Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2024. Vol. 3. No. 2; e-ISSN : 2808-4519

Sunda Script Detection Using You Only Look Once Algorithm

Daffa Arifadilah[1]*, Asriyanik[2], Agung Pambudi[3]

[1,2,3] *Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Muhammadiyah Sukabumi*
Jl. R. Syamsudin, S.H. No.50 Sukabumi, Jawa barat, Indonesia
daffa19af@ummi.ac.id [1], asriyanik263@ummi.ac.id [2], agungpambd@ummi.ac.id [3]

Abstract

The Sundanese script is a writing system used in the Sundanese language, one of the regional languages of West Java, Indonesia. This study investigates the use of the YOLO v8 algorithm for the real-time video detection of Sundanese script. Various versions of YOLO v8, including YOLO v8n, v8s, v8m, v8l, and v8x, were tested to determine the most effective model. After a comprehensive evaluation involving the analysis of mean Average Precision (mAP), F1-Confidence, and precision, the study selected the YOLO v8s model as the primary detection method. YOLO v8s demonstrated superior performance with the highest mAP of 98.835%, an F1-Confidence of 98%, and a precision of 76,2%. This choice was based on a balance between high accuracy and computational efficiency. The results indicate significant potential for object recognition technology in the learning and preservation of Sundanese script.

Keywords: *YOLO v8, Sundanese Script Detection, Real-Time Object Recognition, mAP, F1-Confidence, Precision*

1. Introduction

In the continuously evolving era of information technology, the challenges of preserving culture become increasingly complex. Utilizing technology is considered a crucial step in maintaining cultural sustainability. The use of technology in cultural preservation not only facilitates accessibility to information but also opens new opportunities to explore and gain a deeper understanding of cultural values Therefore, the integration of technology in preserving Sundanese script is viewed as a strategic move to ensure the continuity of ancestral heritage. Sundanese script is a form of writing originating from the Sundanese region, representing a letter system used in the past. Sundanese script can be divided into two forms: standard Sundanese script and ancient Sundanese script. Standard Sundanese script is an adapted and modified form based on ancient Sundanese script. However, due to the changing times and a lack of efforts in preserving cultural heritage, many members of the Sundanese community, if not the majority, no longer recognize or are aware of Sundanese script. Sundanese script plays a crucial role in reflecting the cultural identity of the Sundanese community. Nevertheless, the existence of Sundanese script is often overlooked in the context of modern information technology. The importance of preserving Sundanese script arises from the fact that the neglect of this language and writing can lead to the erasure of historical traces and a decline in understanding local culture. Therefore, this research aims to explore the issues surrounding the significance of Sundanese script and highlights the urgency of developing solutions capable of preserving the authenticity and meaning of this writing. As a solution to the challenges of preserving Sundanese script, this research opts for the utilization of the YOLO algorithm. YOLO, which stands for You Only Look Once, is an object detection algorithm known for its advantages in speed and accuracy. By implementing YOLO v8 in Sundanese script detection, it is anticipated to offer an effective solution for identifying and reconstructing script characters precisely and efficiently. The choice of YOLO as the primary approach also carries the expectation of supporting the use of cutting-edge technology in cultural preservation.

2. Research Method

This research seeks to create an application with the capability to recognize handwritten Sundanese script. To achieve this objective, a specific methodology is necessary, and the chosen approach for this study relies on the YOLO algorithm. Specifically, YOLO V8 is employed to achieve a satisfactory level of accuracy. The SEMMA method is utilized as a clear and easily understandable guide in data mining projects. Notably, within the model-building process, SEMMA's primary advantage is its centralized focus on model development, contributing to a more structured model creation process.

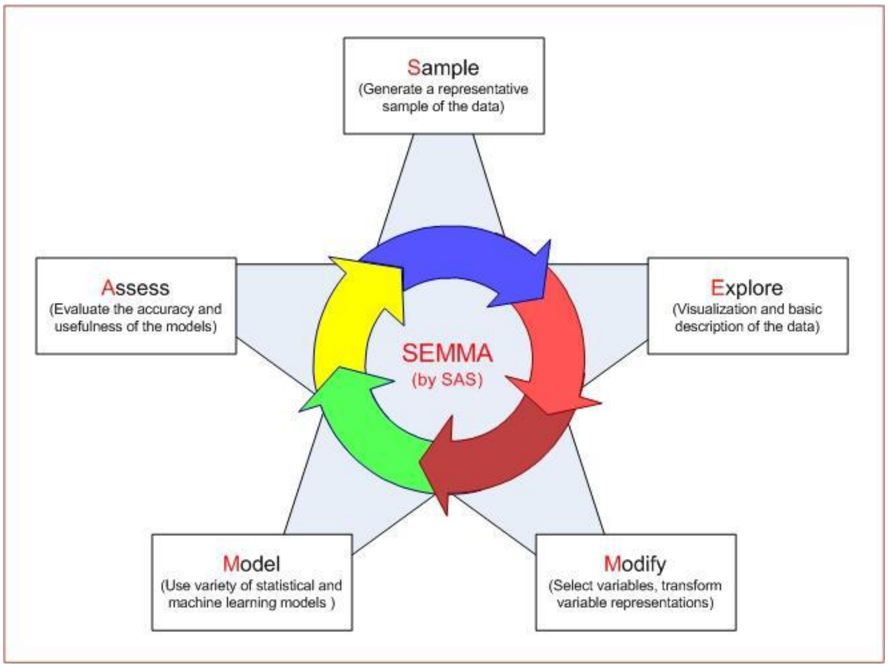

Fig. 1: SEMMA Methodology

2.1. Sample

The dataset was obtained from the GitHub repository "alfiarcm/aksara-sunda," consisting of 3,678 images and 4 CSV files containing labels in an unknown format. All images in each file consist of 30 types of Sundanese script characters.

2.2. Explore

In this stage, information about the acquired dataset is researched, a plan for modeling using YOLO is created, and the images are divided into 30 classes. Subsequently, each class is filtered to consist of 101 images, resulting in a total of 3,033 images.

2.3. Modify

From the prepared 3,033 images, labeling is performed using Roboflow for the YOLO V8 format. After labeling, the dataset is split as shown in the table below:

Table 1: Split Dataset

Split dataset	Percentage (%)	Total Images
Train	70	2,123
Validation	20	607
Test	10	303

2.4. Model

Following the implementation of the YOLO V8 model for the processed dataset, a comprehensive evaluation is conducted by performing experiments with various versions of YOLO v8, namely yolov8n, yolov8m, yolov8l, yolov8n, and yolov8x. The subsequent step involves a thorough comparison between these models based on several crucial metrics, including time hours, mean Average Precision (mAP), precision, and F1-score. The assessment aims to provide a comprehensive understanding of each model's performance across different aspects. Time hours are considered to gauge the efficiency of each version in terms of computational speed and resource utilization. mAP serves as a measure of the models' accuracy in object detection, while precision and F1-score offer insights into the models' ability to make correct positive predictions and achieve a balance between precision and recall.By meticulously comparing these metrics, a thorough analysis will be conducted to determine which YOLO v8 version performs optimally for the specific task of Sundanese script recognition. This comparative analysis will guide the decision-making process, allowing for the selection of the most suitable YOLO v8 model based on the specified evaluation criteria.

2.4. Asses

Assessment of the data is performed after creating the YOLO model to measure its performance. Evaluation includes several important metrics such as accuracy, precision, recall, F1-score, and mAP (mean Average Precision).

1. Precision: Indicates how accurate the model is in predicting positives.

2. Recall: Measures the model's ability to detect true positives.
3. F1-score: Harmonic mean between precision and recall, providing a holistic view of the balance between the two.
4. mAP: Indicates how well the model can provide precision in recognizing and placing objects correctly across different classes and confidence levels.

By assessing the model's performance using these metrics, one can understand how well YOLO can successfully detect objects in the used dataset. The evaluation results guide adjustments or improvements to the model.

3. Results and Discussion

3.1. Data collection (Sample)

In this stage, data collection was carried out by obtaining images from the GitHub repository alfircsm/aksara-sunda, which contains 3678 Sundanese script images and 4 label files. The author extracted the images from the aforementioned GitHub repository. Subsequently, folders were created for each unique value present in these images, corresponding to their respective letter types. The total number of classes obtained was 30.

3.2. Data Description (Explore)

There were several images with names and characteristics that did not match the pre-determined classes. Therefore, a filtering process was implemented to maximize the data quality. From the initial dataset of 3678 images, after the filtering process, the total dataset was reduced to 3033 images with a total of 30 classes. The following table represents the classes of each dataset.

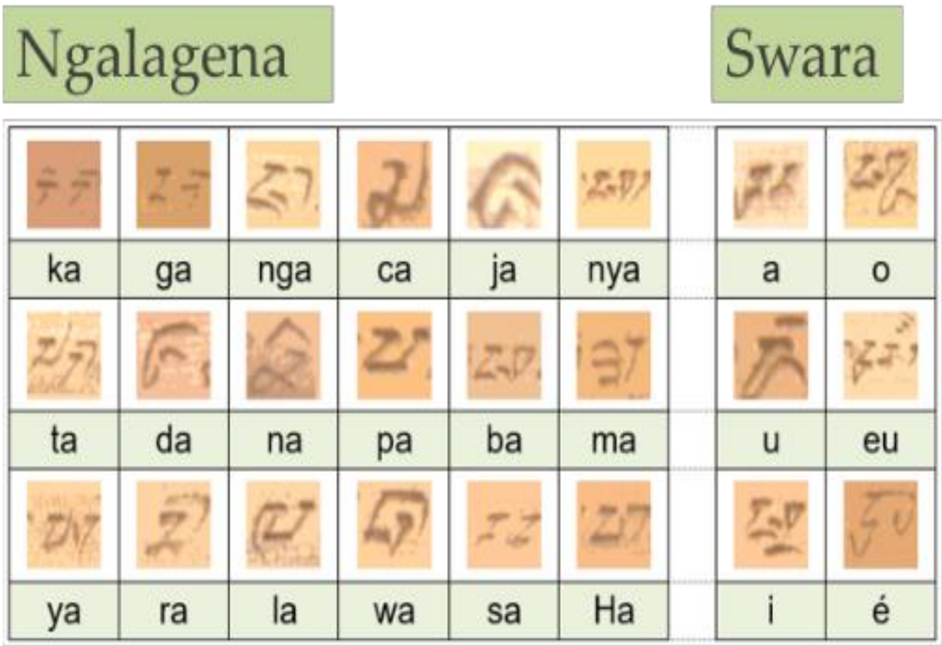

Fig. 2: Sunda Script

3.3. Modification of Data (Modify)

This process was carried out through the Roboflow platform and applied to all previously processed images. Each image was assigned a single label to determine the object's position. This step is crucial to enable YOLO v8 to identify the classes it needs to detect. It serves as a prerequisite before progressing to the next stage, which is the modeling phase.

3.4.1. Data Preprocessing

In the next stage, preprocessing is carried out, where the image data obtained earlier is transformed to meet the requirements for implementation in YOLOv8. Several adjustments are applied to prepare the data for optimal utilization in YOLOv8. The following are some of the applied processes:

1. Auto-Orient
   When capturing an image, it may contain metadata that specifies the orientation to be displayed relative to how pixels are arranged on the disk. This instruction accelerates image encoding at the time of capture, allowing the camera to efficiently sample data from its sensor without unwanted artifacts. This means that most cameras store exactly the same pixel data, whether the camera

is oriented in landscape or portrait mode. The camera only flips slightly to signal to the viewer whether to display the pixels as they are or rotate them 90 or 180 degrees when displaying the image.

2. Resize stretch to 640x640
   The term "resize" refers to adjusting the dimensions of an image according to the requirements, and it can be done to accommodate the location of annotations or labels on the image, eliminating the need for re-labeling. On the other hand, "stretch to" in this context means stretching the image to pixel dimensions specified for the study, which is 640x640. This involves stretching or resizing the image to reach these pixel dimensions.

3.4. Data Modeling (Model)

In this stage, the YOLOv8 architecture will be implemented to create a model for object detection. The goal is to determine the best model based on processing speed and accuracy. Below is a table listing the YOLO models used for object detection with the Open Image v7 dataset:

Table 2: YOLO v8 Types

Model	*Size (pixels)*	*MAPval 50-95*	*SpeedCPU ONNX (ms)*	*SpeedCPU TensorRT (ms)*	*Params (M)*
YOLO v8n	640	18.4	142.4	1.21	3.5
YOLO v8s	640	27.7	183.1	1.4	11.4
YOLO v8m	640	33.6	408.5	2.26	26.2
YOLO v8l	640	34.9	596.9	2.43	44.1
YOLO v8x	640	36.3	860.6	3.56	68.7

All these models will undergo testing to determine the best-performing model by identifying the highest mAP value, considering the number of epochs that will be adjusted based on the GPU memory capacity on Google Colab, which is 77.1 GB. Afterward, the chosen model will be saved in the h5 format for later use in a web application that can access the device's camera to detect Sundanese scripts.

YOLOv8 comes in five scale versions: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra-large). These versions vary in terms of model size and complexity. This research utilizes all YOLOv8 models, and the selection will be based on the highest precision, recall, and mAP50 values with the hyperparameter configuration being an input size of 640 x 640, 15 epochs, and a batch size of 8. The model configurations are described in the following table.

Table 3: Model Configuration

Configuration	***Value***
Size	640x640
epoch	15
batch	8
workers	42
seed	4

3.4.1. Result Model

Results from all models that have undergone the training stages, with selected parameters such as F1-Score, Confidence, Precision, Recall, mAP 50, and Total Time (in hours):

Table 4: Results for Each YOLOv8 Model Type

Model	*F1-Confidence*		*precision*	*recall*	*mAP 50*	**Total Time (in hours)**
YOLO v8n	0.97	0.769	0.96933	0.9746	0.98547	0.233
YOLO v8s	0.98	0.762	0.98088	0.98172	0.98835	0.245
YOLO v8m	0.98	0.803	0.97958	0.9776	0.98753	0.362
YOLO v8l	0.97	0.794	0.97909	0.96822	0.98606	0.549
YOLO v8x	0.98	0.777	0.98212	0.97508	0.98916	0.886

The graph displaying F1-Confidence scores for all models is as follows.

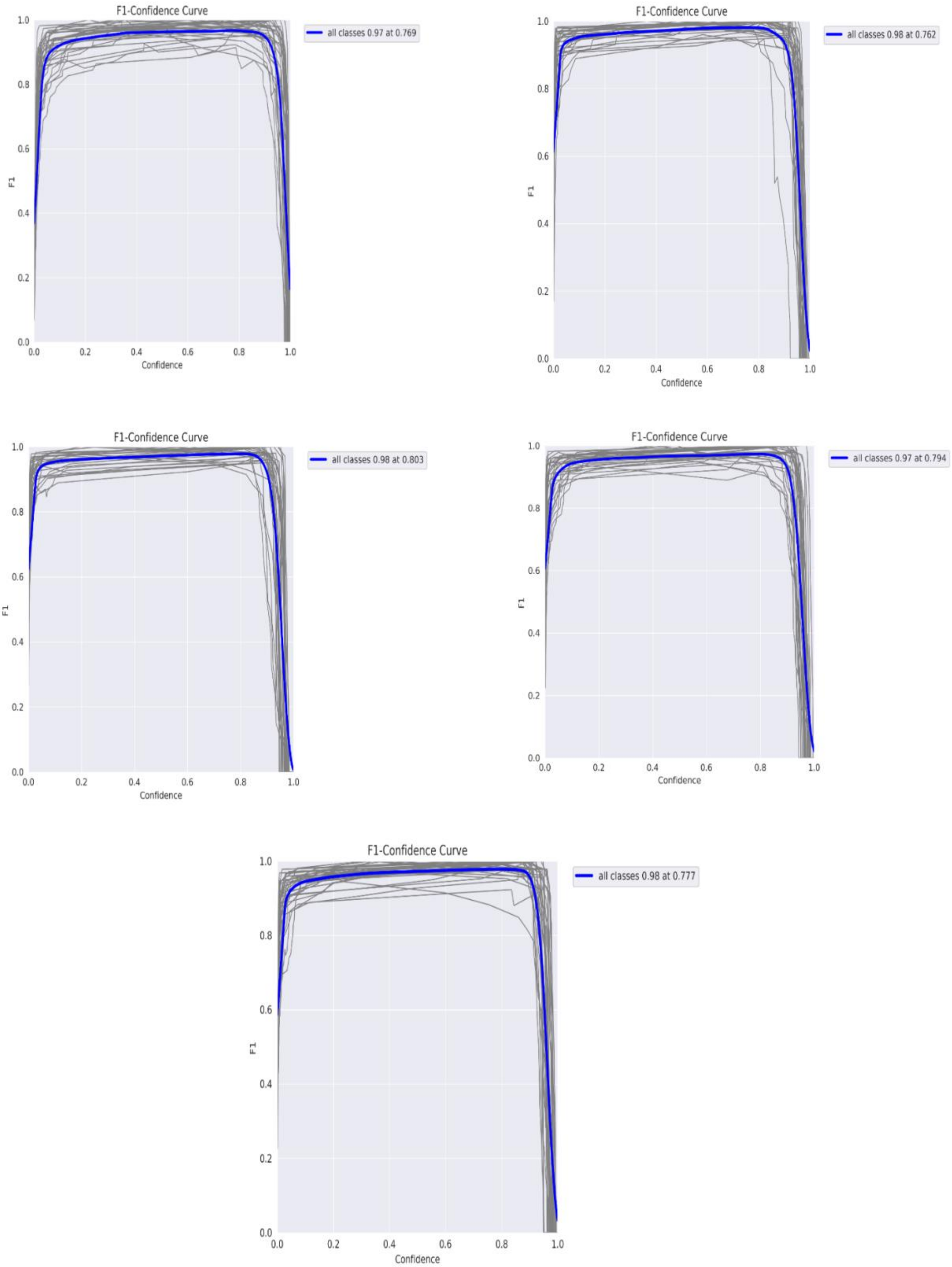

Fig. 3: Graph for F1-Condifence Score each YOLO v8 model type

3.1. Evaluation (Assess)

In this study, we conducted a comprehensive evaluation of various versions of the YOLOv8 model designed for Sundanese script detection. These models include YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x, each with distinct characteristics and complexities.

The evaluation focused on key metrics such as F1-Confidence, precision, recall, and mAP (mean Average Precision), aiming to determine the most effective model in recognizing and classifying Sundanese scripts with high accuracy.
The evaluation results indicate that each model exhibits strengths in various aspects. Overall, the YOLOv8 models demonstrate outstanding capabilities in detecting Sundanese scripts, with high mAP values signifying excellent accuracy.

- YOLOv8n: As the lightest model, it achieves an F1-Confidence of 0.97, precision of 0.769, and the highest mAP value of 0.98547.
- YOLOv8s: Showing improvement, it achieves an F1-Confidence of 0.98, precision of 0.762, and the highest mAP of 0.98835.
- YOLOv8m: Records an F1-Confidence of 0.98, precision of 0.803, and the highest mAP of 0.98753.
- YOLOv8l: Offers an F1-Confidence of 0.97, precision of 0.794, and the highest mAP of 0.98606.
- YOLOv8x: The largest and most complex version attains an F1-Confidence of 0.98, precision of 0.777, and the highest mAP of 0.98916.

ased on a comprehensive evaluation of various YOLOv8 model versions in Sundanese script detection, the researchers decide to use the YOLOv8s model as the selected model. This decision is grounded in the balance between accuracy, processing speed, and resource usage. Although the YOLOv8x model shows slightly higher mAP, YOLOv8s offers nearly equivalent performance with more efficient resource utilization.

3.5. Testing stage

3.5.1. Image-Based Testing

Testing through images was conducted by utilizing six types of Sundanese scripts in each test, repeated three times through the Roboflow website.

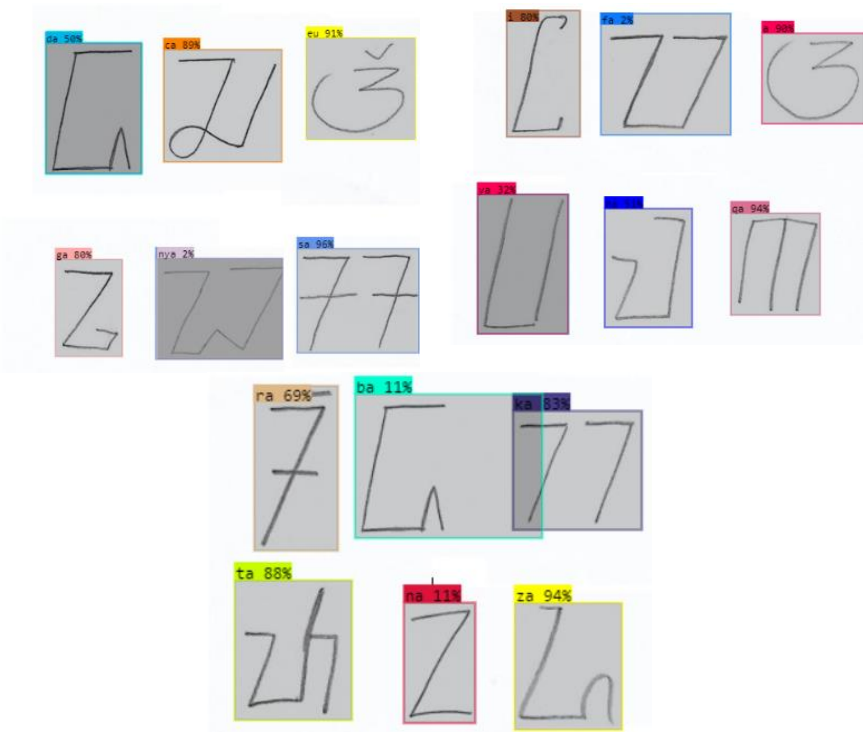

Figure 4: Result for Image-Based Testing

3.5.1. Video-Based Testing

This testing was carried out through the Roboflow website. The video was captured using the research device and utilized the same images as those used in the image-based testing.

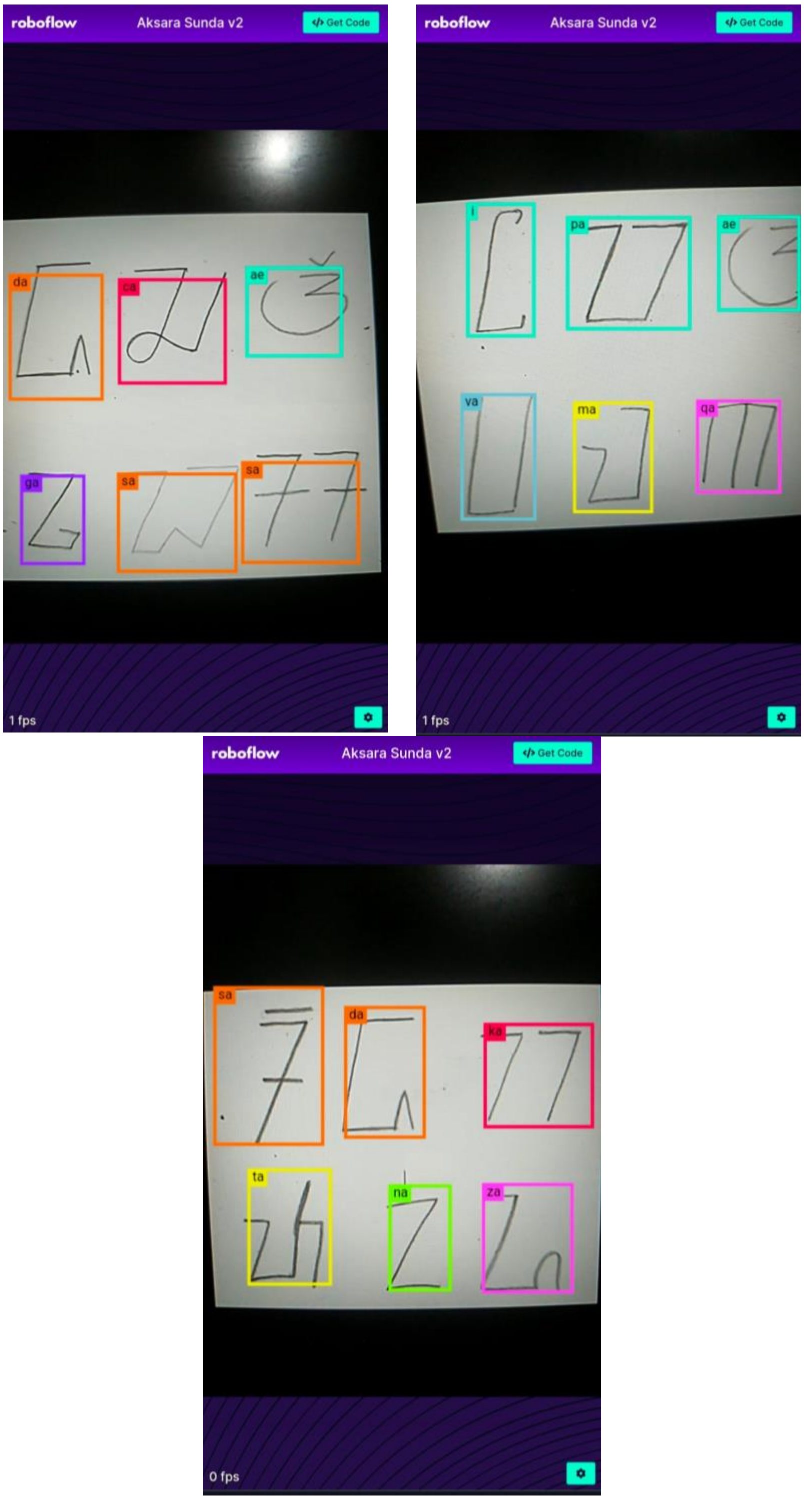

Figure 5: Result for Video-based Testing

4. Conclusion

Concluding our investigation, the focal point of this study was the implementation of the YOLOv8s model for the real-time detection of Sundanese scripts in video streams. The results of our evaluation underscore the exceptional proficiency exhibited by the YOLOv8s model, boasting an impressive mean Average Precision (mAP) score of 98.83%. This outstanding level of accuracy highlights the model's effectiveness in real-time object detection, specifically tailored for Sundanese scripts. With a robust performance, including an F1-Confidence of 98% and a precision rate of 76.2%, the model underscores its reliability in precisely and efficiently identifying Sundanese scripts. These findings not only emphasize the practical significance of the YOLOv8s model in real-world applications but also contribute valuable insights to the field of Sundanese script detection in dynamic video environments.

References

[1] Anita Chaudhari, Brinzel Rodrigues, S. More. (2016a). Melestarikan Aksara Sunda Dengan Aplikasi Multimedia. Ucv, I(02), 390–392. http://dspace.unitru.edu.pe/bitstream/handle/UNITRU/10947/Miñano Guevara%2C Karen Anali.pdf?sequence=1&isAllowed=y%0Ahttps://repository.upb.edu.co/bitstream/handle/20.500.11912/3346/DIVERSIDAD DE MACROINVERTEBRADOS ACUÁTICOS Y SU.pdf?sequence=1&isAllowed=

[2] A Baidillah, I., Darsa, Undang. A., Abdurahman, O., Permadi, T., Gunardi, G., Suherman, A., Ampera, T., Purba, H. S., Nugraha, D. T., & Sutisna, D. (2008). Direktori Aksara Sunda untuk Unicode. Pemerintah Provinsi Jawa Barat Dinas Pendidikan Provinsi Jawa Barat.

[3] Bismantoko, D. H., Sriyanto, S., & Budiawan, W. (2015). Sistem Informasi Transportasi Umum Terintegrasi di Kota Semarang Menggunakan Algoritma Depth First Search (Dfs). Industrial Engineering Online Journal, 4(2).

[4] Fatah, H., Ichsan, N., Wahyuni, T., Ernawati, E., & Indriyanti, I. (2020). RANCANG BANGUN PROGRAM APLIKASI PEMBELAJARAN AKSARA SUNDA BERBASIS ANDROID. SISTEMASI, 9(2). https://doi.org/10.32520/stmsi.v9i2.770

[5] Holida, S. M., Alawiyah, T., & Sutisna, H. (2016). PENERAPAN ANIMASI INTERAKTIF DALAM PENGENALAN AKSARA SUNDA. Jurnal Informatika, 1(2). https://doi.org/10.31311/ji.v1i2.39.

[6] Indra, I. (2019). Apa itu URL? Pengertian URL Beserta Fungsinya. In Niagahoster Blog. https://www.niagahoster.co.id/blog/apa-itu-url/.

[7] Kanani, P., & Padole, M. (2019). Deep learning to detect skin cancer using google colab. International Journal of Engineering and Advanced Technology, 8(6). https://doi.org/10.35940/ijeat.F8587.088619

[8] Krisdiawan, R. A., Fitriani, A., & Budianto, H. (2022). Penerapan Algoritma Recursive Backtracking Sebagai Maze Generator Pada Game Labirin Aksara Sunda. Media Jurnal Informatika, 14(1). https://doi.org/10.35194/mji.v14i1.2326

[9] Lin, Q., Ye, G., Wang, J., & Liu, H. (2021). RoboFlow: a Data-centric Workflow Management System for Developing AI-enhanced Robots. Proceedings of Machine Learning Research, 164, 1789–1794.

[10] Liu, R., & Ren, Z. (2021). Application of Yolo on Mask Detection Task. 2021 IEEE 13th International Conference on Computer Research and Development, ICCRD 2021. https://doi.org/10.1109/ICCRD51685.2021.9386366

[11] Lutz O'reilly, M., Cambridge, B. •, Farnham, •, Koln, •, Paris, •, Sebastopol, •, Taipei, •, & Tokyo, •. (2019). Programming Python

[12] Mulyana, D. I., & Rowis, M. A. I. (2022). Optimization of Text Mining Detection of Tajweed Reading Laws Using the Yolov8 Method on the Qur'an. QALAMUNA: Jurnal Pendidikan, Sosial, Dan Agama, 14(2), 1089–1110. https://doi.org/10.37680/qalamuna.v14i2.3866

[13] Paulus, E., Hadi, S., Suryani, M., Suryana, I., & Simanjuntak, Y. D. (2019). Evaluating Ancient Sundanese Glyph Recognition Using Convolutional Neural Network. Journal of Physics: Conference Series, 1235(1). https://doi.org/10.1088/1742-6596/1235/1/012063

[14] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-December. https://doi.org/10.1109/CVPR.2016.91

[15] Sari, D. P., & Mirza, A. H. (2022). THE DETECTION OF FACE RECOGNITION AS EMPLOYEE ATTENDANCE PRESENCE USING THE YOLO ALGORITHM (YOU ONLY LOOK ONCE). Jurnal Darma Agung, 30(3). https://doi.org/10.46930/ojsuda.v30i3.2187.

[16] Setiawan, R. (2021). Flowchart Adalah: Fungsi, Jenis, Simbol, dan Contohnya. In Dicoding Blog. https://www.dicoding.com/blog/flowchart-adalah/

[17] Tambunan, S., & Naibaho, P. (2021). DETEKSI OBJEK AKSARA BATAK TOBA MENGGUNAKAN FASTER R-CNN dan YOLOV3 DOKUMEN TUGAS AKHIR.

[18] Terven, J. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. In Machine Learning and Knowledge Extraction (Vol. 5, Issue 4, pp. 1680–1716). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/make5040083

[19] Wang, J., Li, L., & Zeller, A. (2020). Better Code, Better Sharing: On the Need of Analyzing Jupyter Notebooks. Proceedings - 2020 ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results, ICSE-NIER 2020, 53–56. https://doi.org/10.1145/3377816.3381724

[20] Wang, X., Zheng, S., Zhang, C., Li, R., & Gui, L. (2021). R-yolo: A real-time text detector for natural scenes with arbitrary rotation. Sensors (Switzerland), 21(3). https://doi.org/10.3390/s21030888

[21] Wang, Y., Zou, H., Yin, M., & Zhang, X. (2023). SMFF-YOLO: A Scale-Adaptive YOLO Algorithm with Multi-Level Feature Fusion for Object Detection in UAV Scenes. Remote Sensing, 15(18). https://doi.org/10.3390/rs15184580

[22] Xiao, L., Zhou, P., Xu, K., & Zhao, X. (2021). Multi-directional scene text detection based on improved yolov3. Sensors, 21(14). https://doi.org/10.3390/s21144870