieve several datasets created by several researchers, namely HOMUS_V2, Rebelo1, Rebelo2, Fornes, OpenOMR, and PrintedMusicSymbols datasets. Each dataset will be extracted to produce six folders representing each of the six datasets. Each of these folders contains various CWMN symbol images. HOMUS_V2 and Fornes are datasets containing handwritten musical notation symbols. Meanwhile, Rebelo1, Rebelo2, OpenOMR, and PrintedMusicSymbols are datasets containing printed music notation symbols. Table 1 shows the overview of each dataset mentioned.

Table 1. Dataset overview

Dataset	Number of classes	Type of musical symbols	Total images	Range of data counts per class
HOMUS_V2	32	Handwritten	15,200	396-801
Rebelo1	30	Printed	7,940	6-897
Rebelo2	56	Printed	7,307	1-508
Fornes	7	Handwritten	4,094	471-820
OpenOMR	15	Printed	706	4-112
PrintedMusicSymbols	36	Printed	213	1-63

These datasets are then combined, forming a unified dataset. In the combining process, the dataset is analysed so that there is no overlap in the classes. Data that has the same class, but different class writings have been changed to be labeled as one proper class label and merged into the same class folder, resulting in a dataset containing 64 classes and 35,460 images of musical symbols. Due to the limited resources used in this study, this dataset was down sampled by determining that each class only accommodates a maximum of 300 images to ease the training process. This dataset will hereinafter be referred to as the Downsampled300Unified dataset. The Downsampled300Unified dataset contains 64 classes and 12,401 images of CWMN symbols (34.97% of the whole unified dataset). After the dataset is created, some classes have too little data. There even exists two classes that only have one sample. This can cause problems when splitting data into training, validation, and test set. Therefore, these classes, which count as many as 14 classes, were removed from the dataset. That way, the Downsampled300Unified dataset contains 50 classes and 12,256 CWMN symbol images (34.56% of the entire unified dataset) with the smallest number of samples being 73. Figure 3 shows several sample images of the Downsampled300Unified dataset.

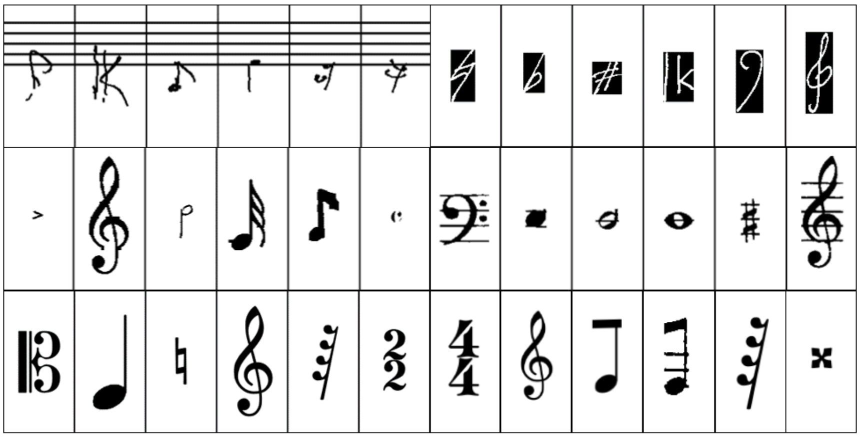

Figure 3. Sample images of the Downsampled300Unified dataset

Figure 4 shows the distribution of the number of images in each class in the Downsampled300Unified dataset and Table 2 shows the amount of data in each class. The dataset has an unbalanced amount of image data. This is left so to evaluate the model's performance in dealing with unbalanced datasets. Thus, no data augmentation process is carried out in this study.

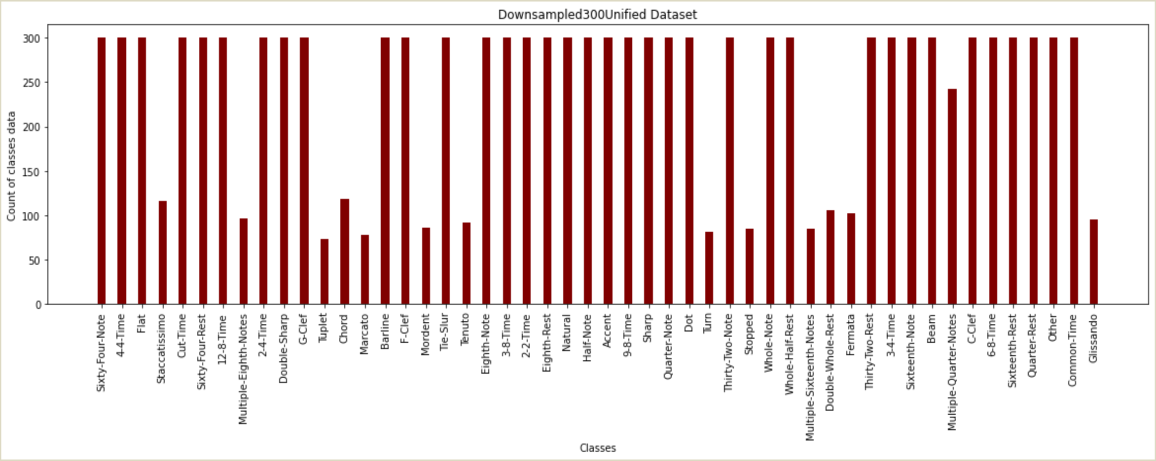

Figure 4. Distribution of the number of images in each class in the Downsampled300Unified dataset

Table 2. The amount of data in each class

Class	Amount of data	Class	Amount of data
12-8-time	300	Quarter-rest	300
2-2-time	300	Sharp	300
2-4-time	300	Sixteenth-note	300
3-4-time	300	Sixteenth-rest	300
3-8-time	300	Sixty-four-note	300
4-4-time	300	Sixty-four-rest	300
6-8-time	300	Thirty-two-note	300
9-8-time	300	Thirty-two-rest	300
Accent	300	Tie-slur	300
Barline	300	Whole-half-rest	300
Beam	300	Whole-note	300
C-Clef	300	Multiple-quarter-notes	243
Common-time	300	Chord	118
Cut-time	300	Staccatissimo	116
Dot	300	Double-whole-rest	106
Double-sharp	300	Fermata	102
Eighth-note	300	Multiple-eighth-notes	96
Eighth-rest	300	Glissando	95
F-clef	300	Tenuto	92
Flat	300	Mordent	86
G-clef	300	Multiple-sixteenth-notes	85
Half-note	300	Stopped	85
Natural	300	Turn	81
Other	300	Marcato	78
Quarter-note	300	Tuplet	73

For the preparation of the experiment, the dataset used will be given a little pre-processing, namely changing the size of the image. Each musical notation image in the dataset will be resized to 299×299 pixels. This image size was chosen because it is the minimum size of the input image that can be accepted by Inception-V3. Other DCNN models are also capable of accepting input images of this size. The input image is not resized for each model to give the same treatment to all models so that the stacking ensemble concept can be fully applied. No further pre-processing method is required to enhance the images on the dataset.

The dataset is then split into train, valid, and test data with the portion of 60%, 20%, and 20% respectively. The amount of train data is set at 60% to prevent the training process from being too long since four DCNNs will go through the classification task. All these split data are then given to each DCNN in the base-classifier segment.

2.3. Ensemble learning

There are three commonly used ensemble methodologies, namely bagging, stacking, and boosting [25]. In this study, stacking ensemble learning has been chosen to be applied to the OMR task. Stacking ensemble learning has been determined to be used because, in this type of ensemble learning, each base-classifier model will be given the whole data in the dataset, in contrast to bagging ensemble learning where each base-classifier model will only get a part or subset of the dataset used. This makes the result obtained in [26] show that stacking ensemble learning can outperform other types of ensemble learnings in the problems studied.

The ensemble learning in this study is built using several DCNN models as the base classifier, namely ResNeXt50, Inception-V3, RegNetY-400MF, and EfficientNet-V2-S. Each model will produce its prediction for all musical symbol images in the given dataset. The prediction results of these models are very likely to have differences. Therefore, in the ensemble learning model, the class prediction results generated by all base classifiers will be used as knowledge and input for a meta-classifier. This meta-classifier will provide the final result of the classification process for all data in the given dataset.

The meta-classifier in this learning ensemble can be performed using various machine learning techniques. Therefore, further analysis is needed to find the best technique to be applied to OMR ensemble learning. Several machine learning techniques that excel in classification tasks have been determined to be used as meta-classifiers in ensemble learning, namely SVM, LR, RF, KNN, DT, and NB. Results of the ensemble learning classification performance of each meta-classifier in each dataset are then reported to determine which of the machine learning techniques produced the best result in the OMR task.

2.4. Evaluation

In this study, the data is split into training, validation, and testing data with percentages of 60%, 20%, and 20% respectively. The split dataset is then given to the ensemble model. The learning ensemble that is built in this study will be given several evaluations. In each classification process carried out by all base-classifier models, accuracy calculations will be carried out. This is done to provide a report on the results of the classification performance carried out by each base-classifier model on the data provided without any ensemble learning.

After the prediction results by the base-classifiers are given to the meta-classifier, the meta-classifier will produce the final prediction results for each data in the dataset. These predictions can be mapped in a multi-class confusion matrix that contains the true positives, true negatives, false positives, and false negatives values. Using these values, the accuracy and F1-score evaluation metric can be calculated. Accuracy score can be calculated as in (1). F1-score will be calculated for each class in the dataset as in (2). After F1-score of each class are calculated, then the average value of the f1-score which concludes the value of the f1 score for the entire ensemble model can be calculated as in (3).

$$Accuracy = \frac{(Total_{True\ Positive} + Total_{True\ Negative})}{n_{data}} \tag{1}$$

$$\text{F1-score} = \frac{True\ Positive}{True\ Positive + \frac{1}{2}(False\ Positive + False\ Negative)} \tag{2}$$

$$\text{AVG F1-score} = \frac{Total\ F1{-}Scores}{n_{class}} \tag{3}$$

Since in this study six meta-classifier methods are analysed, therefore 12 evaluation metric scores are produced in the experiment. Using these evaluation metric scores, the best meta-classifier can be concluded. Meta-classifier that yields the highest evaluation metric scores is considered the best meta-classifier.

The built ensemble model is also evaluated by comparing the model with a comparative model. This is done by comparing the accuracy scores produced by the six meta-classifiers of both ensemble models. Therefore, the comparative ensemble model is also given several experiments using the determined six machine learning techniques as the meta-classifier of the ensemble model. In this way, the improvements provided by the built model ensemble can be clearly reported and assessed.

3. RESULTS AND DISCUSSION

This study has experimented with performing the proposed method of ensemble learning on an OMR task using the determined Downsampled300Unified dataset. The designed ensemble model consists of four DCNNs as the base classifiers, namely ResNeXt50, Inception-V3, RegNetY-400MF, and EfficientNet-V2-S, and one machine learning technique as the meta-classifier of the ensemble model. To determine the best machine learning technique to be used as the OMR ensemble model’s meta-classifier, several machine learning techniques are determined to be performed in the ensemble model, namely SVM, LR, RF, KNN, DT, and NB.

The experiment is done to produce several results. The first result is the performance of the determined four base classifier models (the four DCNNs) on the dataset. The second result is the performance results of the ensemble learning model on the dataset using each meta-classifier. The last result is the best meta-classifier produced in this research through the performed experiment.

The performance of each base classifier model is evaluated using the accuracy score. In the base-classifier segment of the ensemble model, each DCNN base classifier performed the classification task on the given dataset, the Downsampled300Unified dataset. This classification task can be evaluated using the accuracy score, thus the performance of each DCNN can be evaluated. Table 3 shows each DCNN's accuracy scores in classifying CWMN symbols in the Downsampled300Unified dataset. The experiment showed that EfficientNet-V2-S outperformed the other DCNN in terms of accuracy score. The model produced a 0.9735 accuracy score. The best accuracy score ranking was followed by Inception-V3 with an accuracy score of 0.9694, then ResNeXt50 with an accuracy score of 0.9670, and RegNetY-400MF with an accuracy score of 0.9625 respectively.

Table 3. Base classifiers accuracy scores on the Downsampled300Unified dataset

Model	Accuracy score
ResNeXt50	0.9670
Inception-V3	0.9694
RegNetY-400MF	0.9625
EfficientNet-V2-S	0.9735

After the four DCNNs' performances are evaluated, the ensemble model built will then be evaluated. This is done by performing six experiments since in this research, six machine learning techniques are determined to be used as the meta-classifier of the proposed ensemble model. The classification predictions of the meta-classifiers are evaluated using two evaluation metrics, namely the accuracy score and the F1-score. Table 4 shows the accuracy and F1-scores of each meta-classifier in performing the classification task using the Downsampled300Unified dataset.

Table 4. Proposed ensemble model's evaluation metric scores using each meta-classifier

Meta-classifier	Accuracy	F1 score
SVM	0.9747	0.9748
LR	0.9751	0.9752
RF	0.9731	0.9732
KNN	0.9751	0.9751
DT	0.9580	0.9600
NB	0.9662	0.9676

Through the carried-out experiment, it has been obtained that the proposed ensemble learning model has produced outstanding evaluation metric scores. Almost all the evaluation metric scores achieved 0.97 scores with the best value of 0.9751 in terms of accuracy score and 0.9752 in terms of F1-score, both of which were achieved by the LR meta-classifier and also KNN in the accuracy score. This indicates that the proposed ensemble learning model succeeded in achieving a good performance in completing the OMR task. Using this evaluation metric results, the best meta-classifier can be determined. The conducted experiment has shown that LR produced the highest accuracy and F1-score with slight differences from the other meta-classifiers accuracy and F1-scores. Thus, LR succeeded in achieving the best meta-classifier perfectly.

The proposed ensemble learning model is then compared with the ensemble learning that has been proposed by [24]. Therefore, an ensemble model that uses ResNet50, DenseNet161, and GoogLeNet as the base classifiers is also built. The model is also performed in six experiments using each determined machine learning technique determined in this study as the meta-classifier. The experiment is also done using the same dataset, therefore there is no difference in the data volume between the dataset used using both ensemble models. This model is used as the comparative model. The comparison process is done by comparing both models' accuracy scores using each meta-classifier. Table 5 shows the comparison result between the proposed model and the comparative study.

Through this comparison process, it can be seen proved that the proposed model outperformed the comparative model in every meta classifier's ensemble accuracy. The biggest accuracy score margin occurred in the DT meta-classifier, with a margin of 0.0163. the second and the third biggest margin occurred in the RF and NB meta-classifier with the margin of 0.0082 and 0.0077 respectively. The fourth and fifth biggest margin occurred in the LR and KNN meta-classifier with the margin of 0.0053 and 0.0049 respectively. The smallest margin occurred in the SVM meta-classifier with the margin of 0.0029.

Table 5. Comparison between the proposed model and the comparative study

Ensemble model	Base-classifiers	Meta-classifier's accuracy score					
		SVM	LR	RF	KNN	DT	NB
Proposed	ResNeXt50 Inception-V3 RegNetY-400MF EfficientNet-V2-S	0.9747	0.9751	0.9731	0.9751	0.9580	0.9662
Comparative model [24]	ResNet50 DenseNet161 GoogLeNet	0.9719	0.9698	0.9649	0.9702	0.9417	0.9584

4. CONCLUSION

This study aims to build model ensembles with newer and more robust base classifier models and to analyze several machine learning techniques to be used as the ensemble learning model's meta-classifier for the OMR task. The study only focused on boosting the performance result produced by the model without considering other assessment variables, such as time and resources required. A stacking ensemble learning model has been designed to use four DCNNs, namely ResNeXt50, Inception-V3, RegNetY-400MF, and EfficientNet-V2-S, followed by a machine learning meta-classifier. Six machine learning techniques are determined to be analysed, namely SVM, LR, RF, KNN, DT, and NB. The model is tested against the Downsampled300Unified dataset.

The proposed model has succeeded in increasing the ensemble learning performance score compared to the previous study. It is proven that several models that are proven good to be used on the classification task can improve the performance of an OMR ensemble learning model, such as ResNeXt50, Inception-V3, RegNetY-400MF, and EfficientNet-V2-S. As a result, the proposed ensemble learning model succeeded in achieving outstanding evaluation metric scores in completing the OMR task.

Through the conducted experiment performed using only the base-classifier models, it is shown that EfficientNet-V2-S outperformed the other models with a slight difference in the accuracy score. The model succeeded in obtaining an accuracy score of 0.9735. The experiment on the ensemble model has also shown an outstanding result. Almost all the evaluation metric scores achieved 0.97 with the best values of 0.9751 and 0.9752 in terms of accuracy and F1-score, respectively; both of which were achieved by the LR meta-classifier.

This study has succeeded in building an ensemble learning for the OMR task and has produced very good results. The study has also analyzed machine learning techniques that are good for use as a meta-classifier of OMR ensemble learning. This study opens business opportunities to create a music notation recognition software or application that is capable of accurately recognizing musical symbols. So that musicians will be supported in doing their work. Academically, further OMR research that is built using ensemble learning can use this study as a reference. In addition, because the OMR task is similar to the OCR task, the ensemble model proposed in this study can also be built for OCR research.

Although the proposed model has produced good results, there are still some things that can be improved in further research. In this study, the four DCNNs used are the lowest architectural structures of their kind. This is done to ease the process of running experiments on the ensemble model. In the next study, this problem can be experimented with the same DCNN model but using a more complex architectural arrangement, such as using the large or medium version of EfficientNetV2. In addition, other DCNN models that are proven to be faster and do not reduce model performance can also be used. This will make the research not only focused on boosting the evaluation metric results but can also consider the time costs incurred by the model.

ACKNOWLEDGEMENTS

This study is supported by Bina Nusantara university and the authors would like to thank the university for providing the facilities and resources to carry out this research.

REFERENCES

[1] E. Shatri and G. Fazekas, "Optical music recognition: state of the art and major challenges," *Arxiv-Computer Science*, vol. 2, pp. 1–10, 2020.

[2] J. C.-Zaragoza, J. Hajic, and A. Pacha, "Understanding optical music recognition," *ACM Computing Surveys*, vol. 53, no. 4, pp. 1–35, 2020, doi: 10.1145/3397499.

[3] D. J. L. -Mejia, E. P. V. -Uribe, and W. Ugarte, "Content-based image classification for sheet music books recognition," in *2020 IEEE Engineering International Research Conference (EIRCON)*, 2020, pp. 1–4, doi: 10.1109/EIRCON51178.2020.9254010.

[4] Z. Huang, X. Jia, and Y. Guo, "State-of-the-art model for music object recognition with deep learning," *Applied Sciences*, vol. 9, no. 13, pp. 1–14, 2019, doi: 10.3390/app9132645.

[5] L. Tuggener, I. Elezi, J. Schmidhuber, and T. Stadelmann, "Deep watershed detector for music object recognition," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*, 2018, pp. 271–278.

[6] A. N. Younis and F. M. Ramo, "A new parallel bat algorithm for musical note recognition," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 558–566, 2021, doi: 10.11591/ijece.v11i1.pp558-566.

[7] J. Hajič, M. Dorfer, G. Widmer, and P. Pecina, "Towards full-pipeline handwritten OMR with musical symbol detection by U-NETS," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*, 2018, pp. 225–232.

[8] A. Pacha, J. Hajič, and J. C. -Zaragoza, "A baseline for general music object detection with deep learning," *Applied Sciences*, vol. 8, no. 9, pp. 1–21, 2018, doi: 10.3390/app8091488.

[9] A. Baró, P. Riba, and A. Fornés, "A starting point for handwritten music recognition," in *The International Society for Music Information Retrieval 2018 Workshop on Music Reading Systems*, 2018, pp. 1–2.

[10] A. R. -Vila, J. C. -Zaragoza, and J. M. Inesta, "Exploring the two-dimensional nature of music notation for score recognition with end-to-end approaches," in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2020, pp. 193–198, doi: 10.1109/ICFHR2020.2020.00044.

[11] A. R. -Vila, J. C. -Zaragoza, and D. Rizo, "Evaluating simultaneous recognition and encoding for optical music recognition," in *ACM International Conference Proceeding Series*, 2020, pp. 10–17, doi: 10.1145/3424911.3425512.

[12] C. De *et al.*, "Multimodal audio and image music transcription," in *The 22nd International Society for Music Information Retrieval Conference*, 2021, pp. 1–3.

[13] A. Baró, P. Riba, J. C. -Zaragoza, and A. Fornés, "From optical music recognition to handwritten music recognition: a baseline," *Pattern Recognition Letters*, vol. 123, pp. 1–8, 2019, doi: 10.1016/j.patrec.2019.02.029.

[14] S. A. Nawade, R. Pardeshi, C. Dhawale, and M. Hangarge, "Old handwritten music symbol recognition using the combination of foreground and background projection profiles," in *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, 2018, pp. 1–4, doi: 10.1109/IoT-SIU.2018.8519881.

[15] S. A. Nawade, M. Hangarge, C. Dhawale, M. B. I. Reaz, R. Pardeshi, and N. Arsad, "Old handwritten music symbol recognition using directional multi-resolution spatial features," in *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, 2018, pp. 1–4, doi: 10.1109/ICSCEE.2018.8538370.

[16] S. Malakar, M. Ghosh, A. Chaterjee, S. Bhowmik, and R. Sarkar, "Offline music symbol recognition using Daisy feature and quantum Grey wolf optimization based feature selection," *Multimedia Tools and Applications*, vol. 79, no. 43–44, pp. 32011–32036, 2020, doi: 10.1007/s11042-020-09638-3.

[17] A. Neena and M. Geetha, "Image classification using an ensemble-based deep CNN," *Advances in Intelligent Systems and Computing*, vol. 709, pp. 445–456, 2018, doi: 10.1007/978-981-10-8633-5_44.

[18] L. D. Nguyen, R. Gao, D. Lin, and Z. Lin, "Biomedical image classification based on a feature concatenation and ensemble of deep CNNs," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2019, doi: 10.1007/s12652-019-01276-4.

[19] H. Sun and J. Yang, "Domain-specific image classification using ensemble learning utilizing open-domain knowledge," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, 2019, pp. 593–596, doi: 10.1109/ICCNC.2019.8685507.

[20] M. A. Elaziz, A. Mabrouk, A. Dahou, and S. A. Chelloug, "Medical image classification utilizing ensemble learning and levy flight-based honey badger algorithm on 6G-enabled internet of things," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–17, 2022, doi: 10.1155/2022/5830766.

[21] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, vol. 2020, pp. 1–11, 2020, doi: 10.1155/2020/8885861.

[22] D. N. Diniz *et al.*, "A deep learning ensemble method to assist cytopathologists in pap test image classification," *Journal of Imaging*, vol. 7, no. 7, pp. 1–19, 2021, doi: 10.3390/jimaging7070111.

[23] A. U. Berliana and A. Bustamam, "Implementation of stacking ensemble learning for classification of COVID-19 using image dataset CT scan and lung X-ray," in *2020 3rd International Conference on Information and Communications Technology, ICOIACT 2020*, 2020, pp. 148–152, doi: 10.1109/ICOIACT50329.2020.9332112.

[24] A. Paul, R. Pramanik, S. Malakar, and R. Sarkar, "An ensemble of deep transfer learning models for handwritten music symbol recognition," *Neural Computing and Applications*, vol. 34, no. 13, pp. 10409–10427, 2022, doi: 10.1007/s00521-021-06629-9.

[25] A. Parmar, R. Katariya, and V. Patel, "A review on random forest: an ensemble classifier," in *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018*, 2019, vol. 26, pp. 758–763, doi: 10.1007/978-3-030-03146-6_86.

[26] D. Müller, I. S. -Rey, and F. Kramer, "An analysis on ensemble learning optimized medical image classification with deep convolutional neural networks," *IEEE Access*, vol. 10, pp. 66467–66480, 2022, doi: 10.1109/ACCESS.2022.3182399.

[27] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, pp. 5987–5995, doi: 10.1109/CVPR.2017.634.

[28] T. Zhou, Y. Zhao, and J. Wu, "ResNeXt and Res2Net structures for speaker verification," in *2021 IEEE Spoken Language Technology Workshop, SLT 2021 - Proceedings*, 2021, pp. 301–307, doi: 10.1109/SLT48900.2021.9383531.

[29] D. P. Yadav, A. S. Jalal, D. Garlapati, K. Hossain, A. Goyal, and G. Pant, "Deep learning-based ResNeXt model in phycological studies for future," *Algal Research*, vol. 50, p. 102018, 2020, doi: 10.1016/j.algal.2020.102018.

[30] G. Pant, D. P. Yadav, and A. Gaur, "ResNeXt convolution neural network topology-based deep learning model for identification and classification of Pediastrum," *Algal Research*, vol. 48, p. 101932, 2020, doi: 10.1016/j.algal.2020.101932.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[32] C. Wang *et al.*, "Pulmonary image classification based on inception-v3 transfer learning model," *IEEE Access*, vol. 7, pp. 146533–146541, 2019, doi: 10.1109/ACCESS.2019.2946000.

[33] T. Vijayan, M. Sangeetha, and B. Karthik, "Efficient analysis of diabetic retinopathy on retinal fundus images using deep learning techniques with inception V3 architecture," *Journal of Green Engineering*, vol. 10, no. 10, pp. 9615–9625, 2020.

[34] N. Dong, L. Zhao, C. H. Wu, and J. F. Chang, "Inception v3 based cervical cell classification combined with artificially extracted features," *Applied Soft Computing*, vol. 93, 2020, doi: 10.1016/j.asoc.2020.106311.

[35] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10425–10433, doi: 10.1109/CVPR42600.2020.01044.

[36] M. Tan and Q. V. Le, "EfficientNetV2: smaller models and faster training," in *Proceedings of the 38th International Conference on Machine Learning, PMLR*, 2021, pp. 1–11.

[37] Y. Ye *et al.*, “An improved efficientNetV2 model based on visual attention mechanism: application to identification of cassava disease,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–16, 2022, doi: 10.1155/2022/1569911.

[38] B. Z. Demiray, M. Sit, and I. Demir, “DEM super-resolution with efficientNetV2,” *Arxiv-Electrical Engineering and Systems Science*, vol. 1, pp. 1–7, 2021.

[39] C. K. Sunil, C. D. Jaidhar, and N. Patil, “Cardamom plant disease detection approach using efficientNetV2,” *IEEE Access*, vol. 10, pp. 789–804, 2022, doi: 10.1109/ACCESS.2021.3138920.

[40] Z. Wang *et al.*, “Scene classification of remote sensing images using efficientNetV2 with coordinate attention,” in *Journal of Physics: Conference Series*, 2022, pp. 1–6, doi: 10.1088/1742-6596/2289/1/012026.

[41] K. N. Phan, H.-H. Nguyen, V.-T. Huynh, and S.-H. Kim, “Facial expression classification using fusion of deep neural network in video,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 2506–2510, doi: 10.1109/CVPRW56347.2022.00280.

[42] M. K. Mahbub, M. Biswas, A. M. Miah, A. Shahabaz, and M. S. Kaiser, “COVID-19 detection using chest X-ray images with a RegNet structured deep learning model,” *Communications in Computer and Information Science*, vol. 1435, pp. 358–370, 2021, doi: 10.1007/978-3-030-82269-9_28.

[43] J. C. -Zaragoza and J. Oncina, “Recognition of pen-based music notation: the HOMUS dataset,” in *Proceedings - International Conference on Pattern Recognition*, 2014, pp. 3038–3043, doi: 10.1109/ICPR.2014.524.

[44] A. Rebelo, G. Capela, and J. S. Cardoso, “Optical recognition of music symbols - a comparative study,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 13, no. 1, pp. 19–31, 2010, doi: 10.1007/s10032-009-0100-1.

[45] A. Fornés, J. Lladós, and G. Sánchez, “Old handwritten musical symbol classification by a dynamic time warping based method,” in *Graphics Recognition. Recent Advances and New Opportunities*, W. Liu, J. Lladós, and J.-M. Ogier, Eds. Berlin, Heidelberg: Springer, 2008, pp. 51–60, doi: 10.1007/978-3-540-88188-9_6.

[46] A. F. Desaedeleer, “Reading sheet music,” University of London, 2006.

[47] A. Pacha, “Printed music symbols dataset,” *GitHub*, 2017. Online. [Available]. https://github.com/apacha/PrintedMusicSymbolsDataset (accessed Jun. 09, 2022)

[48] A. Pacha and H. Eidenberger, “Towards a universal music symbol classifier,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2018, pp. 35–36, doi: 10.1109/ICDAR.2017.265.

BIOGRAPHIES OF AUTHORS

Francisco Calvin Arnel Ferano is currently a student at Bina Nusantara University, Indonesia, majoring in Computer Science Master Program. He is enrolled in the Master Track of Computer Science study program and is currently under Mrs. Amalia’s guidance for this research. His research interest is in the field of computer science. He can be contacted at email: francisco.ferano@binus.ac.id.

Amalia Zahra is a lecturer at the Master of Information Technology, Bina Nusantara University, Indonesia. She received her bachelor’s degree in computer science from the Faculty of Computer Science, University of Indonesia (UI) in 2008. She does not have a master’s degree. Her Ph.D was obtained from the School of Computer Science and Informatics, University College Dublin (UCD), Ireland in 2014. Her research interests cover various fields in speech technology, such as speech recognition, spoken language identification, speaker verification, and speech emotion recognition. Additionally, she also has an interest in natural language processing (NLP), computational linguistics, machine learning, and artificial intelligence. She can be contacted at email: amalia.zahra@binus.edu.

Gede Putra Kusuma received Ph.D degree in Electrical and Electronic Engineering from Nanyang Technological University (NTU), Singapore, in 2013. He is currently working as a lecturer and head of Department of Computer Science, BINUS Graduate Program, Bina Nusantara University, Indonesia. Before joining Bina Nusantara University, he was working as a research scientist in I2R–A*STAR, Singapore. His research interests include computer vision, pattern recognition, deep learning, face recognition, appearance-based object recognition, and indoor positioning system. He can be contacted at email: inegara@binus.edu.