Journal of Emerging Information Systems and Business Intelligence
ISSN: 2774-3993
Journal homepage: https://ejournal.unesa.ac.id/index.php/JEISBI/

Comparison You Only Look Once (Yolo) Algorithm On Physical Violence Video Detection

Aulia Anisa Puji Rahayu[1], Wiyli Yustanti[2]
[1,2]*State University of Surabaya, Surabaya, Indonesia*
[1]auliaanisa.21017@mhs.unesa.ac.id,[2] wiyliyustanti@unesa.ac.id

ABSTRACT

Physical violence is one of the crimes that often occurs in various environments and can have a serious impact on victims, both physically and mentally. One of the obstacles in handling it is the delay in detecting acts of violence. The solution to this problem is to implement the best algorithm between You Only Look Once (YOLO) version 8 and version 9 to detect physical violence through video automatically and quickly. The dataset used consists of two classes, namely violence and non-violence, which have gone through the process of extraction, data cleaning, and labeling using Roboflow. The model was trained using Google Collaboratory, and the training results were evaluated using mAP, precision, recall, and F1-score metrics. Based on the test results, YOLOv9 obtained the best performance with a precision of 0.8096, recall of 0.8665, F1-score of 0.8363, and mAP of 0.8117. The detection system is then implemented into a web-based application using the Flask framework, which allows users to Upload videos and detect acts of violence automatically. The test results show that the application runs according to its function and is able to detect physical violence well. This research is expected to be a supporting solution in video-based security surveillance systems.

Keyword: Physical Violence, YOLOv8, YOLOv9, Video Detection, Deep learning, Object Detection

Article Info:
Article history:
Received July 11, 2025
Revised August 02, 2025
Accepted September 04, 2025

Corresponding Author
Wiyli Yustanti
State University of Surabaya, Surabaya, Indonesia
wiyliyustanti@unesa.ac.id

1. INTRODUCTION

Physical violence is one of the most serious social problems, because it not only has a physical impact, but also a psychological impact on its victims. Based on data from the World Health Organization (WHO), more than 1.6 million people die each year due to violence, making it one of the leading causes of death in the world [1]. In Indonesia, in 2023 there were 11.099 cases of physical violence against women and 4.025 cases against children. This data shows the urgency of early detection and prevention of violence [2]. A major challenge in addressing violence is late detection, which can worsen the impact on victims. The increasing cases of violence emphasize the importance of more effective prevention and early detection [3]. The development of deep learning technology provides new opportunities in automatically detecting violent acts through images and videos.

Several previous studies have implemented the YOLO (You Only Look Once) algorithm in various versions, such as YOLOv5 and YOLOv8, and showed promising results. One study even combined YOLOv9 with the Threat Events Ontology (TEO) approach to detect violence in surveillance videos, and managed to achieve a mAP value of 83.7% [4]. Research by Sidik (2024) developed child abuse and bullying detection using YOLOv8, with 85% accuracy, 81.8% precision, 90% recall, and 85.7% F1-score. This model excels in real-time detection, but is still limited in the amount and diversity of data. [5]. Arun Akash et al. (2022) examined human violence detection using Inception-v3 for image classification and YOLOv5 for object detection. This study used 10.000 images from video footage and achieved 74% accuracy in detecting violence [6]. However, there are limited studies that specifically compare the performance of the latest versions of YOLO, specifically YOLOv8 and YOLOv9, in the context of physical violence detection through video. Therefore, this study was conducted to fill this gap by comparing the performance of both models using evaluation metrics such as precision, recall, F1-score, and mAP.

The dataset used consists of two classes: violent and non-violent, which have gone through the labeling process using Roboflow. The model training results were also tested directly through the development of a Flask-based web application. The selection of YOLOv8 and YOLOv9 is based on their architectural advantages. YOLOv8 is known for its better efficiency and detection speed compared to previous versions, while YOLOv9 offers a more sophisticated and efficient convolutional layer structure in visual data processing. This research aims to determine the most effective YOLO model in detecting physical violence in real situations, such as in schools, workplaces, and public spaces. In addition, the results of this research are also expected to serve as a practical guide for developers of video-based security monitoring systems.

2. METHODS

2.1 Knowledge Discover in Database (KDD)

This stage follows the stages in Knowledge Discovery in Database (KDD). The stages in KDD are as follows:

A. Selection

In this data selection stage, data selection from a set of operational data will be carried out before the information mining stage in Knowledge Discovery in Database (KDD) begins. According to (Kar, 2020) the minimum YOLO training data that must be collected is 100 data per class, ideally, good training uses 1.000 images [7]. The selected data will be used for the data mining process.

1. Video Collection

The researcher collected data from various sources, one of which was through social media Twitter using the hashtag **#videokekerasan** to obtain videos related to physical violence as the focus of the research. To obtain data on non-violent videos, we purposively sampled videos from YouTube and TikTok based on visual criteria without elements of physical violence. This data was used as a comparison in training and evaluating the detection model.

2. Stages of video to image extraction

The process of extracting images from videos was carried out at Google Collaboratory through several stages, from mounting Google Drive, importing OpenCV libraries, to determining input and output folders. A total of 50 violent videos and 50 non-violent videos were processed by taking frames every 0.5 seconds, then saved as images (.jpg). These images are used for object labeling using Roboflow and become important datasets in training object detection models such as YOLO.

3. Data Cleaning

The first step is to clean the data from outliers, such as unclear traffic images, images that do not fit the research needs, images that are too dark, and noises and other factors [8]. This process is done together with image separation.

Table 1. Total Dataset

Data Type	Total Dataset
Violence Class	2.903
Non-violence Class	2.074
Total Dataset	4.997

From the extraction results, 2.903 images of violence category and 2.074 images of non-violence category were obtained, but after the data cleaning process, only 1.620 images were used. Cleaning is done by the check and save method manually because the data is in the form of non-numeric images, so it takes a long time to delete outlier data that does not match.

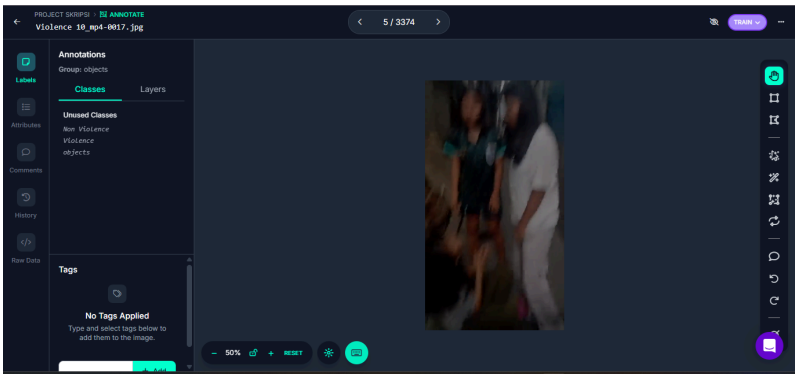

Figure 1. Violence Image Too Blurry

Some extracted images appear blurry or unclear due to the low quality of the violent video, which can interfere with the annotation process and reduce data accuracy. In addition to blurry images, other distractions such as obstructing objects, dark lighting, and noise are also taken into consideration. Unsuitable images will be discarded, while suitable images will be kept by the researcher.

4. Data Labelling

After the data was collected and cleaned, a total of 1.620 images were manually labeled using Roboflow with two classes of violence and non-violence. Labeling was done on individuals who did or did not commit acts of physical violence, with a format suitable for the YOLO architecture. The focus of labeling is on the presence of violent actions in a frame, so the relevant human objects are labeled as a whole, rather than based on specific points. The input image is divided into a number of grids, and each grid is responsible for predicting the

bounding box and probabilities for the objects that appear within it. On each grid, the YOLO algorithm predicts several bounding boxes that are likely to contain objects. Usually, each grid has several bounding boxes called anchor boxes. For each anchor box, the YOLO algorithm predicts an offset value that describes the relative location and size of the bounding box with respect to the grid [9].

Figure 2. Image Labeling

In Figure 3 is the result obtained after labeling.

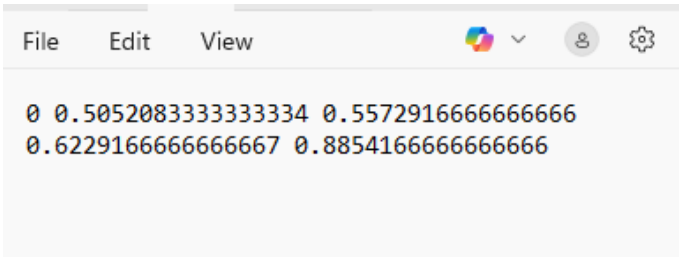

Figure 3 txt file labeling result

Each labeled image will have a ".txt" file containing the labeling information. The first number indicates the class code (0 for non-violence and 1 for violence), followed by the coordinates of the bounding box position (x, y, height, and width). This ".txt" file is created automatically after the labeling process is complete.

5. Data training, validation and testing

The dataset of 1.620 images was divided in a 90:10 proportion, with 162 images as the final testing data and 1.458 images for training and validation. The testing data is used only for the final evaluation, so that the model performance results remain objective and avoid overfitting. Meanwhile, the training and validation data was processed using the 3-Fold Cross Validation technique, where the data was divided into three parts for alternating training and validation. Both models, YOLOv8 and YOLOv9, were trained and tested using the same data split for a fair and accurate performance comparison.

B. pre-Procesing

1. Resize

The extracted and labeled images were resized to 640 x 640 pixels to homogenize the input to the object detection model training. This size was chosen because it is the recommended standard for YOLOv8 and YOLOv9 and offers a balance between detection

accuracy and computational efficiency. With a consistent size, the model can work more effectively without losing important details of the object in the violence video frame. Figure 4 shows the original image before resizing, with dimensions of 840 x 840 pixels.

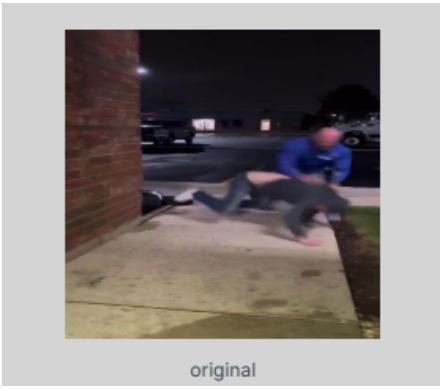

Figure 4. Image Before Resizing

Figure 5 shows the original image with dimensions of 640x640 pixels after resizing.

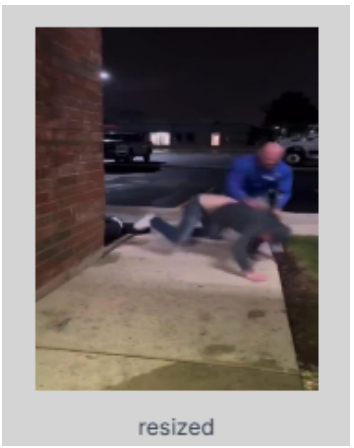

Figure 5. Image Image After Resizing

C. Transformation

At this stage, the data is converted to YOLOv8 and YOLOv9 formats. This data change is also done on the Roboflow website. This transformation process is carried out as in Figure 6.

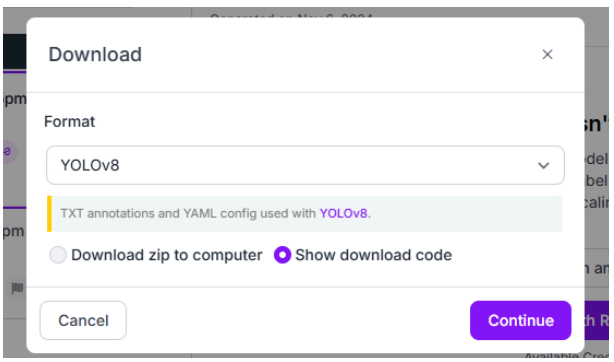

Figure 6. YOLOv8 Transformation

D. Data Mining

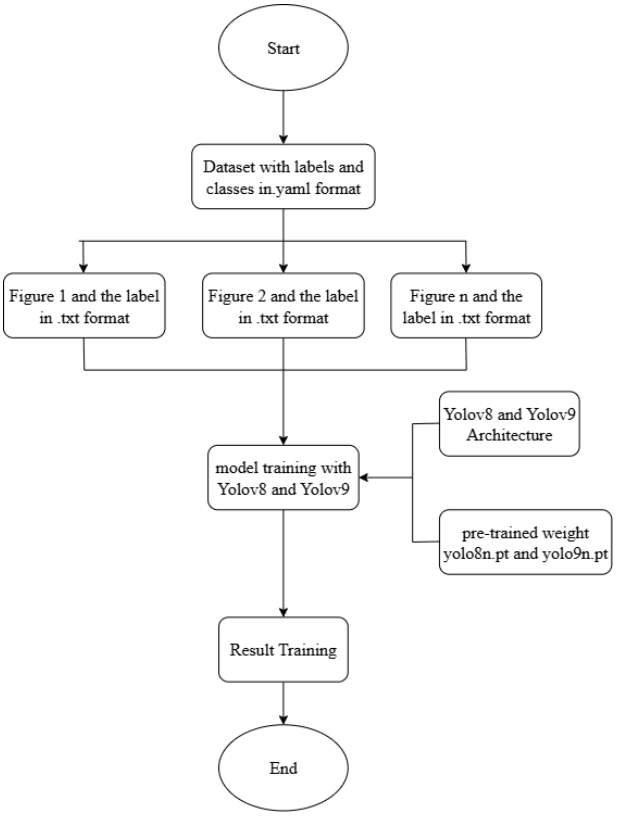

Figure 7. Training YOLO(v8,v9)

The training process was conducted at Google Collaboratory using the pretrained weight YOLOv8n.pt and YOLOv9n.pt. Google Drive is connected in advance to prevent data loss in case of error or runtime disconnection. In this training, hyperparameter tuning is also carried out by adjusting parameters such as learning rate, optimizer, epoch, and batch size to optimize model performance.

E. Interpretation/ Evaluation

The evaluation of model performance in this study refers to the approach used by Jatmiko and Pristyanto (2023), namely by using the confusion matrix and calculating the precision, recall, and f1-score values to assess the quality of model classification. The evaluation stage is carried out to assess the quality of the classification model and identify the causes if the model performance is not good. The main evaluation uses the mAP (mean Average Precision) metric, where the higher the value, the more accurate the object detection. In addition to mAP, other metrics such as precision, recall, and F1-score are also used to evaluate the overall performance of the model. Precision is the correlation of true positive predictions with all positive predictions, recall is the proportion of accurate positive predictions for all true positive data [11]. F1-score is the combined average of precision and recall [12]. mAP is the main indicator in evaluating model performance, this map is the average of Average precision (AP) [13].

2.2 Rapid Aplicataion Development

According to Galil Gibran et al. (2018) in Kendall (2010), Rapid Application Development (RAD) is an object-based approach involving software and system development methodologies [14]. RAD consists of three main stages, namely:

A. Requirements planning

At this stage, a needs analysis is carried out in the development of a violence detection system. In terms of users, the application is intended for the general public who want to automatically detect physical violence in videos using the YOLO algorithm. In terms of hardware, the system requires a 10th generation Intel processor, 8GB RAM, and input devices such as a keyboard and mouse. Meanwhile, software requirements include Windows 11, Google Chrome, Draw.io, and Visual Studio Code.

B. Design

Based on the requirement planning, two main interfaces were designed for the physical violence detection application. First, the initial interface contains the system title, a Select Video button to upload the file, as well as a media player to preview the video and a Detect button to start the analysis. Second, the detection result view displays a media player to view the analyzed video that has been annotated with violence, as well as two buttons: Download Video to save the detection results and Return to Main Page to start a new process. The interface is designed to be simple and easy to use.

C. Implementation Stage

In the implementation stage, the model resulting from the KDD process is integrated into the website using a framework. According to (Maity et al. (2023)) Frameworks involve several main steps, including collection, data labeling, and model architecture or modeling [15]. Flask because it supports the use of YOLOv9 models and is based on Python. The Flask project structure includes a models folder for storing YOLO model files, and an app.py file as the core of the program that calls the library and runs detection when a video is uploaded. The app has two main views: a home page for video upload and detection, and a results page to display the violence-annotated video output.

The detection process starts from loading the YOLO model, capturing video details (such as frames and fps), to applying detection to each frame using bounding boxes. The result is saved as a new video (output.mp4) which is converted to web format so that it can be displayed on the user interface. After implementation, the system was tested using the black box testing method to ensure that all features work as per the function based on the input-output. System performance evaluation was also conducted to assess the accuracy of violence detection in user uploaded videos.

3. RESULTS AND DISCUSSION

3.1 Model Evaluation

The results of the training process of each model based on the division of Cross Validation 3 Folt can be seen in the following table:

Table 2 . Training Result

Model	Iterasions	Process	Precision	Recall	F1 Score	mAP
YOLOv8	1	Train 1	0.9922	0.9957	0.9939	0.9947
		Val 1	0.7063	0.702	0.7774	0.8037
		Test 1	0.9011	0.9108	0.8908	0.8955
YOLOv8	2	Train 2	0.9898	0.9958	0.9928	0.9946
		Val 2	0.8341	0.7375	0.7828	0.83
		Test 2	0.9493	0.9456	0.9475	0.957
YOLOv8	3	Train 3	0.9909	0.9913	0.9911	0.9948
		Val 3	0.8185	0.7398	0.7739	0.8037
		Test 3	0.7257	0.7984	0.7603	0.7049
YOLOv9	4	Train 4	0.9825	0.9756	0.9791	0.9926
		Val 4	0.803	0.8283	0.8351	0.7929
		Test 4	0.8062	0.8622	0.8333	0.8716
YOLOv9	5	Train 5	0.993	0.9951	0.994	0.9947
		Val 5	0.8238	0.7755	0.7989	0.8387
		Test 5	0.9589	0.9601	0.9595	0.6809
YOLOv9	6	Train 6	0.9946	0.995	0.9948	0.9948
		Val 6	0.859	0.8218	0.85	0.8439
		Test 6	0.6636	0.7773	0.716	0.8825

Furthermore, if the average of the performance metrics between YOLOv8 and YOLOv9 in the training (train), validation (val), and testing (test) processes is calculated, an overview of the effectiveness of each version of the model is obtained.

Table 3. Average YOLOv9 Model Evaluation Results

Process	Precision	Recall	F1 Score	mAP@0.5
Train	0.9900	0.9886	0.9893	0.9940
Val	0.8286	0.8085	0.8280	0.8252
Test	0.8096	0.8665	0.8363	0.8117

Table 4. Average YOLOv8 Model Evaluation Results

Process	Precision	Recall	F1 Score	mAP@0.5
Train	0.9910	0.9943	0.9926	0.9947
Val	0.7863	0.7264	0.7780	0.8125
Test	0.8587	0.8849	0.8662	0.8525

The YOLOv9 model was chosen for the deployment process as it showed stable and consistent performance during training, with very high precision and recall values above 0.98 each, indicating its ability to learn the data thoroughly. Although the mAP and F1 Score values on the test data were slightly lower than YOLOv8, YOLOv9 still showed competitive performance and had higher recall values on both validation and test data. This is especially important in the context of violence detection, where the ability of the model to detect as many cases as possible (recall) is preferred.

The evaluation visualization also shows that YOLOv9 is able to achieve a mAP of almost 1.0 at the 40th epoch. The graph shows that the highest F1 Score for all classes is achieved at a confidence of around 0.47 with a value of 0.80. This suggests that the optimal confidence threshold for the model is at that point, where the balance between precision and recall is maximized in distinguishing between violence and non-violence.

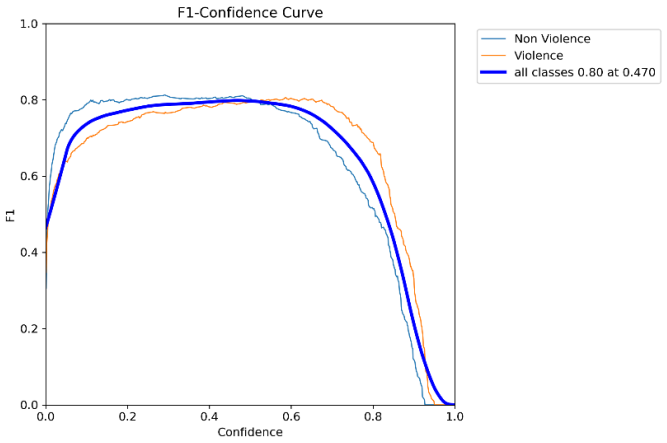

Figure 8. F1-Confidence Curve

3.2 Application Development

At this stage all the designs that have been made will be implemented into an application that will later be used by users to detect physical violence videos. Implementation of this application using the python programming language with flask as a web framework.

a. Result

The results of the physical violence detection application are as follows:

1) Home Page

This home page is the first display that will appear when the user opens the Physical Violence Video Detection System application. In this view, the system is designed with a simple and easy-to-use interface. There are two main components, namely:

a) Select Video button, which serves to upload a video file from the user's device. This video file will be used as input for the violence detection process. The supported video format is generally .mp4.

b) Detection button, which will activate the analysis process of the selected video. The system will detect the presence of physical violence in the video, and at a later stage will display the results in the form of a bounding box and confidence score if violence is detected.

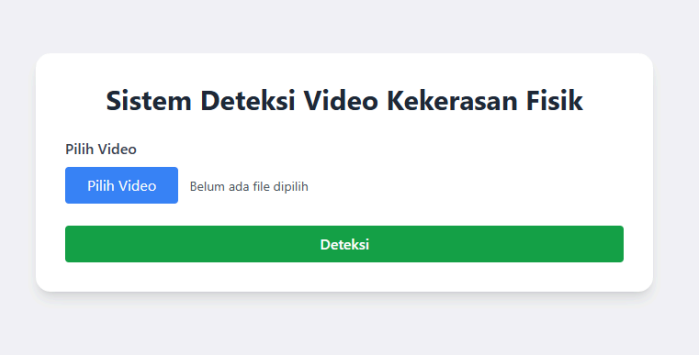

Figure 9. Home Page

2) Display when selecting a video

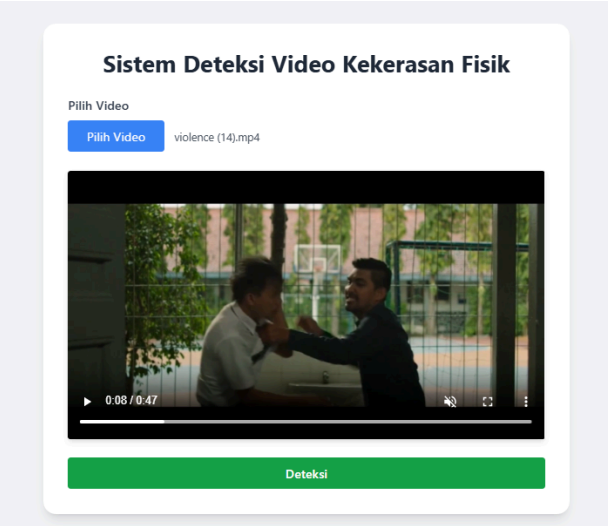

Figure 9. Preview After Video Upload

This is the system interface after the user has selected the video to be analyzed. Once the Select Video button is pressed and the user selects the file (in this example v_08.mp4), the system automatically displays a preview of the video just below the upload area. This preview gives the user a visual overview of the video to be detected, before starting the analysis process.

3) Display When the Detection Process is Complete

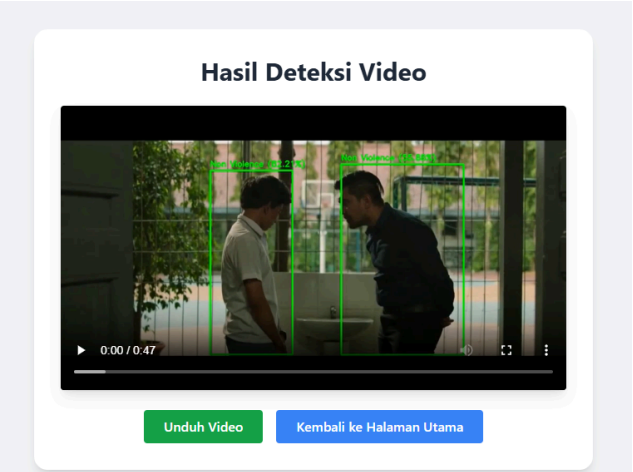

Figure 10. Display After Detection Process

This view shows the video detection result page after the system has completed the analysis process of the previously uploaded video. In this view, the system successfully detects the presence of physical violence, indicated by a red bounding box and the Violence (76.68%) label, which indicates the model's 76.68% confidence level in the violence detected in the frame. The annotated video is displayed so that users can visually verify the detection results. In addition, there are two main buttons, namely the Download Video button that allows users to download the detected video, and the Back

to Main Page button to return to the initial display and perform the detection process on another video.

3.3 Application Testing

In testing this application, researchers use testing with black box testing techniques and analysis tests.

1) Black Box Testing

In this test is done by running test scenarios, the testing process is carried out using five browsers. The following in Table 5 is a list of browsers used.

Table 5. Browser List and Version

Browser	Version
Microsoft Edge (ME)	136.0.3240.92
Google Chrome (GC)	136.0.7103.116
Mozila Firefox (MF)	139.0.1
Brave (BR)	137.1.79.118
Opera (OP)	119.0.5497.40

Table 6. Black Box Testing Result

Test Scenario	Expected result	Result				
		ME	GC	MF	BR	OP
Upload a valid file (mp4)	Displays Video Preview	✓	✓	✓	✓	✓
Upload an invalid file (other than mp4)	Displays an error message so that the user uploads a valid video	✓	✓	✓	✓	✓
Click the Detect Button (when you have uploaded a valid file)	Performs the detection process and results on the result page	✓	✓	✓	✓	✓
Click the Detect Button (when you have not uploaded a file)	Displays an error message so that the user uploads the file first	✓	✓	✓	✓	✓
Click the Download Video Button	The download process runs	✓	✓	✓	✓	✓
Click the Back to main page button	Displays the Main Page	✓	✓	✓	✓	✓

The results of Black Box Testing show that all the main features in the physical violence video detection application run according to their functions in five different browsers, namely Microsoft Edge, Google Chrome, Mozilla Firefox, Brave, and Opera. Each test scenario, such as uploading valid and invalid video files, running the detection process, displaying error messages, downloading the detected video results, and navigating back to the main page, was successfully executed with consistent and expected results across all browsers. This indicates that the application has high cross-platform compatibility and has optimally fulfilled basic functionality aspects.

2) Test Analysis

Analytical tests were conducted to further evaluate the model's ability to detect physical violence in videos. The test was conducted using 20 test videos consisting of 10 videos with violence and 10 videos without violence (non-violence). The symbol

(✓) is used to indicate that the model successfully detected physical violence, while the symbol (x) indicates that the model failed to detect violence in the video.

Table 7. Test Analysis

No	Test Video	Video Type	Browser				
			ME	GC	MF	BR	OP
1	Video 1	Violence	✓	✓	✓	✓	✓
2	Video 2	Violence	✓	✓	✓	✓	✓
3	Video 3	Violence	✓	✓	✓	✓	✓
4	Video 4	Violence	✓	✓	✓	✓	✓
5	Video 5	Violence	✓	✓	✓	✓	✓
6	Video 6	Violence	✓	✓	✓	✓	✓
7	Video 7	Violence	✓	✓	✓	✓	✓
8	Video 8	Violence	✓	✓	✓	✓	✓
9	Video 9	Violence	✓	✓	✓	✓	✓
10	Video 10	Violence	x	x	x	x	x
11	Video 11	Non violence	✓	✓	✓	✓	✓
12	Video 12	Non violence	✓	✓	✓	✓	✓
13	Video 13	Non violence	✓	✓	✓	✓	✓
14	Video 14	Non violence	✓	✓	✓	✓	✓
15	Video 15	Non violence	✓	✓	✓	✓	✓
16	Video 16	Non violence	✓	✓	✓	✓	✓
17	Video 17	Non violence	✓	✓	✓	✓	✓
18	Video 18	Non violence	✓	✓	✓	✓	✓
19	Video 19	Non violence	x	x	x	x	x
20	Video 20	Non violence	✓	✓	✓	✓	✓
Accuracy			0.9	0.9	0.9	0.9	0.9

The results of the analysis test show that the physical violence video detection system using the YOLOv9 model is able to work well and consistently in various browsers. Of the 20 test videos used, the system managed to correctly detect 18 videos, consisting of 9 violence videos and 9 non-violence videos. The accuracy rate obtained was 90% on five different browsers, namely Microsoft Edge, Google Chrome, Mozilla Firefox, Brave, and Opera. This proves that the system has stable performance and is independent of the browser platform used. This consistency is an important indicator in ensuring the reliability of the system when it is widely implemented by users from various devices.

CONCLUSION

Based on the evaluation results, the YOLOv9 algorithm shows superior performance compared to YOLOv8 in detecting physical violence in videos, with a precision value of 0.8096, recall 0.8665, F1-score 0.8363, and mAP 0.8117. These advantages make YOLOv9 more accurate and consistent and recommended for use in video detection systems. In addition, a web-based physical violence detection application was successfully developed using the Flask framework with YOLOv9 model integration. This application allows users to upload videos with no duration limit and automatically detects acts of physical violence, then displays the results in the form of annotated and downloadable videos. Although the duration of the video is not limited, the detection process will take longer as the length of the video increases because the

analysis is done per frame. The app's simple and user-friendly interface makes it practical for the public to use as a digital violence detection and prevention tool.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to the State University of Surabaya for the support and facilities provided during the research process. Special thanks are extended to our academic advisors for their valuable insights and continuous guidance. We also appreciate the assistance of all individuals who contributed to this study, either directly or indirectly, including those who supported the technical implementation and provided feedback on the manuscript. This research was made possible through the collective efforts and encouragement from our colleagues and families. No external funding or grants were received for this work.

REFERENCES

[1] World Health Organization, World report on violence and health, E. G. Krug, L. L. Dahlberg, J. A. Mercy, A. B. Zwi, and R. Lozano, Eds. Geneva, Switzerland: World Health Organization.

[2] Kemen PPPA, “Kemen PPPA, Komnas Perempuan dan Forum Pengada Layanan Luncurkan Laporan Sinergi Data Kekerasan Terhadap Perempuan,” Kementerian Pemberdayaan Perempuan dan Perlindungan Anak, 2024. [Online]. Available: https://www.kemenpppa.go.id/page/view/NTM1MA==

[3] F. S. Pratiwi, “Data jumlah kekerasan terhadap anak di Indonesia menurut jenisnya pada 2023,” *DataIndonesia.id*, 2023. [Online]. Available: https://dataindonesia.id/varia/detail/data-jumlah-kekerasan-terhadap-anak-di-indonesia-menurut-jenisnya-pada-2023

[4] E. K. Elsayed and N. M. Eldesouky, “Violence prediction in surveillance,” Appl. Comput. Syst., vol. 20, no. 3, pp. 1–16, 2024, doi: 10.35784/acs-2024-25.

[5] G. A. Sidik, “Deteksi tindak kekerasan dan perundungan pada anak berbasis YOLOv8 (You Only Look Once),” *Kohesi: Jurnal Sains dan Teknologi*, vol. 3, no. 9, pp. 71–80, 2024.

[6] S. A. Arun Akash, R. S. Moorthy, K. Esha, and N. Nathiya, “Human violence detection using deep learning techniques,” *Journal of Physics: Conference Series*, vol. 2318, no. 1, 2022. [Online]. Available: https://doi.org/10.1088/1742-6596/2318/1/012003

[7] K. Kar, Advanced computer vision with TensorFlow 2.x: Build advanced computer vision applications using machine learning and deep learning techniques. Birmingham, UK: Packt Publishing, 2020

[8] R. Li and Y. Wu, "Improved YOLOv5 wheat ear detection algorithm based on attention mechanism," *Electronics (Switzerland)*, vol. 11, no. 11, 2022. doi: 10.3390/electronics11111717.

[9] M. Hussain, "YOLO-v5 variant selection algorithm coupled with representative augmentations for modelling production-based variance in automated lightweight pallet racking inspection," *Big Data and Cognitive Computing*, vol. 7, no. 2, 2023. doi: 10.3390/bdcc7020046.

[10] R. H. Jatmiko and Y. Pristyanto, "Investigating the effectiveness of various convolutional neural network model architectures for skin cancer melanoma classification," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 23, no. 1, pp. 1–16, Nov. 2023. doi: 10.30812/matrik.v23i1.3185.

[11] E. Dio Bagus Sudewo, M. Kunta Biddini, A. Fadlil, E. D. Sudewo, M. Biddinika, and A. Fadlil, “DenseNet Architecture for Efficient and Accurate Recognition of Javanese Script Hanacaraka Character How to Cite: ‘DenseNet Architecture for Efficient and Accurate

Recognition of Javanese Script Hanacaraka Character,’” vol. 23, no. 2, pp. 453–464, 2024, doi: 10.30812/matrik.v23i2.xxx.

[12] N. W. S. Saraswati, I. W. D. Suryawan, N. K. T. Juniartini, I. D. M. K. Muku, P. Pirozmand, and W. Song, “Recognizing Pneumonia Infection in Chest X-Ray Using Deep Learning,” MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 23, no. 1, pp. 17–28, Oct. 2023, doi: 10.30812/matrik.v23i1.3197.

[13] X. Chen, J. Lv, Y. Fang, and S. Du, “Online Detection of Surface Defects Based on Improved YOLOV3,” Sensors, vol. 22, no. 3, Feb. 2022, doi: 10.3390/s22030817

[14] G. Gibran and S. V. Wahanggara, *RAD*, 2018.

[15] S. Maity, "YOLO (You Only Look Once) algorithm-based automatic waste classification system," *Journal of Mechanics of Continua and Mathematical Sciences*, vol. 18, no. 8, pp. 25–35, 2023.