Deep Learning for Advanced Similar Musical Instrument Detection and Recognition

Christine Dewi and Rung-Ching Chen*

Abstract— **Researchers in the domains of machine learning and artificial intelligence are particularly interested in the problem of object identification. Detection and identification of similar objects are one of the most difficult challenges in computer vision image recognition. This study will identify the following musical instruments: cello, clarinet, erhu, guitar, saxophone, trumpet, French horn, harp, recorder, bassoon, and violin. Various musical instruments have the same size, shape, and sound. Furthermore, we are impressed by the ease with which humans can identify very similar objects, and for computers, this is a difficult task. In this work, we attempted to distinguish between things that seemed to be extremely similar at the level of human perception. Next, we deploy Yolo V4 to determine which musical instruments are comparable to one another. Following that, the performance of the Yolo V4 and Densenet models will be evaluated. We can improve the detection performance of musical instruments that are similar based on the outcomes of our experiments. In comparison to the findings of other experiments, Yolo V4 demonstrates the highest possible average accuracy, coming in at 94.70 percent and better than previous methods.**

Index Terms—**Musical Instrument Detection, Similar object, Densenet, Yolo V4, Deep learning.**

I. Introduction

Object detection is a kind of computer technology related to computer vision and image processing. This technology is concerned with finding examples of semantic objects of a particular class, including musical instruments [1][2], people [3][4], buildings [5], traffic sign [6][7], or cars [8][9] in video and digital images. Despite the widespread use of object detectors, the performance of object detectors may be uneven in some circumstances. The flute and clarinet have several complementary characteristics. As members of the same family, the articulation, and dynamic capabilities of the two instruments are similar. They are identical in terms of shape, size, and sound. Moreover, Cello and violin are classifying as members of the string family, and they are different from each other. The most significant distinction between a cello and a violin is their size.

Manuscript received September 28, 2021; revised August 8, 2022. This paper is supported by the Ministry of Science and Technology, Taiwan. The Nos are MOST-110-2927-I-324-501 and MOST-109-2622-E-324 -004, Taiwan.

C. Dewi is a senior lecturer from Faculty of Information Technology, Satya Wacana Christian University, Central Java, 50711, Indonesia (e-mail: christine.dewi13@gmail.com).

R. C. Chen is the Professor of the Department of Information Management, Chaoyang University of Technology, Taichung, 41349, Taiwan, (e-mail: crching@cyut.edu.tw).

When playing the Cello, it is customary to sit in a sitting posture with the instrument held between the legs. On the other hand, the violin is resting between the shoulder and the chin of the player.

When played, cellos and violins have one thing in common: they both need a bow. As with the violin, the cello is played with the right hand using a bow that crosses the four strings. According to Figure 1, the guitar, the violin, and the cello all have a similar basic shape that varies to some degree. Computers have a considerably more difficult time than people do when trying to identify musical instruments that are like one another. Our research works were particularly impressed by the ease with which humans can identify visual identification difficulties, such as recognizing very similar musical instrument objects. Moreover, while this situation is obvious to humans, it poses difficulties for computers.

Yolo exerts the most impact in situations requiring quicker time detection. It has a high detection rate and a high degree of accuracy. Yolo V4, the latest version of Yolo, was announced in 2020. Most recent scientific models require multiple GPUs for training and large mini-batch sizes. According to past research, when training with a single GPU, the training process is lengthy, tiring, and ultimately ineffective. Yolo V4 [10][11] takes a new solution to this problem by training object detectors on a single GPU with a lower mini-batch size than previously used. With this technique, it is possible to train faster on a single GPU, which is extremely precise.

CNN models, such as Densenet and Yolo V4, as well as feature extraction approaches, are examined in this article. In the course of our investigation, we adapted them to the People Playing Musical Instrument (PPMI dataset) [12]. The PPMI dataset consists of pictures of individuals interacting with twelve distinct musical instruments. On the list are the cello, bassoon, clarinet, flute, French horn, erhu, guitar, harp, recorder, trumpet, saxophone, and violin. In research articles, it is difficult to find many object detectors based on deep learning that are uniquely tailored to the musical instrument identification problem area. We have had difficulty locating one that assesses numerous crucial variables, including *mAP*, *IoU*, and detection time.

The following is a summary of the paper's contributions. Human vision was the first step in the process of identifying items that were quite similar. Yolo V4 is used to identify musical instruments that share similar characteristics. Next, the Yolo V4 model is evaluated in terms of detection time, *mAP*, *IoU*, and floating-point functions (FLOPS). As part of this investigation, we'll identify several musical instruments that are similar.

(a) Bassoon (b) Cello (c) Clarinet (d) Erhu (e) Flute (f) French horn

(g) Guitar (h) Harp (i) Recorder (j) Saxophone (k) Trumpet (l) Violin

Figure 1. Musical Instruments.

The following is the structure of this paper: Related work is included in Section II. Section III describes the approach that we suggest. Section IV included an explanation of the experiment and its findings. A detailed explanation of our study findings is provided in Section V of this report. Finally, in Section VI, conclusions are reached and recommendations for further research are made.

II. Related Works

A. Similar Object Recognition

Deep learning recognition has enabled significant advances in most object identification algorithms [13][14]. Object recognition is simple for humans, but it is very difficult for computers to distinguish two things that are almost identical in appearance and function. The two-stage detector consists of two processes that work together. First, using a region-based CNN (RCNN) [15][16], the detector extracts recommendations for areas where items can be found in the image. After that, it categorizes each region of interest (RoI) separately [17].

However, although the two-stage detector has excellent performance, it has significant drawbacks. It takes a long time to train a model, and longer to test it, because of the two procedures involved. A single-stage detector is recommended to minimize the amount of prediction time. This time is required by predicting the position of the item as well as its class and score at the same time. The single-stage detector does not have a region proposal procedure because it is a one-step detector. Yolo [18] and Single Shot Detector (SSD) [19] are the most representative single-stage detectors. Both have only one CNN architecture [20]. Single-stage detectors are more efficient, have more competitive overall performance, and have fewer model parameters than their two-stage counterparts. Shijin Song et al. [21] developed a more efficient CNN network design that allows small objects to be identified more accurately while requiring less computation and enabling simpler deployment. They cut the CNN network, greatly reducing the size and operating time of the model and maintaining accuracy. The fully convoluted layers are replaced with fully connected layers at the same time, which increases the computational efficiency.

With the help of entropy loss, M. Ju et al. [22] have developed an object identification approach that can more accurately distinguish between objects that have similar appearances. If the detector uses entropy loss, it is more likely to generate accurate predictions about the bounding box class that is being seen. The degradation of trust is also lessened because of this. As a result, similar objects are better able to be detected.

B. Yao et al. [12] proposed a new representation of image features named "grouplets" be developed. Grouplets are used to capture structured information contained in an image by storing several discriminatory visual characteristics and their spatial arrangement in the image. With the use of a dataset consisting of seven different PPMI activity, the author shows that grouplets outperform other state-of-the-art methods in terms of categorization and detection of human-object interactions in a variety of situations.

The Generative Adversarial Network (GAN) was used in conjunction with the Yolo method by C. Dewi et al. [23] to detect musical instruments that are similar to each other. Yolo is fast Region based Convolutional Neural Network (CNN) with powerful computation. Yolo-GAN will increase the Yolo detection process capacity and surpass the original Yolo capability when Deep Convolution Yolo-GAN is used. In our experiment we will implement the newest version of Yolo V4 with the PPMI dataset with 12 different musical instruments.

B. Yolo V4 and Densenet

The most recent version of Yolo is Yolo V4, published by [10] in 2020. Moreover, the Yolo V4 implement CSPDarknet53 [24] as a backbone network. Also, Spatial Pyramid Pooling in the neck structure and Yolo V3 [25] in the head layer. Yolo V4 makes use of a Mish [26] activation mechanism to operate in the backbone. Mish is a contemporary, smooth, and non-monotonic characteristic of the neural activation function that can be observed in Equation (1).

$$f(x) = x \tanh\left(\ln(1 + e^x)\right) \quad (1)$$

Where, $\ln(1 + e^x)$ is the soft plus activation function [27].

In addition, Yolo V4 offers several additional enhancement methods, including the following: (1) Use a new data augmentation technique, such as mosaic and adversarial training, to supplement the existing data. (2) Select the optimal hyperparameters through the Genetic Algorithm. (3) Improve the efficiency of current techniques by making them more suited for efficient training and inference, including SAM and PAN [28]s. Complete Yolo V4 specifications divided by two as follows: (1) Bag of Freebies (BoF) [29] Backbone. (2) Bag of Specials (BoS) Backbone.

Yolo V4 uses many techniques that build on the foundation of Yolo V3 for training and customization, while also enhancing the core network, essentially without making any significant structural changes. Yolo V3 splits the input image into ($N \times N$) grids cells [30] with the same size and forecast bounding boxes and probabilities for each grid cell. It takes advantage of multi-scale fusion to provide predictions, and a single neural network is used to generate a complete picture. Like the previous box, dimension clusters are used to forecast bounding boxes, which are then applied as bounding boxes. Based on this, the K-means approach is utilized to carry out dimensional clustering on the target boxes contained within the dataset. As a result, nine prior boxes of diverse sizes that are equally dispersed over feature graphs at various scales are obtained. Moreover, Yolo V3 admits personal bounding box anchor for each ground truth object [31].

The Densenet is primarily comprised of three components: the Dense Block, the Transition Layer, and the Growth Rate [32]. As input, every layer in Densenet receives everything from previous layers, and as output, every layer in Densenet receives everything from prior levels [33]. Densenet offers a number of appealing benefits, including the fact that it promotes feature reuse while also alleviating the vanishing gradient issue [34][35]. However, there are some apparent flaws in it as well. Before combining feature maps acquired from previous layers by concatenating them, each layer simply combines feature maps obtained from preceding levels by concatenating them without considering the interdependencies between various channels. Second, the connection between the interlayer feature map and the feature map of the interlayer is not clearly shown. Adaptively learning the correlation coefficients between layers by modeling the correlation of feature maps between the layers is very beneficial [36]. There are m layers contained within each Dense Block, with each layer being connected in a feed-forward method to all the consecutive layers that come after it. If x_m is denoted as the output from the m^{th} layer, it is calculated using Equation (2):

$$x_m = H_m([x_1, x_2, \dots, x_{m-1}]), \qquad (2)$$

where H_m shows how the combination method might work in this layer. Within it, each individual feature layer is subjected to the processing of a concatenation function.

III. Methodology

Yolo V4 for the identification of musical instruments is described in the following sections. Figure 2 depicts an overview of system methods. The BBox mark tool [37] was adopted to make a bounding box for all object. The labeling procedure was carried out for each class and there may be multiple marks on one image. Following that, each class label is associated with a single training model. The bounding box labeling tool's return values are object coordinates (x_1, y_1, x_2, y_2). These coordinates of items are distinct from the input value of Yolo. Instead, the Yolo input value is the center point and width and height *(x, y, w, h)*. Consequently, the bounding box coordinates in the Yolo input format must be adjusted by the system because of the change. The modification process is based on Equations (3) – (8).

$$dw = 1/W \qquad (3)$$

$$x = \frac{(x_1 + x_2)}{2} \times dw \qquad (4)$$

$$dh = 1/H \qquad (5)$$

$$y = \frac{(y_1 + y_2)}{2} \times dh \qquad (6)$$

$$w = (x_2 - x_1) \times dw \qquad (7)$$

$$h = (y_2 - y_1) \times dh \qquad (8)$$

H denotes the image's height, dh denotes the image's absolute height, W denotes the image's width, and dw denotes the image's absolute width. W denotes the image's width, and dw denotes the image's absolute width. As a result, float values relative to the image's width and height (dw and dh) may be anywhere in range 0 to 1.

The following is the procedure for detecting objects using Yolo V4:

(1) Separates the image into $N \times N$ grids. Each grid generates a total of K bounding boxes by utilizing anchor box computation. It estimates B boundary boxes for every grid cell, along with a confidence score for each forecast boundary box.

(2) No matter how many boxes B were in the array, only one object was recognized. Additionally, it predicts the probability of C conditional classes and one for each class (for the probability of the object class occurring).

(3) The CNN layers used to obtain all features from the image and predicts the $b = [b_x, b_y, b_w, b_h, b_c]^T$ and the $class = [class_1, class_2, \dots, class_c]^T$.

(4) Estimates the optimum confidence IoU_{pred}^{truth} of the K bounding boxes with the threshold IoU_{thres}. If $IoU_{pred}^{truth} > IoU_{thres}$, meaning that the bounding box contains the object. The bounding box would not contain the object otherwise.

(5) The algorithm then selects the most probable category for the item's kind from the available choices. Non-Maximum Suppression (NMS) is used in our studies to conduct a maximum local search on drop boxes, redundant output, and object detection findings. (6) The last step produces an image that has been classified and labeled with the class. When it comes to detecting objects in an image, non-maximum suppression is an important step to take. NMS is a classic algorithm that analyzes detection candidates and only retains the best of them after thorough evaluation. However, because the computations are expensive, it is difficult to accelerate with traditional hardware architectures. This type of problem is addressed by the quadric architecture, which provides the performance required for resource-constrained edge performance.

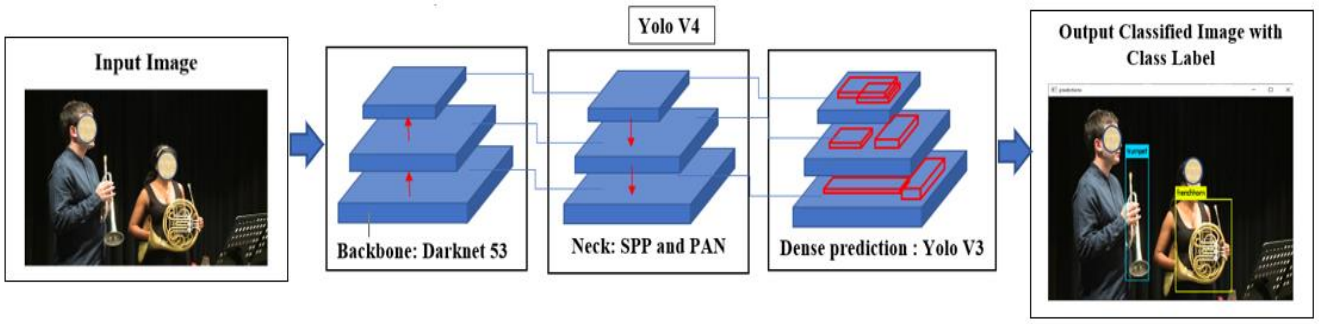


Figure 2. An overview of the system.

IV. Experiment Results

A. Dataset

The PPMI dataset includes photographs of people engaging with a variety of musical instruments. Bassoon, cello, clarinet, French horn, erhu, flute, guitar, harp, saxophone, trumpet, recorder, and violin are among the instruments in the dataset. B. Yao gathered images of musical instruments and published them in [12].

TABLE I
Musical Instrument Dataset.

Class Name	Total Image	Training	Testing
Bassoon	362	253	109
Cello	322	225	97
Clarinet	316	221	95
Erhu	337	236	101
Flute	315	221	95
French horn	327	229	98
Guitar	326	228	98
Harp	332	232	100
Recorder	309	216	93
Saxophone	326	228	98
Trumpet	330	231	99
Violin	340	238	102
Total Image	3942	2759	1183

Aditya Khosla gathered images of instruments such as the cello, clarinet, harp, recorder, and trumpet, which he then published in September 2010. Initially, the dataset included 100 pictures in each category for training and 100 images for testing. Table I contains a glimpse of the dataset. We trained and tested our models in this article using the PPMI dataset.

The collection contains images of individuals performing musical instruments from a variety of angles, positions, and backgrounds. The diversity of individuals who play musical instruments is determined by the way the instruments are performed. Using data augmentation techniques such as rotation and flip, we expand each category's dataset. The collection comprises of 309-462 photos per category. The total number of photos in our dataset is now 3942, including 2759 for training and 1183 for testing.

An Nvidia GTX2070 Super GPU accelerator, an AMD Ryzen 7 3700X Central Processing Unit (CPU) with an 8-core processor, and 32GB of DDR4-3200 memory were all components of the environment that was used to train the musical instrument recognition model.

B. Densenet and Yolo V4 Training Result

Our experiment enhances the Densenet and Yolo V4 configurations during the training phase by applying a learning rate of 0.001 to evaluate, a learning rate decay of 0.1 at every epoch, and a momentum learning rate of 0.9. Figure 3(a) indicates that the training process with Densenet is consistent. After 27000 epochs, the training stage stays the same and ends after 45000 epochs. Densenet uses *max batches = 45000*, mask scale = 1, and the training loss value reaches 0.0731. Also, Yolo V4 uses *learning rate = 0.0013*, burn in = 1000, *max batches = 24000*, *policy = steps*, *steps = 19200, 21600*, *scales = 0.1,0.1*, and the iteration is unsteady and goes up and down, ending at 24000 epochs with a loss value of 0.7576 in Figure 3. (b).

Moreover, training performance result shown in Table II. Yolo V4 achieve the loss value 0.758 with 60.11% *IoU* and 81.32% *mAP*. Thus, our Yolo V4 training model detected the objects with high accuracy. In other hand, Densenet exhibit 74.77% *mAP* with 50.69% *IoU*. In this study, IoU is utilized to determine the degree to which our projected border overlaps with the ground truth, which is the boundary of the real object being investigated. Yolo V4 exhibited a greater mean absolute performance than Densenet in almost all the testing groups.

IoU computes the overlap ratio between the prediction (pred) and ground-truth (gt) border boxes, as indicated in Equation (9)[38][39][40].

$$\boldsymbol{IoU} = \frac{\boldsymbol{Area_{pred} \cap Area_{gt}}}{\boldsymbol{Area_{pred} \cup Area_{gt}}} \tag{9}$$

Nonetheless, the output samples fall into three categories. True positive (TP) is the number of properly identified samples; false positive (FP) is the number of incorrectly recognized samples [41][42], true negative (TN) is the number of unrecognized samples. Precision and recall are described by [43][44] in Equation (10)-(11).

$$Precision\ (P) = \frac{TP}{TP+FP} \tag{10}$$

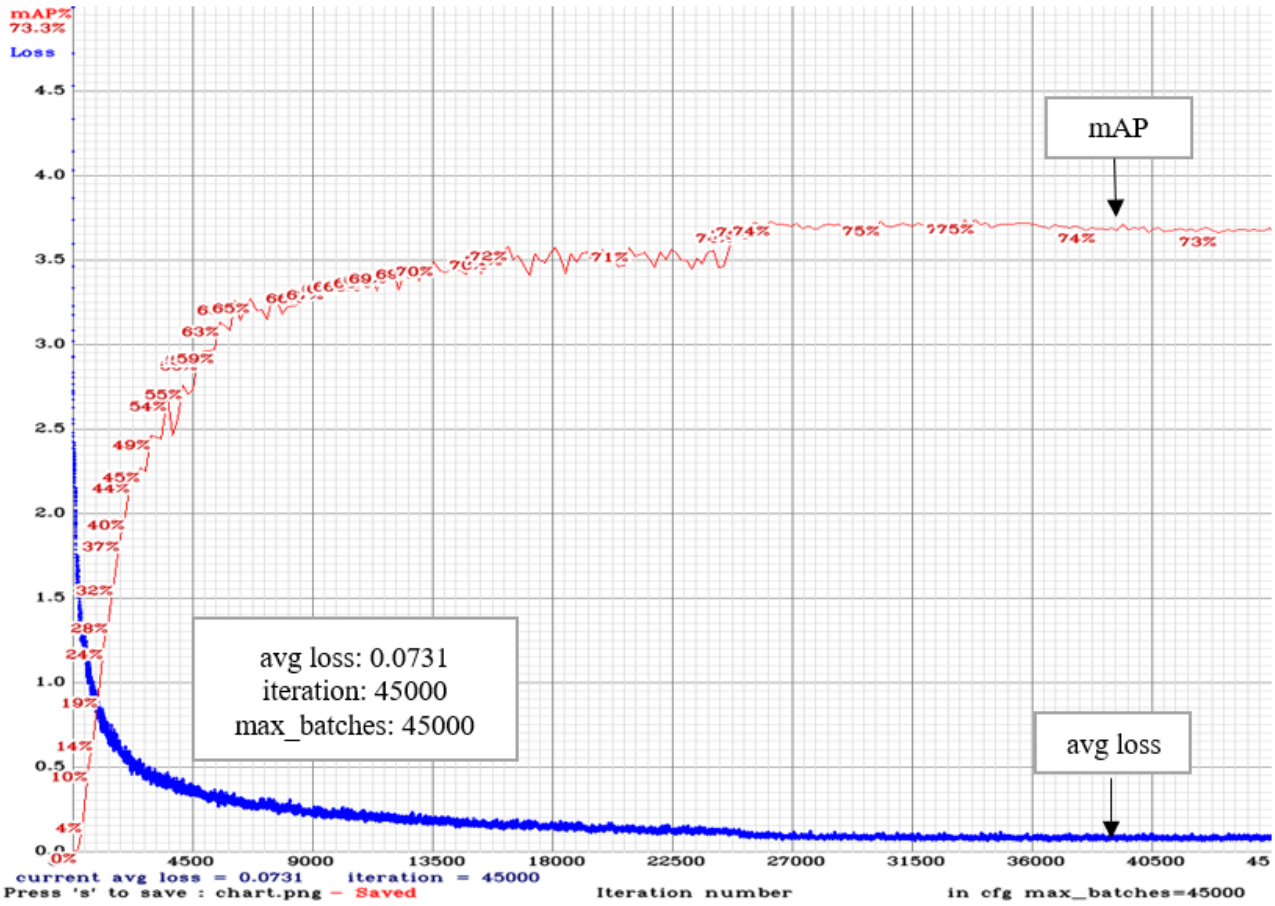
$$Recall\ (R) = \frac{TP}{TP+FN} \tag{11}$$

Moreover, F1 is shown in Equation (12) [45][46][47].

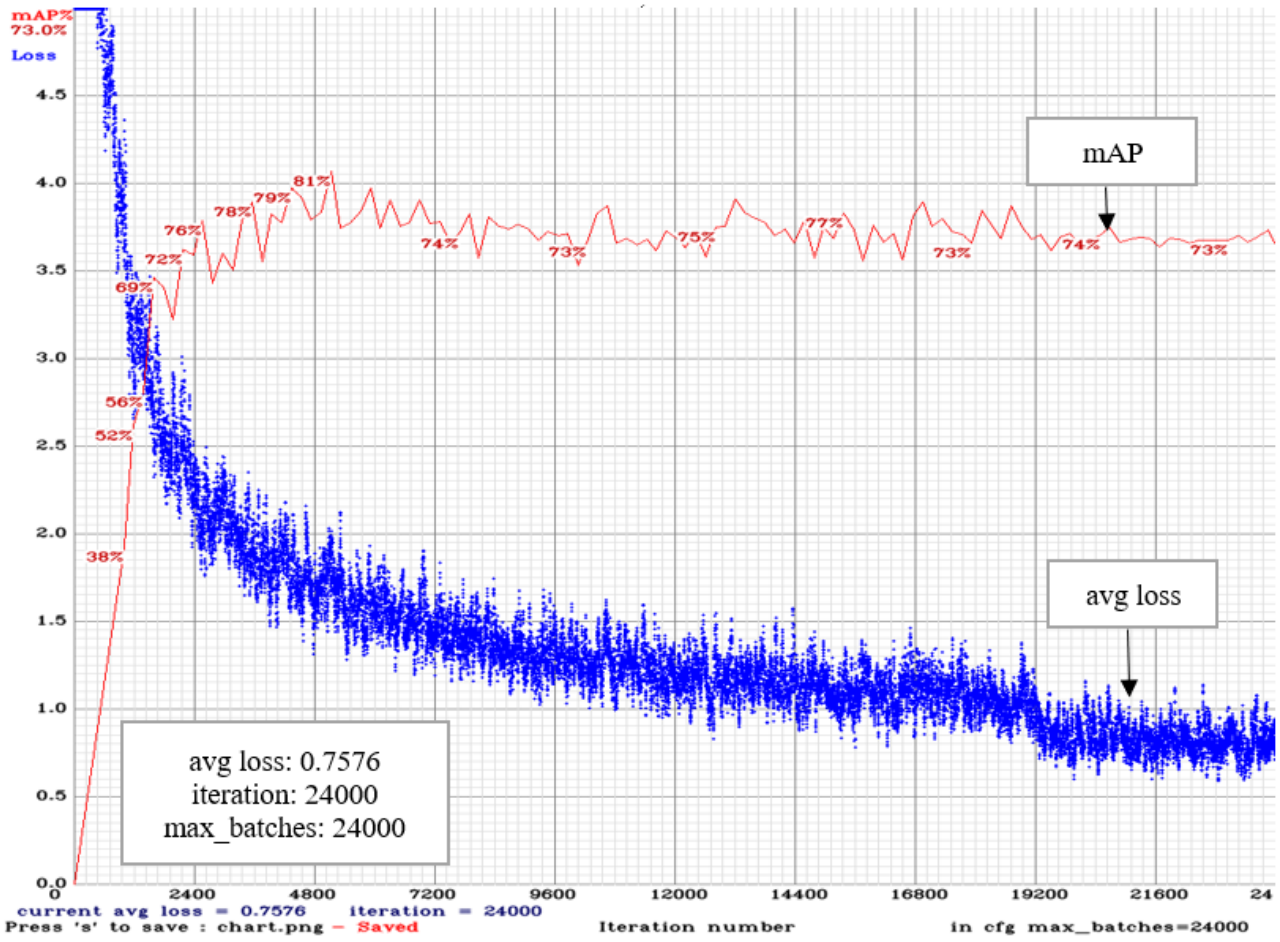
$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

TABLE II
TRAINING PERFORMANCE RESULTS.

Model	Loss Value	Class Name	Class ID	AP (%)	TP	FP	Precision	Recall	F1-score	IoU (%)	mAP@0.50 (%)
Yolo V4	0.758	Bassoon	0	77.59	63	19	0.77	0.78	0.78	60.11	81.32
		Cello	1	77.85	56	13					
		Clarinet	2	69.73	58	40					
		Erhu	3	86.45	71	21					
		Flute	4	71.22	68	26					
		French horn	5	83.45	66	9					
		Guitar	6	90.55	63	6					
		Harp	7	98.09	66	5					
		Recorder	8	71.32	82	29					
		Saxophone	9	88.92	71	21					
		Trumpet	10	77.42	59	23					
		Violin	11	83.25	78	26					
Densenet	0.073	Bassoon	0	72.25	57	28	0.67	0.74	0.7	50.69	74.77
		Cello	1	79.19	63	22					
		Clarinet	2	66.03	56	47					
		Erhu	3	78.36	61	23					
		Flute	4	75.22	68	43					
		French horn	5	82.26	75	27					
		Guitar	6	79.46	57	14					
		Harp	7	93.95	63	11					
		Recorder	8	52.82	63	58					
		Saxophone	9	83.05	69	26					
		Trumpet	10	61.73	51	31					
		Violin	11	69.87	78	46					



(a)



(b)

Figure 3. Training result using (a) Densenet and (b) Yolo V4.

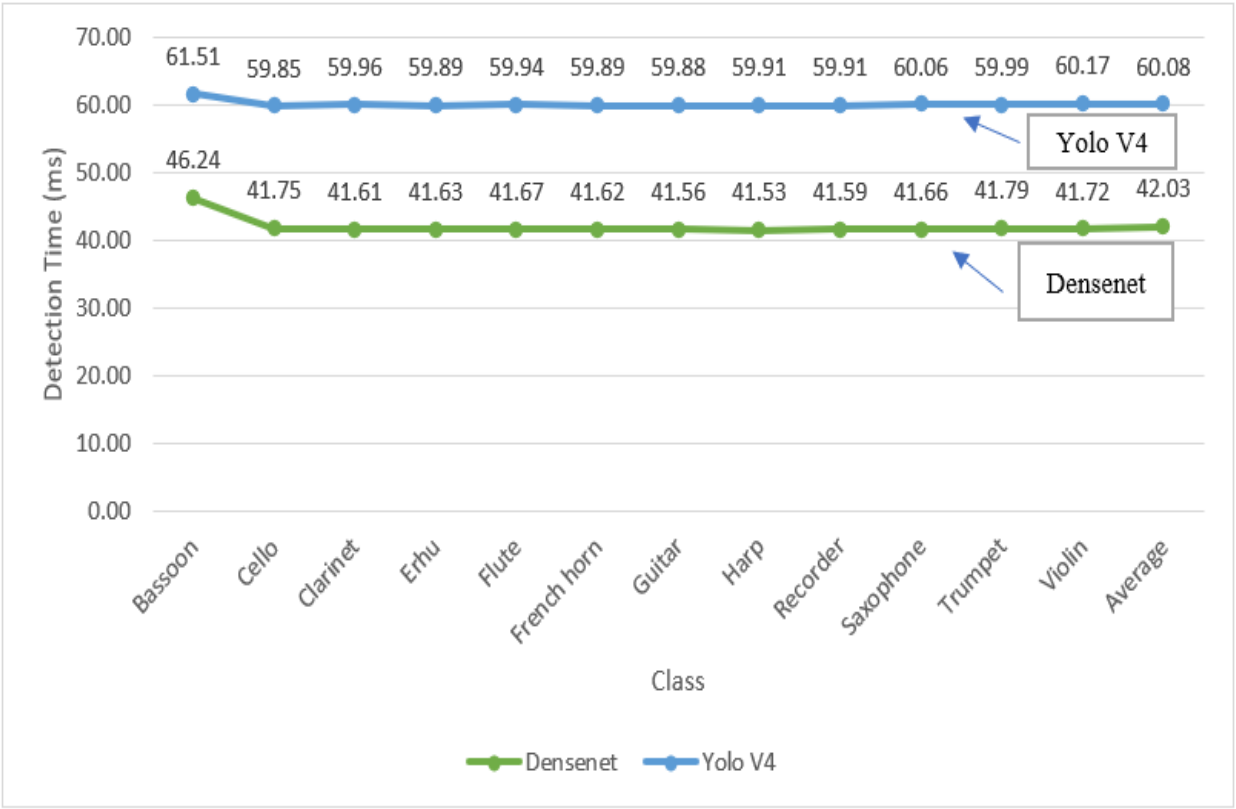


Figure 4. Result of comparison of detection time (milliseconds).

TABLE III
TESTING ACCURACY RESULTS PERFORMANCE COMPARISON

Class Name	Class ID	DCYolo-GAN [23]	Yolo V2 [23]	Grouplet [12]	Resnet 50 SPP [48]	Densenet	Yolo V4
Bassoon	0	82%	82%	78.50%	85%	96%	95%
Cello	1	90%	86%	87.60%	81%	95%	92%
Clarinet	2	93%	92%	95.70%	89%	90%	93%
Erhu	3	98%	98%	84.00%	81%	77%	95%
Flute	4	89%	86%	87.70%	82%	91%	91%
French horn	5	92%	92%	87.70%	78%	79%	96%
Guitar	6	98%	96%	93.00%	79%	94%	95%
Harp	7	100%	100%	76.30%	98%	94%	96%
Recorder	8	84%	84%	84.60%	85%	79%	89%
Saxophone	9	98%	98%	82.30%	93%	89%	99%
Trumpet	10	90%	90%	87.10%	85%	88%	99%
Violin	11	98%	96%	76.50%	80%	74%	96%
	Average	92.67%	91.67%	85.10%	84.64%	87%	94.70%

In addition, Equations (13) [18] shows the Yolo loss function.

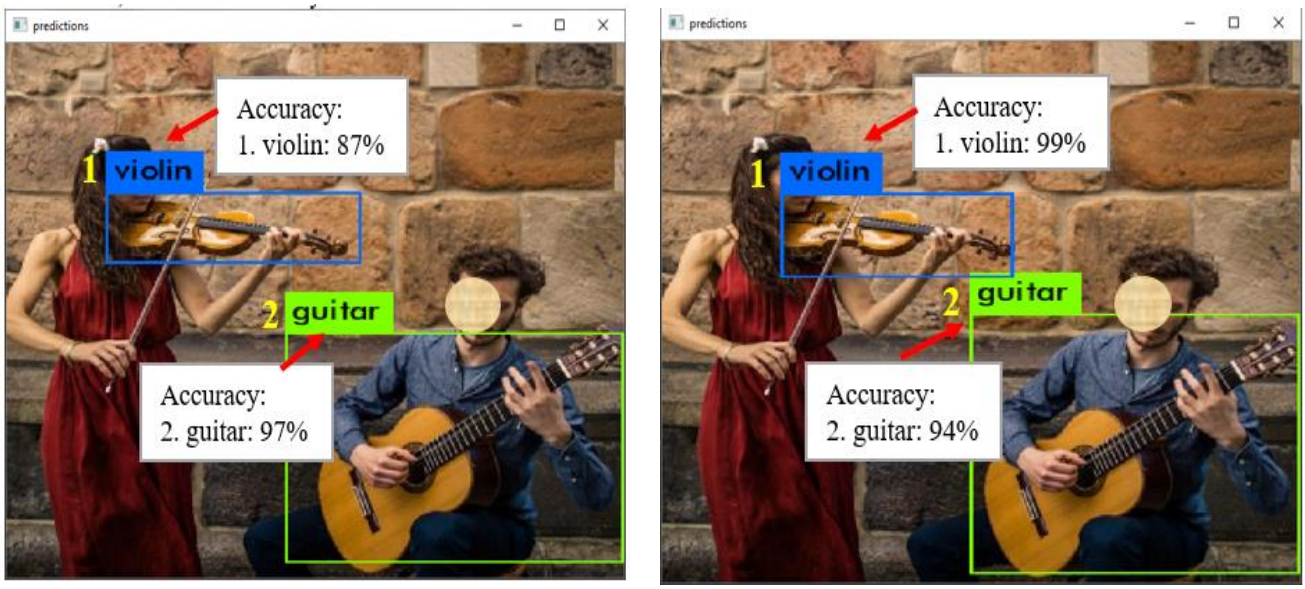
$$\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\mathbb{I}_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y-\hat{y}_i)^2]$$
$$+\boldsymbol{\lambda_{coord}}\sum_{i=0}^{s^2}\sum_{j=0}^{B}\mathbb{I}_{ij}^{\boldsymbol{obj}}\left[\left(\sqrt{\boldsymbol{w_i}}-\sqrt{\widehat{\boldsymbol{w}}_i}\right)^2+\left(\sqrt{\boldsymbol{h_i}}-\sqrt{\widehat{\boldsymbol{h}}_i}\right)^2\right]$$
$$+\sum_{i=0}^{s^2}\sum_{j=0}^{B}\mathbb{I}_{ij}^{obj}\left(C_i-\hat{C}_i\right)^2$$
$$+\sum_{i=0}^{s^2}\mathbb{I}_{i}^{obj}\sum_{c\epsilon classes}(p_i(c)-\hat{p}_i(c))^2 \quad (13)$$

where $\mathbb{I}_{ij}^{\boldsymbol{obj}}$ indicates if the object is present in cell *i*, and $\mathbb{I}_{ij}^{\boldsymbol{obj}}$ denotes that the $\boldsymbol{j^{th}}$bounding box predictor in cell *i* is responsible for the prediction. Next, $(\hat{\boldsymbol{x}},\hat{\boldsymbol{y}},\hat{\boldsymbol{w}},\hat{\boldsymbol{h}},\hat{\boldsymbol{c}},\hat{\boldsymbol{p}})$ are represented as the center coordinates, width, height, confidence, and category probability of the predicted bounding box. Those symbols without the cusp are true labels. Furthermore, our works set the $\boldsymbol{\lambda coord}$ to 0.5, indicating that the width and height errors are less effective in the calculation. Then, $\boldsymbol{\lambda noobj}$ = 0.5 is used to reduce the impact of many empty grids on the loss value.

V. DISCUSSIONS

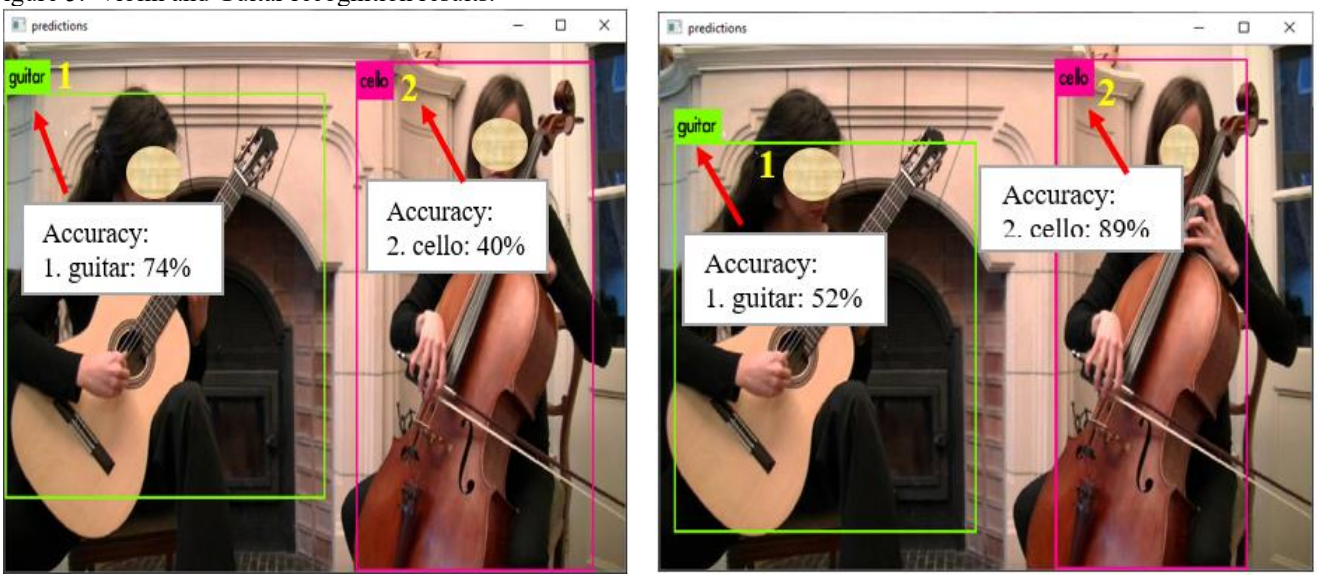
Table III shows the results of tests performed using images other than those in our dataset and the accuracy of the findings obtained. Overall, Yolo V4 is more precise than the previous version. Yolo V4 increases the accuracy of previous method in all class except for Clarinet, Erhu, Guitar, Harp, and Violin. Moreover, Saxophone and Trumpet leading the highest accuracy 99% for Yolo V4. Followed by French horn 96%, Basson 95%, Cello 92%, Flute 91%, and Recorder 89%. The optimum total average accuracy obtained by Yolo V4 with 94.70% accuracy. Next, DCYolo-GAN [23] exhibits 92.67%, Yolo V2 [23] 91.67%, Densenet 87% and Grouplet [12] gains 85.10% accuracy. Harp instrument obtained the highest accuracy 100% by using DCYolo-GAN and Yolo V2 [23]. Also, Clarinet musical instrument exhibits the maximum accuracy 95.70% employing Grouplet [12]. Figure 4 describes the comparison results between Yolo V4 and Densenet in terms of detection time. The average detection time for the Yolo V4 is 60.08 milliseconds. Furthermore, the average detection time for Densenet is 42.03 milliseconds. In terms of detection time, it can be determined that Densenet is faster than Yolo V4.

The clarinet and the flute are two wind instruments that are quite comparable to one another in terms of their appearance, the way they are played, and the dimensions of the instruments. Comparable musical instruments include the guitar, violin, and cello. Guitar, violin, and cello are all stringed instruments. Although these three musical instruments are comparable to one another in terms of color, shape, and size, their dimensions couldn't be more dissimilar. The violin is the smallest instrument, the guitar is in the middle, and the cello is the biggest. Figure 5 shows the recognition result of guitar and cello. Densenet gave the workspace more space of 104.86 MB, the total number of BFLOPS was 31.883, and 306 layers were loaded from the weights-file. Image 4.jpg is predicted in 49.242 milliseconds with the result violin obtains 87%, and guitar 97% accuracy shown in Figure 5(a). Furthermore, recognition result of Yolo V4 by using the same image describe in Figure 5(b). Image 4.jpg is predicted in 67.740000 milliseconds as a result the violin gains 99% and the guitar 94% accuracy. Yolo V4 loaded 162 layers from weights-file and required additional workspace size 52.43 MB and total BFLOPS 59.634. Yolo V4 gets the highest average accuracy across all violin and guitar classes, although it requires more time to recognize objects in an image. Yolo V4 exhibits the optimum total BFLOPS 59.643 compared to Densenet.



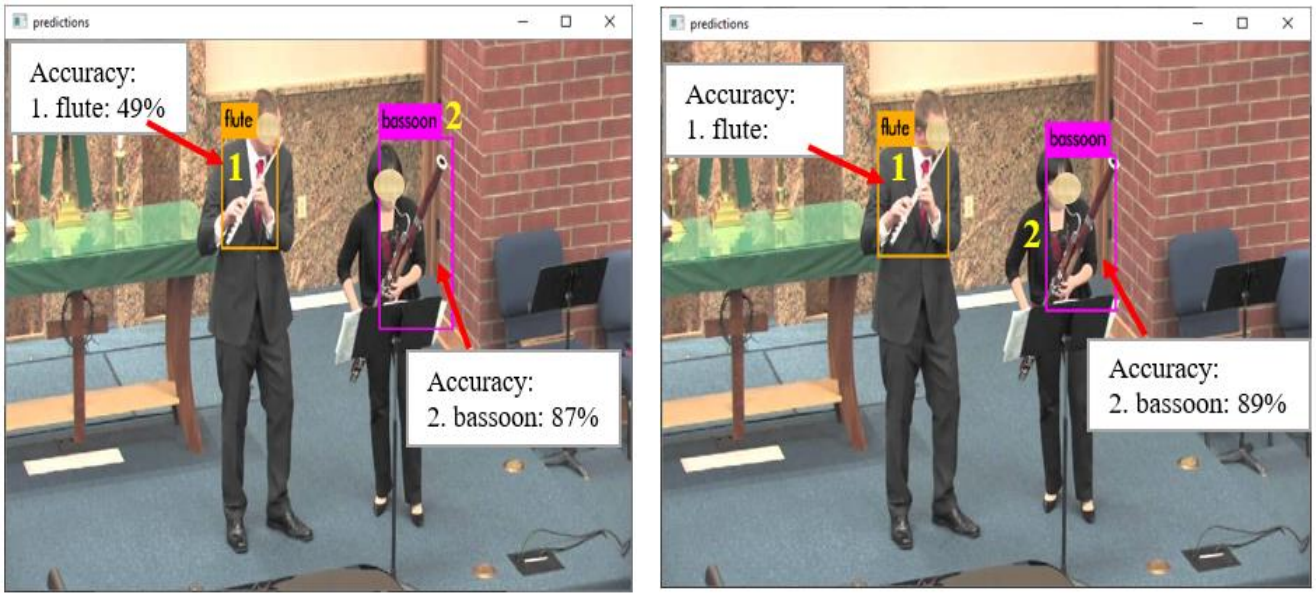
(a) Densenet (b) Yolo V4

Figure 5. Violin and Guitar recognition results.



(a) Densenet (b) Yolo V4

Figure 6. Guitar and Cello recognition result.



(a) Densenet (b) Yolo V4

Figure 7. Flute and Bassoon recognition result.

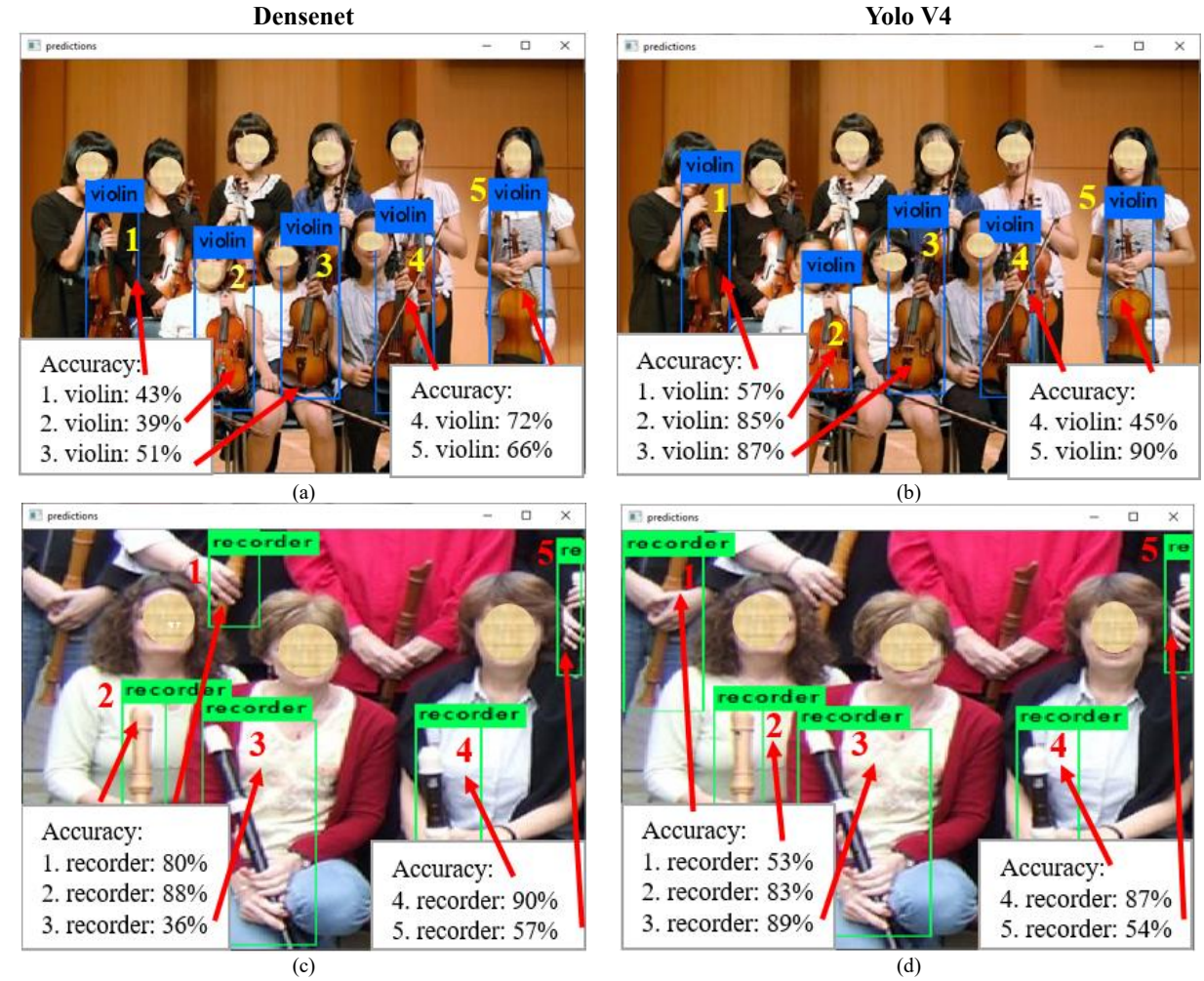


Figure 8. Violin and Recorder recognition result.

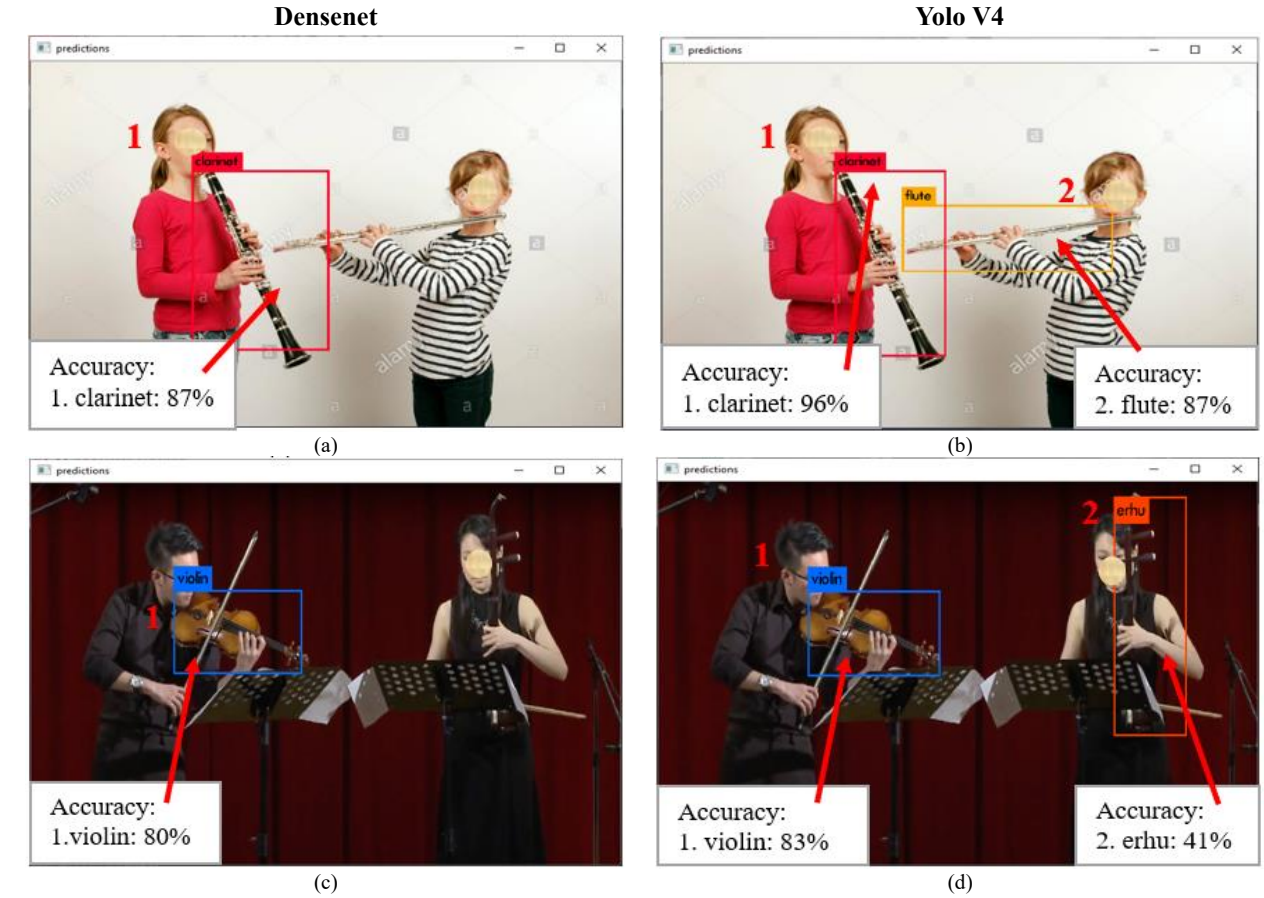


Figure 9. The missed detection results.

Moreover, Figure 6(a) describes the recognition result of guitar and cello using Densenet. After load 306 layers from weights-file, Images 5.jpg is predicted in 49.173 milli-seconds. As a result, guitar achieves 74% and cello 40%accuracy. In the other hand, Yolo V4 predicted guitar 52% and cello 89% in 68.222 milli-seconds that shown in Figure 6(b). Flute and bassoon recognition result example describes in Figure 7. The two instruments are quite similar in shape and size. Flute obtained 49% accuracy and bassoon 87% using Densenet with prediction time 48.632 milli-seconds and it is presented in Figure 7(a). Yolo V4 predicts flute 64% and bassoon 89% accuracy in 67,876 milliseconds as shown in Figure 7(b).

As a consequence of the test results in Figure 5, Figure 6, and Figure 7, it can be concluded that each model can correctly identify all classes with coordinate ranges and bounding box accuracy. The recognition result of violin and recorder with multiple objects could be seen in Figure 8. The optimum accuracy is obtained by Yolo V4 in Figure 8(b). Yolo V4 can recognize 5 recorders in the image with the accuracy 53%, 83%, 89%, 87%, and 54%, successively. The violin9.jpg image is predicted in 67.898000 milliseconds as shown in Figure 8(b) and can recognize 5 violins with 57%, 85%, 87%, 45%, and 90% accuracy, respectively. However, Figure 9 describes the missed detection results. In addition, Densenet shows missed detections in Figure 9(a) and Figure 9(c). Furthermore, Densenet can only detect the clarinet with 87% accuracy in 48,521 milliseconds as shown in Figure 9(a). Yolo V4 can detect all musical instruments in Figure 9(b) and Figure 9(d). Yolo V4 can detect guitar 52% accuracy and cello 89% accuracy as seen on Figure 9(b).

Some benefits of Yolo V4 is as follows: (1) Yolo V4 is not only an effective and potent object detection model, but it also makes it possible for anyone with a graphics processing unit (GPU) that is either a 1080 Ti or a 2080 Ti to train a super-fast and accurate object detector. (2) Verification has been performed on the influence of the latest object detection methods, including the "Bag-of-Freebies" and the "Bag-of-Specials," on the training of detectors. (3) All of the state-of-the-art methods, such as Cross-iteration batch normalization (CBN) [49], Path aggregation network (PAN) [50] , and others, have been improved to be more efficient for single general processing unit (GPU) [51] training.

VI. Conclusions

In this study, the primary focus is on how we attempted to differentiate between things that appear extremely like the human eye. Densenet and Yolo V4 are used to identify musical instruments in our investigations. In this research, we recognized several similar music instruments. Our work examines CNN models combined with various backbone architectures and extractor features, including the Densenet and Yolo V4, for object recognition. In this experiment, the key features of the detector are investigated. Moreover, we were able to increase the performance of the detection of similar music instruments based on the results of our experiment. Yolo V4 showed the maximum average accuracy of 94.70% compared to the previous results, Grouplet [12] only achieves accuracy 85.10 %, DC-Yolo GAN [23] 92.67 %, and Yolo V2 91.67%.

As part of our future research, we would like to identify an incorrectly shaped musical instrument in an image. In addition, we are incorporating Explainable Artificial Intelligence (XAI) in our future research to shed more light on the image.

References

[1] A. C. M. Ribeiro, R. C. Scharlach, and M. M. C. Pinheiro, “Assessment of temporal aspects in popular singers,” *CODAS*, vol. 27, no. 6, 2015, doi: 10.1590/2317-1782/20152014234.

[2] Y. Lavinia, H. Vo, and A. Verma, “New colour fusion deep learning model for large-scale action recognition,” *Int. J. Comput. Vis. Robot.*, vol. 10, no. 1, pp. 41–60, 2020, doi: 10.1504/IJCVR.2020.104356.

[3] J. Wetzel, A. Laubenheimer, and M. Heizmann, “Joint Probabilistic People Detection in Overlapping Depth Images,” *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2972055.

[4] S. Saponara, A. Elhanashi, and A. Gagliardi, “Implementing a real-time, AI-based, people detection and social distancing measuring system for Covid-19,” *J. Real-Time Image Process.*, 2021, doi: 10.1007/s11554-021-01070-6.

[5] T. Bai *et al.*, “An optimized faster R-CNN method based on DRNet and RoI align for building detection in remote sensing images,” *Remote Sens.*, vol. 12, no. 5, 2020, doi: 10.3390/rs12050762.

[6] C. Dewi, R. C. Chen, and H. Yu, “Weight analysis for various prohibitory sign detection and recognition using deep learning,” *Multimed. Tools Appl.*, vol. 79, no. 43–44, pp. 32897–32915, 2020, doi: 10.1007/s11042-020-09509-x.

[7] C. Dewi, R.-C. Chen, and S.-K. Tai, “Evaluation of Robust Spatial Pyramid Pooling Based on Convolutional Neural Network for Traffic Sign Recognition System,” *Electronics*, vol. 9, no. 6, p. 889, 2020, doi: 10.3390/electronics9060889.

[8] X. Xi, Z. Yu, Z. Zhan, Y. Yin, and C. Tian, “Multi-Task Cost-Sensitive-Convolutional Neural Network for Car Detection,” *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2927866.

[9] S. Qin and S. Liu, “Towards end-to-end car license plate location and recognition in unconstrained scenarios,” *Neural Comput. Appl.*, 2021, doi: 10.1007/s00521-021-06147-8.

[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” *arXiv:2004.10934*, pp. 1–17, 2020, [Online]. Available: https://arxiv.org/abs/2004.10934.

[11] D. L. Yuan and Y. Xu, “Lightweight Vehicle Detection Algorithm Based on Improved YOLOv4,” *Eng. Lett.*, vol. 29, no. 4, pp. 1544–1551, Dec. 2021.

[12] B. Yao and L. Fei-Fei, “Grouplet: A structured image representation for recognizing human and object interactions,” 2010, doi: 10.1109/CVPR.2010.5540234.

[13] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[14] S.-K. Tai, C. Dewi, R.-C. Chen, Y.-T. Liu, X. Jiang, and H. Yu, “Deep learning for traffic sign recognition based on spatial pyramid pooling with scale analysis,” *Appl. Sci.*, vol. 10, no. 19, p. 6997, 2020, doi: 10.3390/app10196997.

[15] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[16] G. Cheng, Y. Si, H. Hong, X. Yao, and L. Guo, "Cross-Scale Feature Fusion for Object Detection in Optical Remote Sensing Images," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, 2021, doi: 10.1109/LGRS.2020.2975541.

[17] C. Dewi, R.-C. Chen, X. Jiang, and H. Yu, "Deep convolutional neural network for enhancing traffic sign recognition developed on Yolo V4," *Multimed. Tools Appl.*, 2022, doi: https://doi.org/10.1007/s11042-022-12962-5.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[19] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[20] S. Takezaki and K. Kishida, "Construction of CNNs for Abnormal Heart Sound Detection using Data Augmentation," Oct. 2021.

[21] S. Song, Z. Que, J. Hou, S. Du, and Y. Song, "An efficient convolutional neural network for small traffic sign detection," *J. Syst. Archit.*, no. 97, pp. 269–277, 2019, doi: 10.1016/j.sysarc.2019.01.012.

[22] M. Ju, S. Moon, and C. D. Yoo, "Object Detection for Similar Appearance Objects Based on Entropy," 2019, doi: 10.1109/RITAPP.2019.8932791.

[23] C. Dewi, R. C. Chen, Hendry, and Y. T. Liu, "Similar Music Instrument Detection via Deep Convolution YOLO-Generative Adversarial Network," 2019, doi: 10.1109/ICAwST.2019.8923404.

[24] C. Wang, H. M. Liao, Y. Wu, and P. Chen, "CSPNet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPR Workshop)*, 2020, p. 2.

[25] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *CoRR*, vol. abs/1804.0, pp. 1–6, 2018, [Online]. Available: http://arxiv.org/abs/1804.02767.

[26] D. Misra, "Mish: A self regularized non-monotonic neural activation function," *arXiv*, pp. 1–14, 2019.

[27] Q. Liu and S. Furber, "Noisy softplus: A biology inspired activation function," in *Lecture Notes in Computer Science*, 2016, pp. 405–412, doi: 10.1007/978-3-319-46681-1_49.

[28] D. Benyang, L. Xiaochun, and Y. Miao, "Safety helmet detection method based on YOLO v4," 2020, doi: 10.1109/CIS52066.2020.00041.

[29] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of Freebies for Training Object Detection Neural Networks," *arXiv:1902.04103v3*, pp. 1–9, 2019.

[30] H. Chen, Z. He, B. Shi, and T. Zhong, "Research on Recognition Method of Electrical Components Based on YOLO V3," *IEEE Access*, vol. 7, pp. 157818–157829, 2019, doi: 10.1109/ACCESS.2019.2950053.

[31] Liquan Zhao and S. Li, "Object Detection Algorithm Based on Improved YOLOv3," *Electronics*, vol. 9, no. 3, p. 537, 2020.

[32] N. Ghatwary, X. Ye, and M. Zolgharni, "Esophageal Abnormality Detection Using DenseNet Based Faster R-CNN with Gabor Features," *IEEE Access*, vol. 7, pp. 84374–84385, 2019, doi: 10.1109/ACCESS.2019.2925585.

[33] C. Dewi, R. C. Chen, Y. T. Liu, X. Jiang, and K. D. Hartomo, "Yolo V4 for Advanced Traffic Sign Recognition with Synthetic Training Data Generated by Various GAN," *IEEE Access*, vol. 9, pp. 97228–97242, 2021, doi: 10.1109/ACCESS.2021.3094201.

[34] S. Zhai, D. Shang, S. Wang, and S. Dong, "DF-SSD: An Improved SSD Object Detection Algorithm Based on DenseNet and Feature Fusion," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2971026.

[35] T. Li, W. Jiao, L. N. Wang, and G. Zhong, "Automatic DenseNet sparsification," *IEEE Access*, vol. 8, pp. 62561–62571, 2020, doi: 10.1109/ACCESS.2020.2984130.

[36] C. Dewi, R.-C. Chen, Y.-T. Liu, and S.-K. Tai, "Synthetic Data generation using DCGAN for improved traffic sign recognition," *Neural Comput. Appl.*, vol. 33, no. 8, pp. 1–15, 2021, [Online]. Available: https://doi.org/10.1007/s00521-021-05982-z.

[37] "Bbox label tool," 2019. https://github.com/puzzledqs/BBox-Label-Tool.

[38] Á. Arcos-García, J. A. Álvarez-García, and L. M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems," *Neurocomputing*, vol. 316, pp. 332–344, 2018, doi: 10.1016/j.neucom.2018.08.009.

[39] C. Dewi, R. C. Chen, Hendry, and H. Te Hung, "Comparative Analysis of Restricted Boltzmann Machine Models for Image Classification," in *Asian Conference on Intelligent Information and Database Systems ACIIDS 2020*, 2020, vol. 12034 LNAI, pp. 285–296, doi: 10.1007/978-3-030-42058-1_24.

[40] C. Dewi and R.-C. Chen, "Random Forest and Support Vector Machine on Features Selection for Regression Analysis," *Int. J. Innov. Comput. Inf. Control*, vol. 15, no. 6, pp. 2027–2038, 2019.

[41] R. C. Chen, C. Dewi, S. W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *J. Big Data*, vol. 7, no. 52, pp. 1–26, 2020, doi: 10.1186/s40537-020-00327-4.

[42] B. Sun, W. Li, H. Liu, J. Yan, S. Gao, and P. Feng, "Obstacle detection of intelligent vehicle based on fusion of lidar and machine vision," *Eng. Lett.*, vol. 29, no. 2, 2021.

[43] H. Yang *et al.*, "Tender Tea Shoots Recognition and Positioning for Picking Robot Using Improved YOLO-V3 Model," *IEEE Access*, vol. 7, pp. 180998–181011, 2019, doi: 10.1109/ACCESS.2019.2958614.

[44] Y. Yuan, Z. Xiong, and Q. Wang, "An Incremental Framework for Video-Based Traffic Sign Detection, Tracking, and Recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1918–1929, 2017, doi: 10.1109/TITS.2016.2614548.

[45] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model,"

Comput. Electron. Agric., vol. 157, pp. 417–426, 2019, doi: 10.1016/j.compag.2019.01.012.

[46] R. Shi, T. Li, and Y. Yamaguchi, "An attribution-based pruning method for real-time mango detection with YOLO network," *Comput. Electron. Agric.*, no. 169, pp. 1–11, 2020, doi: 10.1016/j.compag.2020.105214.

[47] H. Kang and C. Chen, "Fast implementation of real-time fruit detection in apple orchards using deep learning," *Comput. Electron. Agric.*, vol. 168, pp. 1–10, 2020, doi: 10.1016/j.compag.2019.105108.

[48] C. Dewi and R.-C. Chen, "Combination of Resnet and Spatial Pyramid Pooling for Musical Instrument Identification," *Cybern. Inf. Technol.*, vol. 22, no. 1, p. 104, 2022, [Online]. Available: https://cit.iict.bas.bg/CIT-2022/v-22-1/10341-Volume22_Issue_1-07_paper.pdf.

[49] Z. Yao, Y. Cao, S. Zheng, G. Huang, and S. Lin, "Cross-iteration batch normalization," 2021, doi: 10.1109/CVPR46437.2021.01215.

[50] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[51] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "GPU computing," *Proc. IEEE*, vol. 96, no. 5, 2008, doi: 10.1109/JPROC.2008.917757.