DOI: 10.14421/ijid.2022.3766
IJID (International Journal on Informatics for Development), *e-ISSN: 2549-7448*
Vol. 11, No.2, 2022, Pp. 242-251

Real-time Smartphone Usage Surveillance System Based on YOLOv5

Rr. Hajar Puji Sejati
Faculty of Science & Technology
University of Technology Yogyakarta
Yogyakarta, Indonesia
hajarsejati@gmail.com

Rodhiyah Mardhiyyah
Faculty of Science & Technology
University of Technology Yogyakarta
Yogyakarta, Indonesia
rodhiyah.office@gmail.com

Zulkhairi
Faculty of Information Technology
University of Nahdlatul Ulama Yogyakarta
Yogyakarta, Indonesia
zulkhairi.jr@unu-jogja.ac.id

Nur Istiqomah
Faculty of Science & Technology
University of Technology Yogyakarta
Yogyakarta, Indonesia
istiqomahn390@gmail.com

R. Imam Budi Prasetya
Faculty of Science & Technology
University of Technology Yogyakarta
Yogyakarta, Indonesia
prasmanan0@gmail.com

Article History
Received November 24th, 2022
Revised January 13th, 2023
Accepted January 31st, 2023
Published February 2023

***Abstract*—The digital era affects students' attitudes toward utilizing applications as learning media. This phenomenon can be used to boost student achievement, but it can also have negative consequences, such as chatting while studying or cheating on school exams. To support the positive and reduce the negative impact of smartphone use, it is necessary to supervise this activity. The supervision can be done by utilizing a camera to detect a smartphone. The YOLOv5 algorithm was used, which is known for its good speed and accuracy in object detection. This smartphone detection system can be controlled, so the application is adjustable to the needs of learning activities. Collecting a dataset, annotating, training objects, writing program code, and testing the system are all stages in the development of this system. The dataset used in this research consists of 1,038 smartphone images from the internet and camera-captured images. This detection system was built to assist teachers in monitoring the use of smartphones by students. The results of this model training are 77.7% mean average precision, 93.2% precision rate, and 71.7% recall rate under varying lighting conditions.**

Keywords—digital literacy; smartphone detection; YOLOv5 algorithm; learning media; students' attitude

1 INTRODUCTION

During the COVID-19 pandemic, the shift to online learning has had an impact on the intensity of students' reading activity in their daily lives. In the world of education, reading is a central part of the learning process that cannot be denied. Reading is a cognitive process that uses reading material and requires understanding to know the meaning of a subject. Embedding a culture of literacy in students' habits can affect their level of future success in the long run. Lack of literacy in children at a young age is a serious threat to their future [1].

Digital literacy is a person's ability to interact, communicate, and seek information in his life through existing digital media interactions. Digital literacy can also be expressed as the ability to use and utilize digital tools such as computers, laptops, and cell phones to obtain or convey information [2].

Schools can implement learning by utilizing smartphones as learning media in class to increase digital literacy. Its use positively impacts students by making it easier for them to find something new and learn it. However, the existence of smartphones also has negative impacts, such as chatting while studying or cheating during school exams [3].

One solution to problems related to the negative impacts of smartphone usage in class is some degree of supervision [4]. Supervised smartphone use can take advantage of artificial intelligence (AI). AI is a branch of computer science that has the ability to learn, collect, solve problems, memorize, acquire knowledge, and understand human language in a natural fashion [5]. When an AI is created, it can be installed into a machine or computer to perform a specific task in the same way that a human can [6]. One of its applications is to utilize a device that can detect smartphone use in class, especially in learning sessions where the class teacher does not allow the use of smartphones. This device can be deactivated if the teacher allows the use of smartphones in their class.

The device built uses a webcam that functions to detect the presence of smartphones used by students in class during learning sessions. The device program was built using the Python programming language. Python is a high-level object-oriented programming language (OOP) and supports multi-platform operating systems. The development of the Python language is becoming increasingly popular around the world, and Python programmers have created many additional libraries for various purposes. The reason for its popularity is that Python is open source, which means that the code is open to the public so that everyone can take it, modify it, and contribute to developing this language [7] [8]. The detection method in this study uses the YOLO algorithm. Yolo's algorithm is well-known for its speed and accuracy in object detection [9].

This study aims to design a smartphone detection device for the classroom that can be activated and deactivated accordingly. The purpose of the ability to turn off the system is so that the teachers can use smartphones as a learning medium and students can gain a broader understanding of digital literacy. Conversely, when the system is activated, teachers can monitor students' smartphone usage activities if the learning session does not allow the use of smartphones through the monitor screen. The monitor is used to display the smartphone detection results along with the object name that appears in the bounding box. This detection system can assist the teacher in improving student attention, minimizing distraction from other smartphone features that are not related to the learning session, and eventually promoting digital literacy activities.

Research on smartphone usage has been discussed in order to summarize the findings of published studies that testify to the detrimental effects of smartphone use on students. Some of these studies discover that the advancement of smartphone technology can aid students in finding better data or information related to learning materials. However, smartphone technology has another undesired consequence that can have a negative impact on student learning success. This situation is caused by the misuse of smartphones by students and will likely decrease student learning success [10]. Another study identified a concerning level of nomophobia and smartphone addiction among nursing students that caused stress and anxiety and adversely affected sleep, learning, and academic performance [11]. Similar research also indicates that research is being conducted to investigate the current impact of smartphones on students. This research utilizes data from the Seoul Educational Longitudinal Study, specifically the 9th-year panel data, to analyze the relationship between the age at which students first started using smartphones and the duration of their addiction to them. Additionally, the study looks into the relationship between smartphone addiction and the student's independent learning ability and academic achievement scores [12].

Previous studies on object detection, such as the research that discusses advanced driver-assistance system (ADAS), aims to improve the driving experience for drivers. The ADAS technology helps to identify dangerous driving conditions that can lead to accidents. The study compares various state-of-the-art object detection algorithms such as YOLOv3, YOLOv4, and YOLOv5, using metrics like mean average precision (mAP) and frames per second (FPS) on benchmark datasets. The findings show that YOLOv5 outperforms the other algorithms, with a 95% accuracy and faster detection speed [13]. Fig. 1 illustrates a comparison of YOLOv5 with other state-of-the-art models on rural road datasets. As seen in the figures, YOLOv5 has a balanced performance when compared to the other models. However, it outperforms the other two algorithms in the comparison. The reason for this is that YOLOv5's primary focus is on increasing speed and accuracy, making it the best fit for the problem at hand.

In the research entitled "Implementation of a Convolutional Neural Network for Infant Pain Detection Through Facial Imagery with YOLO," a pain detection system was created in infants aged 0–12 months using the NVIDIA Jetson Nano Developer Kit as a tool to help detect pain in infants by using the Convolutional Neural Network (CNN) model using the PyTorch framework and the You Only Look Once (YOLO) algorithm with three classifications for detecting sad, neutral, and pain.

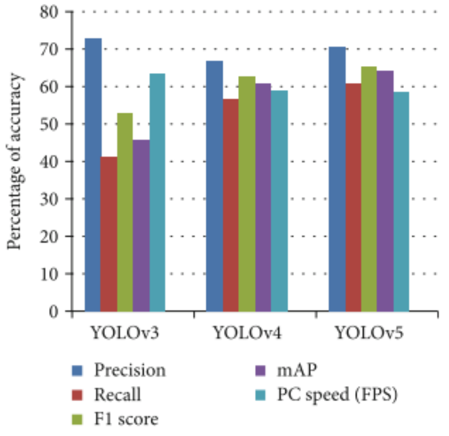

Figure 1. A comparison of the latest models for analyzing rural road datasets.

The researchers came to the conclusion that the YOLO algorithm was able to detect the three classifications with a sad mAP@0.5 value of 97.9%, a neutral value of 99.2%, a pain value of 96.9%, and a model accuracy of 70%. The results of the random data test and data from the Imogiri 1 Health Center obtained an accuracy value of 90% [14].

Some of these studies include a field project personnel detection system using the color and shape of helmets to detect the personnel rank as well as enforce employee access levels on production sites [15]. Another study used a highway road user detection system to determine vehicle density [16], capture still images from video frames to detect vehicle type [17], perform automatic license plate detection [9], and use a face detection system for employee attendance using the OpenCV library and the Yolo algorithm [18].

Research on smartphone detection mostly discusses object detection using smartphones as a detection medium. This is like designing a system that farmers could use as a scale pest detector for early prevention of pest damage. The developed system is combined with a smartphone device to help farmers improve their efficiency [19].

The author's contribution to this research is the use of smartphones as the object of detection. This is because the author has not found any research that discusses the use of smartphones as an object of detection. Research on the use of smartphones has been widely discussed in terms of both its positive and negative effects on students' learning outcomes.

The use of smartphones in object detection research is seen by the author as a medium to detect objects, not as research objects.

2 METHOD

The research object in these studies is a smartphone usage detection system for students whose learning in class utilizes the YOLO detection method. This algorithm was chosen because it has high accuracy [20] compared to the YOLOv4 algorithm. A comparison of the YOLOv4 and YOLOv5 algorithms has been carried out, which shows that YOLOv5 has the best performance [13].

To build this detection system, a specific device is needed for running the program. The devices used are the CPU as a processing center and a camera or webcam to capture smartphone objects. This system was built using the Python programming language, and the libraries used are Matplotib, Numpy, OpenCV-Python, Pillow, PyYAML, Request, Scipy, Torch, Torchvision, Tqdm, Protobuf, Tensorboard, Panas, and Seabord.

2.1 YOLOv5 Algorithm

Yolo is an algorithm used for real-time object detection. The YOLO architecture is similar to CNN. The working principle of YOLO is that it divides the input image into several boxes and predicts the bounding box and probability for each box. If an object's center falls within one of the cell boxes, then that box must detect the object. Each cell box predicts the bounding box and confidence score of each bounding box. The confidence score represents the percentage of certainty and accuracy in the model that there is an object in the box. There are five predictions in the bounding box, namely x, y, w, h, and confidence. The coordinates (x, y) represent the center of the box relative to the grid boundaries. The coordinates (w, h) or width and height represent the center of the box relative to the image. And the confidence represents the intersection of the predicted box and the ground truth box [21, 22]. It was first introduced in a research paper by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi in 2016. Yolo is an efficient object detection technique that utilizes a convolutional neural network, and it is particularly useful in applications that require real-time processing [23].

YOLOv5, which has undergone multiple revisions and enhancements. It was created by Ultralytics and is implemented using Pytorch [23]. YOLOv5 is an algorithm that enhances the speed at which objects are detected in real-time [13]. The YOLOv5 model consists of a backbone model, a neck model, and a head model. The backbone model is a feature extractor network that aims to extract features from the input image. The neck model has the main function of generating feature pyramids, which is an important block that allows the model to detect objects of diverse sizes and scales by building feature maps of varying scales. The head model generates predictions for bounding boxes and object classes [24].

The principle of the YOLOv5 algorithm is to detect objects or targets by assessing pixel blocks based on the colors and shapes that have been trained [25]. The YOLOv5 algorithm architecture is shown in Fig. 2.

The architecture of YOLOv5 consists of several components:

a. The backbone network: This is the feature extractor that is used to extract features from the input image. It typically consists of several convolutional layers and pooling layers.
b. The neck: The neck is a set of layers that are added on top of the backbone network to refine the features and make them more suitable for object detection.
c. The head: The head is part of the network that predicts the bounding boxes and class probabilities. It typically consists of several layers, including fully connected layers and a final output layer.

d. The anchor box layer: anchor boxes are predefined bounding boxes that are used to help the model make more accurate predictions. The anchor box layer is used to adjust the predicted bounding boxes to match the anchor boxes.
e. The SPADE layer: SPADE is a new architecture that allows the model to learn scale-aware features. It is used to predict object shapes and sizes more accurately.
f. The multi-scale layer is used to perform multi-scale inference, which is when the image is resized and processed multiple times at different scales to detect objects of various sizes.
g. The Non-Maximum Suppression (NMS): layer is used to eliminate overlapping boxes and select the most likely object in each grid cell.
h. The post-processing layer is used to filter the output to ensure that only the most likely object is detected.

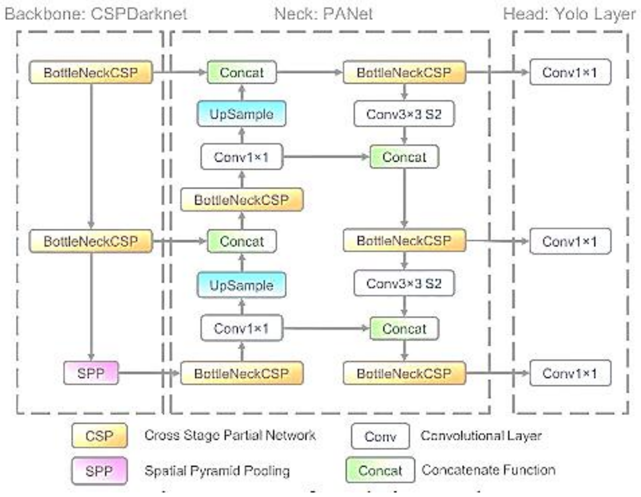

Figure 2. YOLOv5 algorithm architecture

YOLOv5 consists of several stages to perform object detection:

a. Preprocessing: The input image is resized and normalized to match the model requirements.
b. Feature extraction: The CNN extracts features from the image using convolutional layers and pooling layers. The output of this stage is a feature map.
c. Prediction: The feature map is then passed through several layers to predict the bounding boxes and class probabilities. The network predicts a grid of bounding boxes for each object, along with its class probabilities.
d. Non-maximum suppression (NMS): The algorithm uses NMS to eliminate overlapping boxes and select the most likely object in each grid cell. This helps to reduce the number of false positives and improve the overall accuracy of the model.
e. Post-processing: The final output of the model is a set of bounding boxes and class labels for each object in the image, along with the associated confidence scores. Multi-scale inference: YOLOv5 also uses multi-scale inference, which is when the image is resized and processed multiple times at different scales to detect objects of various sizes.
f. Anchor boxes: Anchor boxes are pre-defined boxes that are used to help the model make more accurate predictions, by defining the "prior" of the object size.
g. SPADE: SPADE is used to predict object shapes and sizes more accurately; it is a new architecture that allows the model to learn scale-aware features.

2.2 *System Development Stages*

Structured development stages are formulated to make the system-building process more straightforward. Fig. 3 shows the structured development stages of the system's reaction.

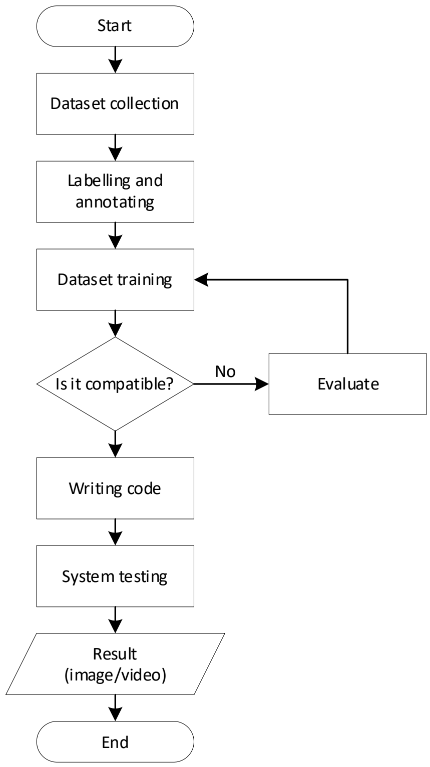

Figure 3. System development stages

The individual development stages for making the system are as follows:

1) Collecting the datasets. A dataset is a collection of data that contains past information and is ready to be extracted into new information. Datasets generally have more than one variable and contain certain topics [26]. The dataset determines whether the detection results will be optimal or not [27]. The dataset collection itself does not have minimum and maximum limits, but the more datasets that are successfully collected, the better. The dataset used in this study is an image of a smartphone object. The dataset used is 1,038 smartphone objects with various types of smartphones and positions. Fig. 4 shows some examples of images taken as a dataset. The images used as the dataset are taken from smartphone photos of various types and brands from the internet.

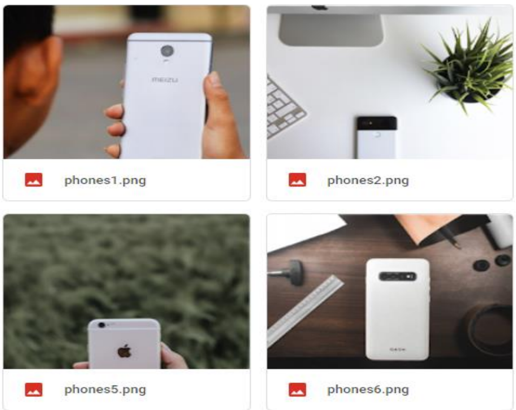

Figure 4. Images for dataset example

2) Labeling or annotations. In this stage, the system designer labels objects (annotations). Annotation is done by giving a bounding box to mark recognized objects [28]. Basically, the labeling and annotation should be done by someone who is an expert in the field, but because the object being labeled is a smartphone, the labeling can be done by the system maker himself. The labeling procedure is a manual process. Fig. 5 is an example of a labeling or annotation procedure. Dataset labeling is done by the system builder by marking objects in the image using a rectangle selection tool.

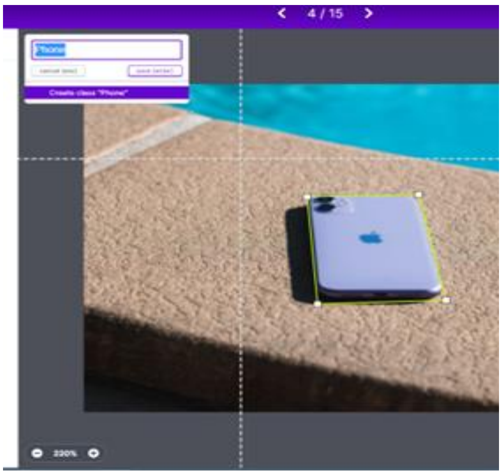

Figure 5. Annotation process

3) Training the datasets. This stage is conducted so that the object can be recognized by the system. Training dataset used to train models, train algorithms, and perform probability calculations based on each data learning result [29]. If the results of the training are as expected, then the process is continued to the next step, contrary, if it is not appropriate then an evaluation is performed until the training matches the dataset. The results of the training data can be seen in Fig. 6.

Figure 6. Training result

4) Coding. This stage is the stage of writing program code using the Visual Studio Code text editor, the Python programming language, and the Yolov5s model. In the source code configuration script, the program will determine which device type will be used.
5) Testing the system. This stage is for testing the system's ability to detect objects in the form of smartphones and capture them using a camera or webcam. The result is an image of an object on a computer screen with a description of the name of the object listed on the bounding box, and then displaying the test results in the form of photos or videos.

2.3. *System Workflow*

For the system to detect objects in the form of smartphones properly, it is necessary to conduct data training. The system will learn to recognize the smartphone that has been collected as a dataset. After the data is trained, the system is then tested to find out how it works. This system has the main function of detecting the use of smartphones in the classroom. The way the system works is that, at first, the system must be activated in order to detect the smartphone. Furthermore, the camera stands by to detect the presence of smartphone objects used in class. The detection results will be displayed on the monitor screen. The way the system works is as shown in Fig. 7.

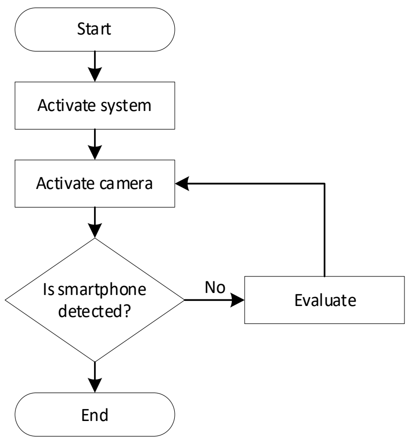

Figure 7. System workflow

3 RESULT AND DISCUSSION

The dataset used in this study consist of 1038 images originating from the internet and camera captures. Dataset is divided into three parts, namely a training set of 92%, a validation set of 4%, and a testing set of 4%. The training conditions used are auto-orientation with resize 416 x 416, reverse augmentation done with horizontal and vertical angles, and 90-degree angle rotation done with options clockwise, counter-clockwise, and reverse (upside down), as well as image blur.

3.1. Training Result

Training is carried out in several stages, namely, the dataset is trained with auto-orientation conditions with a size of 416 x 416, then reverse augmentation, and angular rotation. The following are some graphs of the results of the dataset training that has been done.

1) *Training Result using mAP_0.5*

mAP_05 is the distance matrix range, where mAP is an extension of the mean average precision. The training results in Fig. 8 show the average precision of image matching starting from the beginning of the training, which can be seen on the X starting from the 0 range and ending in the 150 range. The Y, which shows the level of image detection precision starting from 0 to 1, shows a good training test result, where the average training result shows a successful level of precision up to 0.95 mAP. However, the precision has decreased from 0.9 to 0.75 in the range of 50, which can be seen on the X.

2) *Training result using mAp_0.5:0.95*

The results of training with mAP_05:0.95 are the same as the comparison of training with mAP_0.5. Only the size of the matrix range is reduced to clearly show the results of training that runs at a precision level. The training results show that the level of accuracy increases at the accuracy point of 0.35 and is stable up to 0.75. The training steps displayed on the X start from 0 to 150 steps; the steps on the X replace the time steps used during the training.

3) *Training result for detection accuracy*

The matrix in Fig. 9 shows a more detailed accuracy chart than the mAP_0.5 and mAp_0.5:0.95 matrices, but the displayed movements show a more dominant accuracy level in each image during training. Because these two things are always related, the level of accuracy will always be followed by a recall or return value. In the matrix of accuracy results above, the accuracy level experienced significant fluctuation. The 23rd step is the step that experiences the first drastic decrease, namely the accuracy level dropping from 0.86 to 0 accuracy. The second drastic decrease occurs at step 34, where the accuracy level, which is worth 0.9, also drops to 0. The level of accuracy is stable from step 37 to 39, where it rises from 0 to a value of 0.88 and remains stable until the final step. The level of accuracy fluctuates because the datasets held for training have different factors.

4) *Training result for recall data*

The recall matrix or value shows a condition where, when doing training, the existing data will lose the labeling box. Fig. 10 shows a stable return value following the existing level of accuracy until the last step. The return value did not experience a significant decrease.

5) *Training result for box_loss data*

The box-loss matrix as shown in Fig. 11 is a matrix that shows annotation boxes that are lost, missing, or not shown in the image when conducting training procedures. The matrix starts with the height of the missing box data, which is at a value of 0.04. This happens because the image data to be trained has just started. The missing annotation box starts to decrease as training goes on until it reaches 0.015, meaning the annotation box was not lost and the training was successfully carried out. The missing annotation box will decrease with a stable indication of accuracy that can be seen in the accuracy matrix.

6) *Training result for clc_loss data*

The cls_loss matrix, as shown in Fig. 12, is a matrix showing missing classifications, but in this research, when conducting class loss data training, the missing or inappropriate image classifications decrease until the last step. This decrease is indicated by the results of the matrix, starting from the value of 6.5 E-3 and decreasing to 1.5 E-3 data class loss. The value of the matrix above is below the value of 0, meaning that the level of loss presents but within normal limits for the dataset trained.

7) *Training result for obj_loss data*

The obj_loss matrix as shown in Fig. 13 is a lost object detection matrix, specifically for smartphone and laptop object detection. Similar to the box_loss matrix, loss of objects occurs when training starts. The accuracy of smartphone and laptop objects detected during training begins to increase. This is indicated by the decreasing rate of object loss until the last step, when the loss value decreases to 9E-3, which can be seen on the Y axis of the matrix.

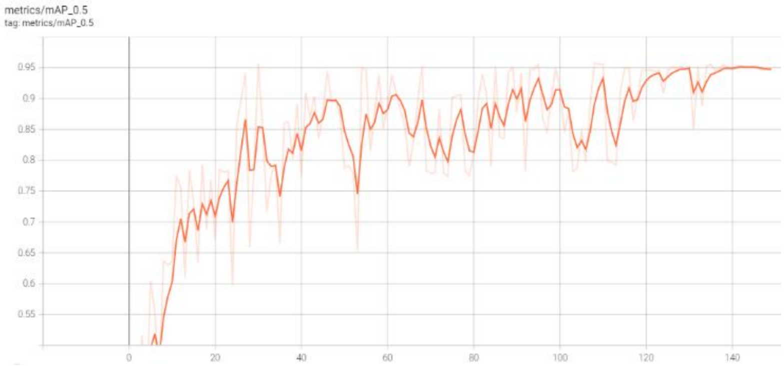

Figure 8. Training result using mAP_0.5

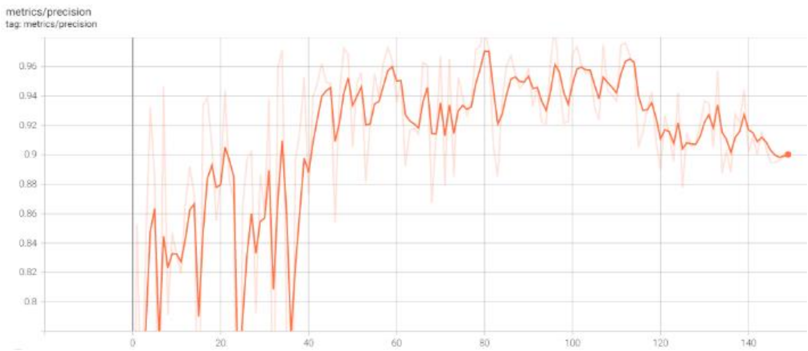

Figure 9. Training result for detection accuracy

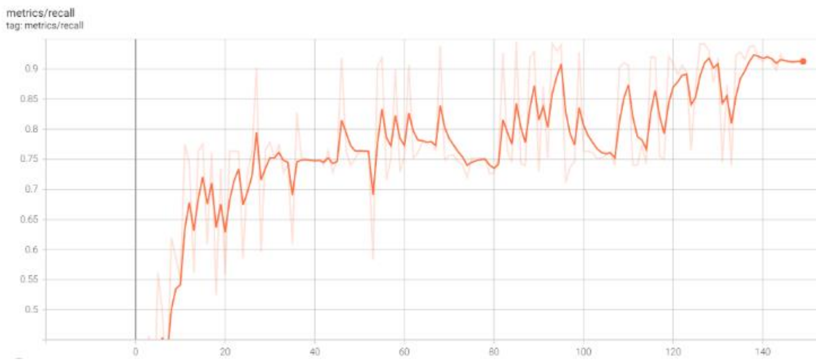

Figure 10. Training result for recall data

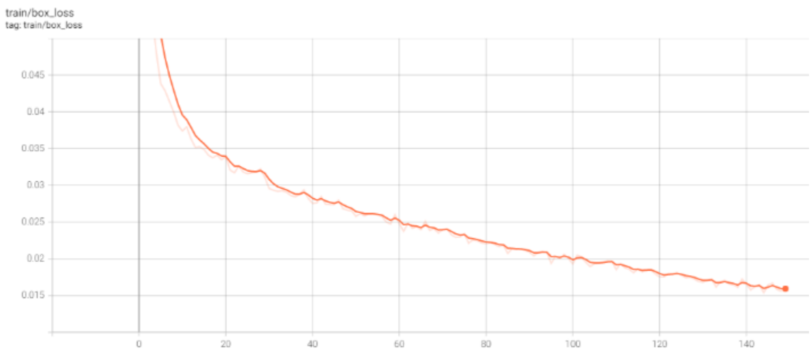

Figure 11. Training result for box_loss data

Figure 12. Training result for clc_loss data

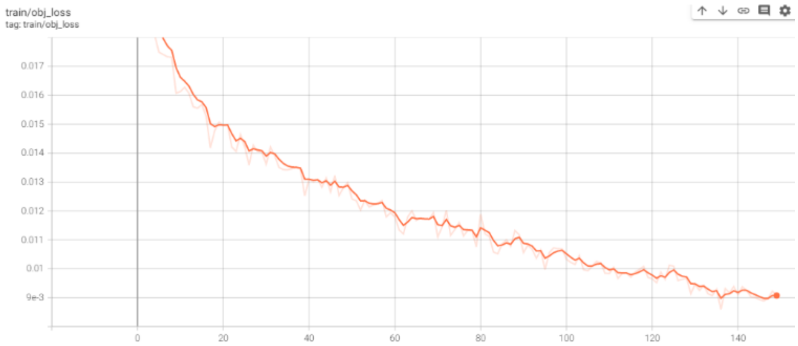

Figure 13. Training result for obj_loss data

8) *Validation result for box_loss*

The val/box_loss matrix in is almost the same as the train/object_loss matrix, but it shows only validation data. The validation data comes from the distribution of training results. The validation results show 43 images with an accuracy of 4% of the total training results. The missing box data begins with high accuracy from the result matrix. This is normal because the new data will start in the training process and the missing box data will decrease along with the end of the image training process. In the validation matrix of the missing data box results, the range of occurrence only starts from 0.03 to 0.02, meaning that the data losses are minimal and do not interfere with the existing dataset.

9) *Validation result for obj_loss*

The val/obj_loss matrix, similar to the val/box_loss matrix, starts with a high level of object loss. The initial process of image training will result in a large object loss, but the object loss will decrease as the training process progresses. The results of the loss object validation matrix are in the range 5.8E-3 to 3.8E-3; this range is also not much different from the results of the obj_loss validation.

10) *Validation result for cls_loss*

The val/cls_loss matrix tells a similar story to the train/cls_loss matrix. This matrix shows the missing classification results. The missing classification is in the validation results, but with different results. As shown in the matrix display below, the classification results on validation show missing data starting from the range 2.6E-3 to 1E-3. Like the other matrices, this one's value is still below 0, which means that it does not have much influence on the results of the training on the dataset.

3.2. *Testing*

3.2.1. *Testing using Single Smartphone:* In the first trial, testing was performed on one object, as shown in Fig. 14, with a test time of 1 minute. There was one smartphone that was detected 12 times in a span of 2.8 seconds. The VSCode terminal indicates the amount of each detection time, and the detection running results can be seen in Fig. 15.

3.2.2. *Testing on Multiple Smartphone:* In the second trial, tests were carried out on 2 objects, as shown in Fig. 16, with a test time of 1 minute. In this test, 2 smartphones were detected, and these smartphones were detected 26 times in a span of 7.045 seconds.

Figure 14. Testing on single smartphone (single object)

```
0: 320x416 1 phone, Done. (0.250s)
0: 320x416 1 phone, Done. (0.228s)
0: 320x416 1 phone, Done. (0.234s)
0: 320x416 1 phone, Done. (0.217s)
0: 320x416 1 phone, Done. (0.226s)
0: 320x416 1 phone, Done. (0.221s)
0: 320x416 1 phone, Done. (0.216s)
0: 320x416 1 phone, Done. (0.235s)
0: 320x416 1 phone, Done. (0.238s)
0: 320x416 1 phone, Done. (0.232s)
0: 320x416 1 phone, Done. (0.220s)
0: 320x416 1 phone, Done. (0.247s)
```

Figure 15. VSCode terminal result

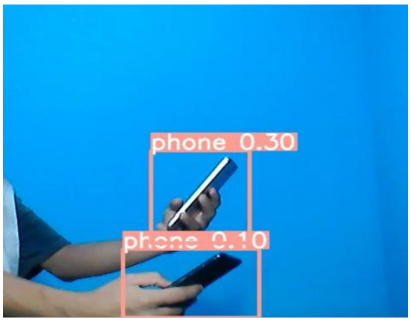

Figure 16. Testing result on 2 objects

In the third trial, three objects were tested as shown in Fig. 17 with a test time of 1 minute and 30 seconds; in this test, there were three smartphones detected, which were detected 47 times within a time span of 12.583 seconds.

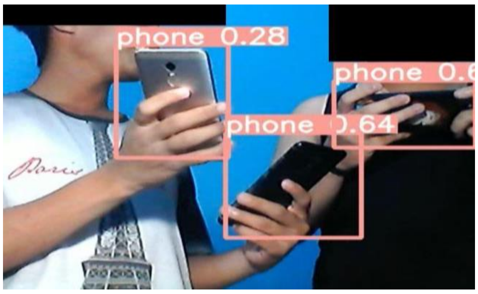

Figure 17. Testing result on 3 objects

The results of the research show that the YOLOv5 algorithm successfully recognizes objects. This is demonstrated in the detection results that appear with the bounding box, along with the predicted object name of the detected object and the displayed confidence value.

4 CONCLUSION

This object detection system was made to detect smartphone objects during the learning session in class. This system is used to assist teachers in supervising the use of smartphones by student session in class. This system is used to assist teachers in supervising the use of smartphones by students. The YOLO V5 algorithm was used to build this system. This system's testing stage was successful, and it can detect objects on smartphones. Smartphones can be detected both in dark and well-lit conditions and from many positions, such as the front view, side view, and back view. Smartphones that are actively used will be detected if they are in the classroom, as well as in single- and multiple-object conditions. From the results of the model training, it was found that the mean average precision (mAP) was 77.7%, the precision value was 93.2%, and the recall value was 71.7%. The drawback of this system is that there are still a few smartphone objects that are not detected. This system can assist teachers in supervising the use of smartphones by students in class or during exams. Real-time camera detection results are displayed on a monitor screen to assist teachers or supervisors in supervising smartphone use. Smartphones detected by the system will be displayed with a bounding box and a confidence value. The drawback of this system is that there is no automatic notification when the system detects a smartphone, so the supervision process must be performed by directly observing the monitor screen.

AUTHOR'S CONTRIBUTION

Rr. Hajar Puji Sejati is the first author, conducting data analysis and application concept design; Rodhiyah Mardhiyyah is the second author, performing a literature review of the previous research. Zulkhairi is the third author, conducting application development and data collection. Nur Istiqomah is the fourth author, assisting the second author on the literature review. R. Imam Budi Prasetya is the fifth author, assisting the third author on application development and data collections.

COMPETING INTERESTS

Complying with the publication ethics of this journal, Rr. Hajar Puji Sejati, Rodhiyah Mardhiyyah, Zulkhairi, Nur Istiqomah, and R. Imam Budi Prasetya, as the authors of this article, declare that this article is free from conflict of interest (COI) or competing interest (CI).

ACKNOWLEDGMENT

The author would like to thank the Ministry of Education, Culture, Research, and Technology 2022 for providing funding support for this research.

REFERENCES

[1] Y. Intaniasari and R. D. Utami, "Jurnal basicedu," *J. Basicedu*, vol. 6, no. 3, pp. 4987–4998, 2022, [Online]. Available: https://jbasic.org/index.php/basicedu/article/view/2996/pdf.

[2] I. W. Ningsih, A. Widodo, and Asrin, "Urgensi Kompetensi Literasi Digital dalam Pembelajaran pada Masa Pandemi Covid-19," *J. Inov. Teknol. Pendidik.*, vol. 8, no. 2, pp. 132–139, 2021, [Online]. Available: https://journal.uny.ac.id/index.php/jitp/article/view/35912/17365.

[3] S. Fitri, F. D. Saputra, and M. Taufiq, "Pengaruh Penggunaan Smartphone Terhadap Minat Belajar Siswa SMK Negeri 1 Tasikmalaya," *JUPEIS J. Pendidik. dan Ilmu Sos.*, vol. 1, no. 3, 2022, [Online]. Available: http://jurnal.jomparnd.com/index.php/jp/article/view/65/174.

[4] R. Deol and Meenakshi, "Impact of Mobile Phone Use on School

Student Interpersonal Communication," *Turkish Online J. Qual. Inq.*, vol. 12, no. 6, pp. 565–574, 2021, [Online]. Available: https://www.tojqi.net/index.php/journal/article/view/1186/533.

[5] M. Rizky and A. Subiyakto, "Pemanfaatan Artificial Intelligence dalam Menghadapi Pademi Covid-19 : Systematic Literatur Riview," *J. Sist. Cerdas*, vol. 05, no. 01, pp. 46–52, 2022, [Online]. Available: https://apic.id/jurnal/index.php/jsc/article/view/184/99.

[6] M. Siahaan, C. H. Jasa, K. Anderson, M. V. Rosiana, S. Lim, and W. Yudianto, "Penerapan Artificial Intelligence (AI) Terhadap Seorang Penyandang Disabilitas Tunanetra," *J. Inf. Syst. Technol.*, vol. 01, no. 02, pp. 186–193, 2020, [Online]. Available: https://journal.uib.ac.id/index.php/joint/article/download/4322/1122.

[7] Y. Rana, "Python : Simple Though an Important Programming Language," *Int. Res. J. Eng. Technol.*, vol. 6, no. 2, pp. 1856–1858, 2019, [Online]. Available: https://www.irjet.net/archives/V6/i2/IRJET-V6I2367.pdf.

[8] S. Raschka, J. Patterson, and C. Nolet, "Machine Learning in Python : Main Developments and Technology Trends in Data Science , Machine Learning , and Artificial Intelligence," *Information*, vol. 11, no. 4, p. 193, 2020, doi: 10.3390/info11040193.

[9] A. Aprilino, I. Husni, and A. Amin, "Implementasi Algoritma Yolo dan Tesseract OCR pada Sistem Deteksi Plat Nomor Otomatis," *J. TEKNOINFO*, vol. 16, no. 1, pp. 54–59, 2022, [Online]. Available: https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/article/view/1522.

[10] A. A. Putra, I. W. Wahyuni, Alucyana, and Ajriya, "Pengaruh Penggunaan Handphone Pada Siswa Sekolah Dasar," *Al-Hikmah J. Agama dan Ilmu Pengetah.*, vol. 18, no. 1, pp. 79–89, 2021, doi: 10.25299/al-hikmah:jaip.2021.vol18(1).6531.

[11] L. M. Ramjan *et al.*, "The negative impact of smartphone usage on nursing students: An integrative literature review," *Nurse Educ. Today*, vol. 102, no. April, p. 104909, 2021, doi: 10.1016/j.nedt.2021.104909.

[12] S. Han, "Impact of Smartphones on Students: How Age at First Use and Duration of Usage Affect Learning and Academic Progress," *Technol. Soc.*, vol. 70, p. 102002, 2022, doi: https://doi.org/10.1016/j.techsoc.2022.102002.

[13] J. S. Murthy, G. M. Siddesh, W. C. Lai, B. D. Parameshachari, S. N. Patil, and K. L. Hemalatha, "Object Detect: A Real-Time Object Detection Framework for Advanced Driver Assistant Systems using YOLOv5," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/9444360.

[14] T. Abuzairi, Nurdina Widanti, Arie Kusumaningrum, and Yeni Rustina, "Implementasi Convolutional Neural Network Untuk Deteksi Nyeri Bayi Melalui Citra Wajah Dengan YOLO," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 624–630, 2021, doi: 10.29207/resti.v5i4.3184.

[15] B. Widodo, H. Armanto, and E. Setyati, "Deteksi Pemakaian Helm Proyek Dengan Metode Convolutional Neural Network," *J. Intell. Syst. Comput.*, vol. 3, no. 1, pp. 23–29, 2021, [Online]. Available: https://jurnal.stts.edu/index.php/INSYST/article/view/157/83.

[16] M. R. Ardiansyah, Y. Supit, and M. S. Said, "Sistem Vidi Komputer untuk Kalkulasi Kepadatan Kendaraan Menggunakan Algoritma Yolo," *J. Sist. Inf. dan Tek. Komput.*, vol. 7, no. 1, 2022, [Online]. Available: http://ejournal.catursakti.ac.id/index.php/simtek/article/view/123/138 .

[17] A. Lazaro, "Deteksi Jenis Kendaraan di Jalan Menggunakan OpenCV," Institut Teknologi Sepuluh Nopember, 2017.

[18] H. O. K. Sugianto, M. A. D. Widyadara, and A. B. Setiawan, "Implementation of Face Recognition for Attendance Using Yolo V3 Method," *Semin. Nas. Inov. Teknol.*, vol. 6, no. 2, pp. 50–55, 2022, [Online]. Available: https://proceeding.unpkediri.ac.id/index.php/inotek/article/view/2559 .

[19] J. W. Chen, W. J. Lin, H. J. Cheng, C. L. Hung, C. Y. Lin, and S. P. Chen, "A Smartphone-Based Application for Scale Pest Detection Using Multiple-Object Detection Methods," *Electron.*, vol. 10, no. 4, pp. 1–14, 2021, doi: 10.3390/electronics10040372.

[20] Q. Aini, N. Lutfiani, H. Kusumah, and M. S. Zahran, "Deteksi dan Pengenalan Objek Dengan Model Machine Learning: Model YOLO," *CESS (Journal Comput. Eng. Syst. Sci.*, vol. 6, no. 2, pp. 192–199, 2021, doi: 10.24114/cess.v6i2.25840.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[22] A. N. Sugandi, B. Hartono, and K. Kunci, "Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO," *Pros. 13th Ind. Res. Work. Natl. Semin.*, pp. 13–14, 2022.

[23] M. Lamane, M. Tabaa, and A. Klilou, "Classification of Targets Detected by mmWave Radar using YOLOv5," *Procedia Comput. Sci.*, vol. 203, pp. 426–431, 2022, doi: 10.1016/j.procs.2022.07.056.

[24] A. Saffari, S. Y. Tan, M. Katanbaf, H. Saha, J. R. Smith, and S. Sarkar, "Battery-Free Camera Occupancy Detection System," *EMDL'21 Proc. 5th Int. Work. Embed. Mob. Deep Learn.*, pp. 13–18, 2021, [Online]. Available: https://dl.acm.org/doi/10.1145/3469116.3470013.

[25] K. A. Baihaqi and C. Zonyfar, "Deteksi Lahan Pertanian Yang Terdampak Hama Tikus Menggunakan Yolo v5," *J. Inform.*, vol. 11, no. 02, pp. 1–9, 2022, [Online]. Available: https://journal.unsika.ac.id/index.php/syntax/article/view/7226/3496.

[26] K. Harian, "Pengertian Dataset dan Jenis-jenisnya," *Kumparan.com*, 2021. .

[27] R. Imantiyar and D. H. Fudholi, "Kajian Pengaruh Dataset dan Bias Dataset terhadap Performa Akurasi Deteksi Objek," *PETIR (Jurnal Pengkaj. dan Penerapan Tek. Inform.*, vol. 14, no. 2, pp. 258–268, 2021, doi: 10.33322/petir.v14i2.1350.

[28] P. Kowalczyk, J. Izydorczyk, and M. Szelest, "Evaluation Methodology for Object Detection and Tracking in Bounding Box Based Perception Modules," *Electron.*, vol. 11, no. 8, pp. 1–26, 2022, doi: 10.3390/electronics11081182.

[29] R. A. Anggraini, G. Widagdo, A. S. Budi, and M. Qomaruddin, "Penerapan Data Mining Classification untuk Data Blogger Menggunakan Metode Naïve Bayes," *J. Sist. dan Teknol. Inf.*, vol. 7, no. 1, pp. 47–51, 2019, doi: 10.26418/justin.v7i1.30211.