Jurnal Sistem Informasi dan Informatika (Simika) *P-ISSN: 2622-6901*
Vol 8. No. 2 Tahun 2025 *E-ISSN: 2622-6375*

AI-BASED FACIAL DE-IDENTIFICATION FOR CHILDREN'S DIGITAL PRIVACY

Komang Putra Satria Negara[1], Ida Nurhaida[2,3]
[1,2]Program Studi Informatika, Universitas Pembangunan Jaya
[3]Center of Urban Studies, Universitas Pembangunan Jaya
Jl. Cendrawasih Raya Blok B7/P Bintaro Jaya, Sawah Baru, Ciputat, Tangerang Selatan, Provinsi Banten, Indonesia
e-mail: [1]**komang.putrasatria@student.upj.ac.id,** *[2,3]ida.nurhaida@upj.ac.id

Abstract
Social media has become an inseparable part of daily life in today's digital era. Many parents frequently share photos of their children online, exposing them to risks related to privacy and security. This research addresses such issues by developing an Android-based facial de-identification application that utilises the YOLOv8 algorithm to protect minors' privacy. The methodology involves several stages: data collection, pre-processing, model training, and application development. The dataset includes over 2,889 images of children, which were augmented to enhance its size and diversity. YOLOv8, a state-of-the-art object detection algorithm, was trained with these images to achieve high precision and recall in identifying children's faces. The developed application integrates the YOLOv8 model within a user-friendly interface built with Flutter. Results indicate that YOLOv8 effectively detects children's faces with a high precision of 95%, accuracy of 88%, recall of 92%, and mAP50 of 0.977. While the model demonstrates strong performance on training data, there is room for improvement on unseen data. By leveraging YOLOv8 and providing an accessible mobile application, the work allows parents to protect their children's identities online. The application mitigates risks of unauthorised use and exploitation of children's images by enabling facial de-identification, thus promoting safer online practices for families.

Keyword: *facial de-identification, object detection, privacy, security*

INTRODUCTION

Human trafficking remains a major global issue, with statistics highlighting its severe prevalence. According to the Global Report on Trafficking in Persons 2024 by the United Nations Office on Drugs and Crime (UNODC), the number of detected human trafficking victims worldwide increased by 25% in 2022 compared to 2019, with children now accounting for 38% of total detected victims (*United Nation Office on Drugs and Crime*, 2024). The report also revealed a 31% rise in detected child victims during the same period, with a notable 38% increase among girls. In Indonesia, data from the Information System for the Protection of Women and Children (SIMFONI PPA) recorded 206 child victims of human trafficking in 2023 (*Hari Dunia Anti Perdagangan Orang 2024, Menteri PPPA: Lawan Dan Akhiri Segala Bentuk Perdagangan Orang*, 2024). Another report from Indonesia's Ministry of Women's Empowerment and Child Protection stated that 97% of human trafficking victims were women and children. These statistics highlight the vulnerabilities of children, particularly as their private lives are increasingly exposed online (*Hari Dunia Anti Perdagangan Orang 2024*, 2024).

Social media has become an essential part of people's lives, and in Indonesia, the majority of the 63 million active internet users, 95%, use social media, making the country the fourth-largest Instagram user base in the world (Annur, 2023). This widespread usage has resulted in a massive amount of personal content shared online, including photos and videos of children. Frequent sharing of children's pictures by parents, often without their knowledge or consent, poses significant risks to their privacy and safety. Research shows that 75% of parents use social media to share monthly content (Livingstone et al., 2024). These images can be exploited for malicious purposes, such as cyberbullying, identity theft, kidnapping, and even

human trafficking. One case that highlights these dangers involved photos of the children of Indonesian celebrity Mona Ratuliu and the niece of Audi Marissa, which were found on an Instagram account linked to a human trafficking syndicate (Endra & Ratri, 2021). Indonesia faces challenges in protecting children from digital exploitation due to the widespread use of the internet and social media. The technology-based solutions are required for proactively identifying and mitigating potential threats. This research aims to develop an Android application leveraging the YOLOv8 deep learning algorithm to enhance children's privacy protection before sharing content on social media.

Deep learning, a branch of machine learning based on artificial neural networks, utilises multiple network layers to achieve advanced performance in fields like speech recognition, visual object detection, drug discovery, and genomics (LeCun et al., 2015). Computer vision, a subfield of deep learning, enables computers to mimic human visual processing by interpreting and understanding visual data (Zhang, 2023). YOLO (You Only Look Once) is a deep learning-based computer vision algorithm for object detection that processes detection in a single step. YOLO is a Convolutional Neural Network (CNN) designed to recognize image patterns. CNNs are widely used in various research applications, such as identifying mango species based on leaf characteristics (Wisak et al., 2024). Another application of CNN is optimizing image classification for different types of garbage (Permana et al., 2022). Unlike earlier methods relying on sliding windows or region proposals with separate classification stages, YOLO simplifies the process for efficiency and accuracy (Terven & Esparza, 2023). Since its introduction in 2015, YOLO has evolved from YOLOv1, which faced challenges with unusual object aspect ratios and detecting adjacent objects, to YOLOv8, offering enhanced performance through improved architecture and detection accuracy. YOLOv8 features three core components: CSPDarknet53 as the backbone for feature extraction, a neck for detecting objects of varying sizes, and a head for generating final predictions like bounding boxes, confidence scores, and classifications. It uses loss functions such as localisation loss for bounding box accuracy, confidence loss for object presence, and classification loss for correct labelling. Studies highlight YOLO's advancements; YOLOv3 outperformed SSD and Faster R-CNN on the Microsoft COCO dataset (Srivastava et al., 2021), while YOLOv4 surpassed F-RCNN and SSD in vehicle recognition (Kim et al., 2020). In 2023, YOLOv8 demonstrated further refinement in detection and accuracy.

The analysis of different YOLO object detection models shows that YOLOv8 performs better than previous versions (YOLOv7, v6.2.0, and v5-7.0) in two ways. First, it achieves higher accuracy while using fewer parameters, meaning it's more efficient in terms of model size. Second, when tested on an A100 GPU, it delivers faster processing speeds while maintaining better accuracy than older versions. Figure 1 describes the performance comparison between YOLOv8 and previous versions.

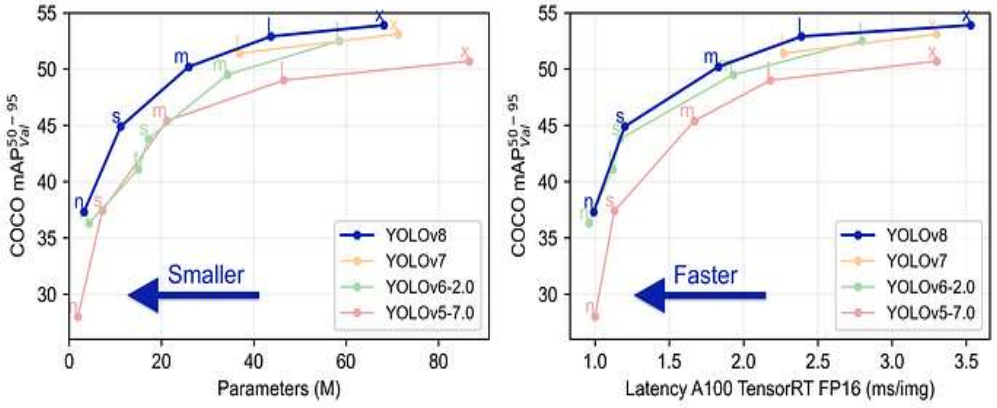

Figure 1. YOLO Comparision with Previous Versions
Source: (Takase, 2023)

Earlier research has explored using deep learning to classify people into age groups. Thaneeshan and colleagues built a model that predicted both age and gender, but only achieved 57.60% accuracy for age estimation. This low accuracy shows how challenging it is to tell apart overlapping age groups (Thaneeshan et al., 2022). In another study, researchers used MTCNN and VGG-Face techniques to sort faces into different age ranges. However, their method had trouble correctly classifying adults because facial features look very similar between nearby age groups. The problem was made worse by having too few training examples, which caused overfitting (Gyawali et al., 2020). A third study used convolutional neural networks (CNNs) with a large face database called All-Age Face (AAF). The researchers used the FaceNet model to extract facial features and achieved better results - 84.90% accuracy on their training data and 85.12% on their test data (Mustapha et al., 2021).

YOLOv8 has been utilised in various studies due to its high performance and accuracy in object detection. For example, YOLOv8 has been applied to brain tumour segmentation in MRI images, achieving a box accuracy of 86% precision, 87% recall, 89% mAP50, and 71% mAP50-95 (Ardiansyah et al., 2024). The algorithm has also been employed to detect dangerous weapons in images, resulting in a precision of 84%, recall of 77%, mAP of 84%, and an F1-Score of 88% (Maulana et al., 2024). YOLOv8's capability for human object detection was analysed using modified architecture models trained four times. The results showed mAP values of 76 for the default architecture, 66 for Model 1, 81 for Model 2, and 80 for Model 3 (Setiyadi et al., 2023). YOLOv8 has also been implemented in robotics. A study on object detection robots demonstrated excellent performance, with precision and recall values approaching 1 for all object classes. Specifically, chairs, humans, and trash bins achieved precision and recall values of 0.994 or higher, with mAP50-95 values of 0.891, 0.874, and 0.894, respectively. Bottles and buckets also showed good results, with mAP50-95 values of 0.857 and 0.90 (Rasjid et al., 2024). Furthermore, YOLOv8 has been applied in mobile applications, such as Bali's Endek fabric classification. The evaluation showed a precision of 0.943, recall of 0.954, mAP50 of 0.956, and mAP50-95 of 0.8 (Rahaditya Kusuma et al., 2024). Another application involved a human entry and exit counter system in buildings. The implementation achieved a precision of 79.6%, a recall of 61.7%, and a mean average precision (mAP) of 72.3% (Fahrezi & Widiyanto, 2024).

RESEARCH METHOD

Figure 2 illustrates the research flowchart used as a guide in preparing this research.

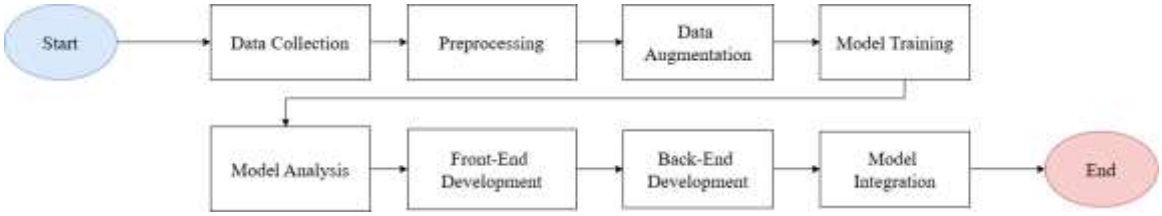

Figure 2. Research Flow

a. Data Processing

1. Data Collection

   The collected data consists of 5.315 images, including photos of babies, families, children, and other pictures featuring minors. These photos were sourced from free stock image providers such as Unsplash.com, Pexels.com, Pixabay.com, and open-source repositories. The lowest image quality available is selected to lower the computational costs. Figure 3 shows an example of the dataset used in this research.

Figure 3. Dataset Example

2. Data Preprocessing

   All collected images underwent preprocessing to ensure consistency. First, Images were resized to 640x640 pixels using bicubic interpolation to maintain aspect ratio while fitting within a standardised black frame.

3. Data Annotation

   Roboflow is used to annotate images or place bounding boxes around identified children's faces. In the annotation process, a single class "children" was defined to encompass all individuals under 18 years, regardless of specific age subcategories. Bounding boxes were drawn tightly around visible facial regions of individuals under 18 years An example of this annotation process can be seen in Figure 4.

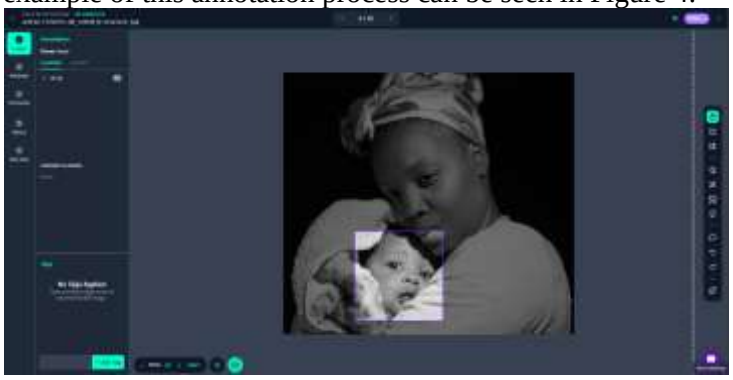

Figure 4. Data Annotation in Roboflow

4. Data Augmentation

   The images obtained from the stock photo sites have varying resolutions, so they are resized to fit inside a 640x640 black frame. The augmentation processes include flipping (horizontal and vertical), rotating by 90 degrees, rotating between -12 to 12 degrees, shearing, and applying blur up to 0.9px. This annotation process increased the dataset from 5315 images to 12757 images. The dataset is split into 11163 images for training, 1063 for validation, and 531 for testing. Table I explains the effects of the various augmentation techniques.

Table 1. Augmentation Techniques

Technique	Effects on Dataset
Rotation	Can lead to improved recognition across orientations.
Shearing	Creates new perspectives and distortions, leading to better model adaptability to skewed
Flipping	Improving object recognition performance for symmetrical features.
Blurring	Introducing variations in image quality, making the model more robust to noise and focus variations

b. Training Model

The training process uses a YOLOv8 model version with a nano size and 80 training epochs. After each epoch, precision, recall, mAP60-95, dfl_loss, cls_loss, and box_loss are displayed.

c. Model Analysis

Below are the metrics used in analyizing the model‘s performance.

1. *Precision and Recall*
   Precision is the proportion of true positive predictions out of the total number of true positives and false positives, while recall is the proportion of true positive predictions compared to the total number of positive samples in the dataset. (Chumachenko et al., 2022).
2. *Mean Average Precision (mAP)*
   Mean Average Precision is calculated as the mean of the Average Precision (AP) values across all the classes in the model (Solawetz, 2020) . This metric calculates the average precision for each class and then computes the mean across all classes.
3. *Confusion Matrix*
   As shown in Table 2, the Confusion matrix is a two-dimensional matrix that summarises the classification performance of a classifier on test data, indexed in one dimension by the actual class of an object and in the other dimension by the class predicted by the classifier (Ting, 2010). The confusion matrix will be utilized to compute accuracy, precision, recall, and the F1 score derived from equations (1), (2), (3), and (4), respectively. (Sharma, 2023).

$$\text{Precision} = \frac{\text{TP}}{\text{TP+ FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{2}$$

$$\text{Accuracy} = \frac{\text{Correct Prediction}}{\text{All Prediction}} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \tag{3}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision + Recall}} \tag{4}$$

Table 2. Confusion Matrix

		Assigned Class	
		Positive	**Negative**
Actual Class	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

d. Front-end Application Development

In this stage, the application interface is designed and implemented using the Dart programming language and Flutter as its framework. Flutter is a free and open-source framework developed by Google for building cross-platform applications with a single codebase. The interface prototype is shown in Figure 4. Users can choose images from two different sources: gallery and camera.

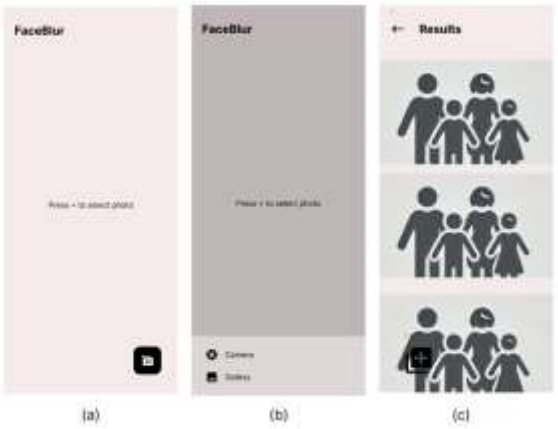

Figure 5. Application Interface Prototype: (a) Home Screen (b) Select Image (c) Result Page

e. Back-End Application Development

Python and Flask are used as the back-end programming language and framework. In this stage, a function is created to receive an image and generate an output where all children's faces are anonymised using Gaussian blur with the help of the OpenCV library. The next step is to create an API endpoint that receives an image input from the user and then sends a response containing a link to the processed image.

f. Model Integration

The Android app integrates the YOLOv8 model by consuming a REST API built with Python and Flask. When a user uploads an image, the app sends the image to the back-end via an HTTP POST request. The back-end processes the image using the YOLOv8 model, which detects children's faces and returns the image with blurred faces as the response. The app then receives the processed image and displays it to the user.

RESULTS AND DISCUSSION

The training process of YOLOv8 was performed using Python 3.10.13 and a Tesla T4 GPU for 3 hours. This process resulted in several findings.

a. Performance Metrics

The precision curve shows a rapid improvement in the ability to correctly identify children's faces, with stabilisation at a high value of approximately 0.95. Similarly, a steady increase in recall is shown, stabilising around 0.92, which indicates the successful identification of the vast majority of children's faces in the dataset. Further insight is provided by the Mean Average Precision (mAP). A rapid reach to nearly 0.98 for mAP@50, which measures accuracy at a lenient overlap threshold, shows strong performance in general detection. The mAP@50-95 score, which measures how precisely the model draws the bounding box around a face, was about 0.67. This result indicates that while the model is effective at finding faces, it is less accurate at defining their exact outlines, particularly in complex images with overlapping objects. Figure 5 illustrates training performance metrics for YOLOv8: Precision, Recall, mAP@50, and mAP@50:95.

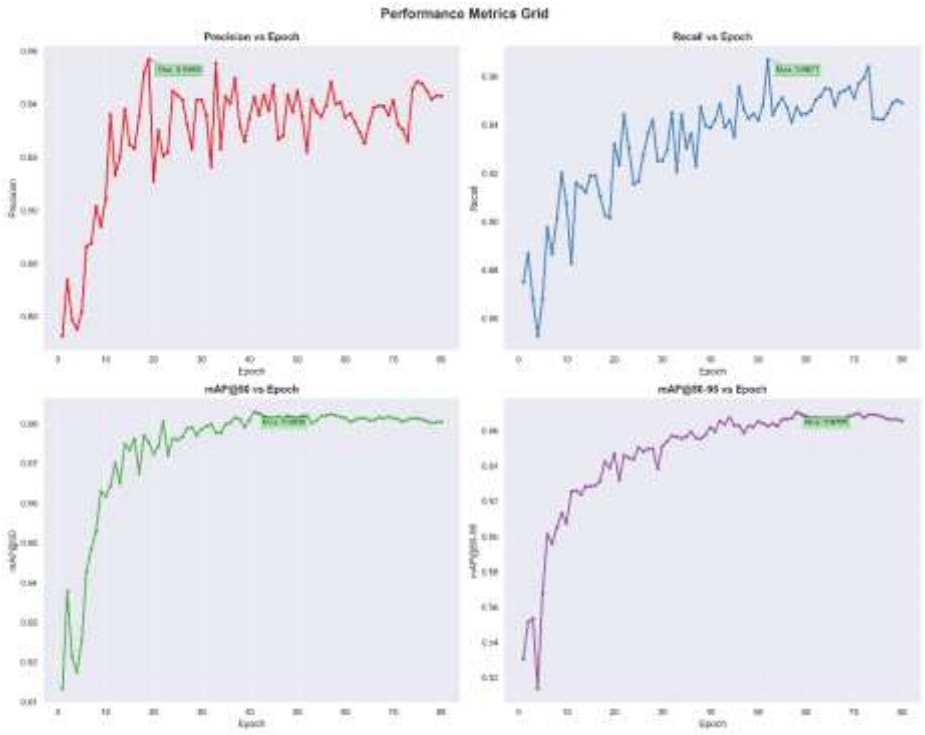

Figure 6. Training performance metrics for YOLOv8

The mAP@50 metric rises from 0.6 to 0.65 within the first 20 steps and then stabilises, showing a small improvement of localisation and classification performance at a lenient IoU threshold of 0.5. The mAP@50-95, which uses stricter IoU thresholds, increases slowly and stabilises around 0.65–0.66. However, the lower mAP@50-95 values suggest room for improvement in handling complex or overlapping objects. Table 3 illustrates the model's performance at various epochs.

Table 3. Performance Metrics at Various Epochs

Epoch	Metrics			
	Precision	**Recall**	**mAP50-95**	**mAP50**
10	0.90432	0.90756	0.60754	0.96179
30	0.94169	0.92523	0.6513	0.97864
50	0.94497	0.94194	0.66529	0.98139
70	0.9414	0.95111	0.66885	0.98146
80	0.94307	0.94908	0.66547	0.98043

b. ***Loss Metrics***

During training, a consistent downward trend was shown by the classification loss (cls_loss), bounding box regression loss (box_loss), and distribution focal loss (dfl_loss), indicating the effective minimisation of errors and learning of features from the training data. For the validation data, a decrease and stabilisation were also seen in the classification loss, suggesting good generalisation for classifying faces without significant overfitting. However, some fluctuations and a slight increase in later epochs were shown by the validation box loss and DFL loss (val/dfl_loss). This divergence suggests that a slight overfitting to the training data's specific bounding box details may have begun, making it less precise on unseen validation images.

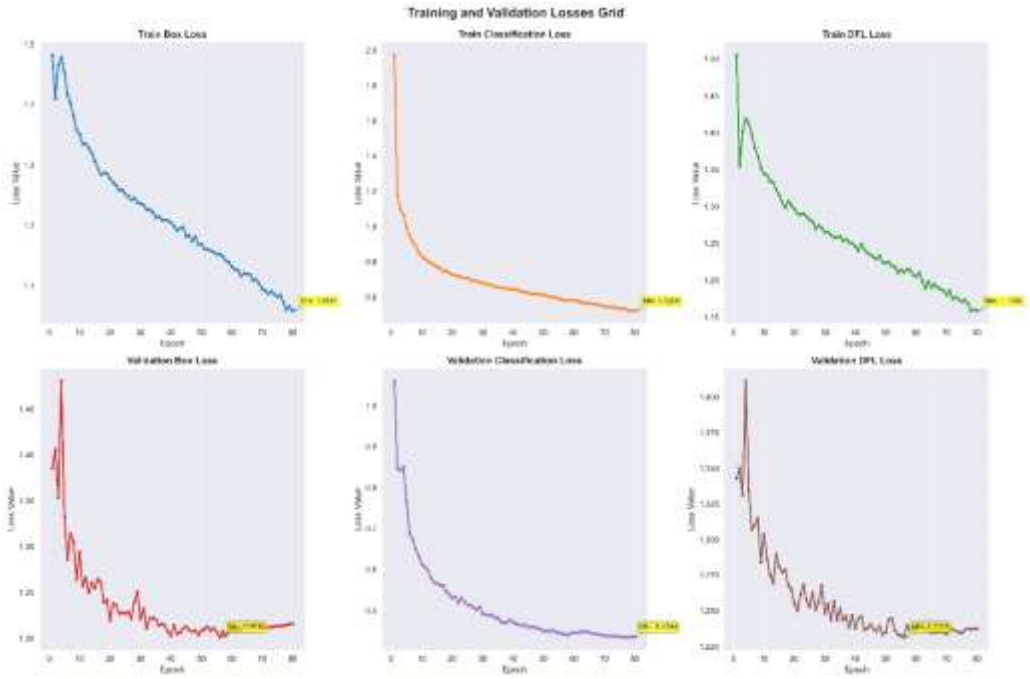

Figure 7. *classification loss* (cls_loss), *bounding box regression loss* (box_loss), and *distribution focal loss* (dfl_loss).

The val/cls_loss decreases quickly and stabilises for validation losses, suggesting good generalisation for classification without significant overfitting. However, the val/box_loss fluctuates throughout training, indicating variability in localisation performance. These fluctuations may arise from challenging validation examples, like objects with varying scales or complex back-grounds, which are less common in the training set. The val/dfl_loss curve initially stabilises but begins to increase slightly in later stages of training, diverging from the steady downward trend of the train/dfl_loss. This divergence suggests the model may be overfitting to the training data for fine-grained bounding box predictions. Overfitting in this context indicates that while the model has learned to localise objects effectively in the training set, it struggles to replicate this precision on the validation set.

c. Confusion Matrix

Figure 7 illustrates that the confusion matrix shows a classification model's performance for child and background classes.

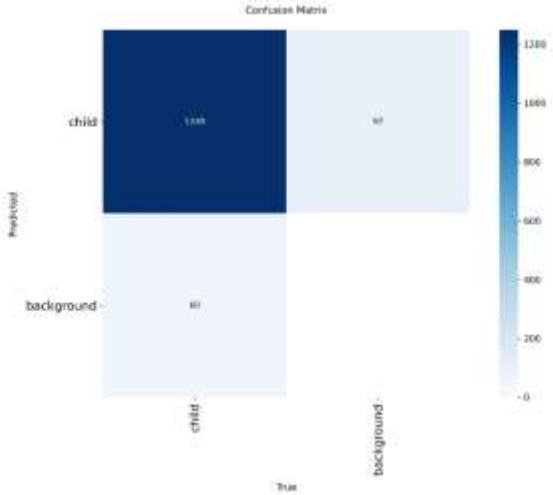

Figure 8. Confusion Matrix

The diagonal entries represent correct predictions, with 1249 instances of the "child" class and 25 instances of the "background" class correctly identified. Off-diagonal entries represent mis-classifications: 97 "child" instances were predicted as "background," and 60 "background" in-stances were predicted as "child." These values suggest that the model performs well overall but exhibits a slightly higher rate of false negatives (misclassifying "child" as "background") than false positives. In the facial de-identification application, false positives may lead to unnecessary blurring of non-child objects, including adult faces, which can degrade image quality and negatively impact the user experience. Conversely, a false negative occurs when the system fails to detect a child's face, leaving it identifiable and potentially exposing the child to risks such as the malicious use of their images. The calculation of precision, recall, F1 Score, and overall ac-curacy are presented in Table 4.

Table 4. Metric Results

Metrik	Formula	Calculation	Final Result
Precision	$\frac{TP}{TP+FP}$	$\frac{1249}{1249+97}$	0.928 (92.8%)
Recall	$\frac{TP}{TP+FN}$	$\frac{1249}{1249+60}$	0.954 (95.4%)
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	$\frac{1249+0}{1406}$	0.88 (88%)
Skor F1	$\frac{2 \times Precision \times Recall}{Precision + Recall}$	$\frac{2 \times 0.928 \times 0.954}{0.928 + 0.954}$	0.941 (94%)

d. Model Testing with Sample Images

This test evaluates the YOLOv8 model's performance on a collection of unseen sample images, focusing on its ability to detect and anonymise the faces of minors. The testing phase utilises images excluded from the training and validation datasets to ensure unbiased performance metrics. Table 5 presents the images used along with the corresponding processed results.

Table 5. Real Image Test

NO	Image	Result
1		
2		

3		
4		
5		
6		
7		

The sample image testing analysis reveals both strengths and limitations of the YOLOv8 model's performance in facial detection and anonymisation. While the model successfully detects and blurs most children's faces across various lighting conditions and angles, handling close-ups and group shots effectively demonstrates some notable challenges. The model occasionally pro-duces false positives by blurring adult faces, particularly evident in the sample images (1), (3), (7), suggesting difficulties in age-based differentiation.

e. Model Implementation in Mobile Application

The integration process begins on the front end, where the user selects an image from the gallery or captures one using a camera. The front end, built with Flutter, sends the image to the back end via an HTTP POST request to a REST API. The back end, implemented with Python and Flask, processes the image using a YOLO model and OpenCV. The YOLO model detects faces and generates bounding boxes, while OpenCV applies a Gaussian blur to anonymise the detected faces. The back end returns the anonymised image to the front end, which displays it to the user. Figure 8 describes the integration process between the frontend and backend.

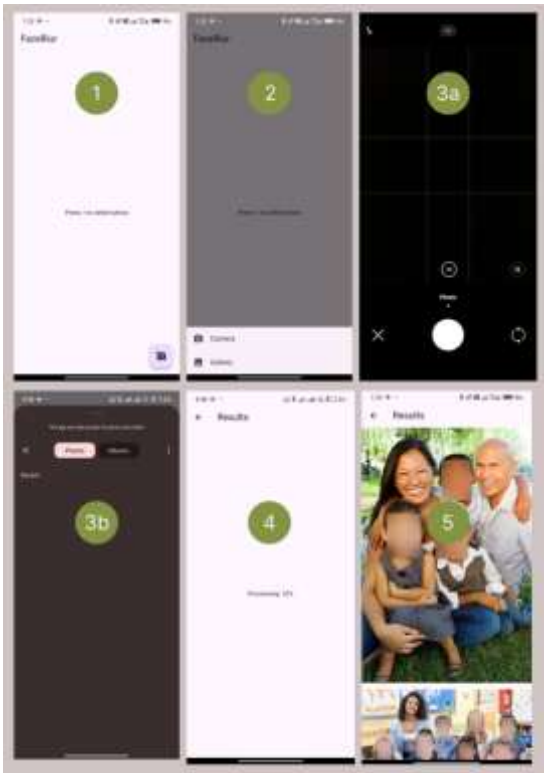

Figure 9. Application Interface

Figure 8 illustrates the user interface. The process is explained step-by-step:

1. Photo Selection Interface (Steps 1 – 3)
   The user initiates the process by selecting a photo through the app. They are given two options: using the camera to take a new photo or choosing existing images from the gallery.
2. Processing Stage (Step 4)
   After selecting an image, the app processes the photograph using the YOLOv8 model to detect minor faces. The application sends the image(s) via the HTTP Post Request in this screen. The progress is displayed to keep the user informed about the status of the operation.
3. Displaying the Result (Step 5)
   The application outputs the anonymised image once processing is complete. Faces identified as minors are blurred.

The primary device used for testing the application was the Xiaomi Redmi Note 13 Pro, with a Helio G99 processor and 8 GB of RAM, running Android 14. The application demonstrated optimal performance on this device, with no observed crashes or issues. Additional testing was conducted on a Samsung Galaxy A30, featuring 4 GB of RAM and an Exynos 7904 Octa-Core processor running Android 11. The application performed consistently well on both devices. An Android 13 emulator was utilised for development purposes, yielding similar performance results. Furthermore, the application was tested on an emulated Android 7.0 (Nougat) environment, where it also operated smoothly. The application's simplicity is attributed to consistent performance across different devices and Android versions.

f. Results Discussion

Previous studies have developed age classification or age range estimation systems using deep learning models but have not integrated them into a system. This study not only

implements the model to a system in the form of an application but also utilises face detection results to anonymise children's faces. Although the developed model shows suboptimal performance on the testing dataset, it still demonstrates the potential for accurately detecting children's faces.

CONCLUSION

This study successfully developed an Android-based application utilising YOLOv8 for facial de-identification to protect minors' privacy. The application demonstrates significant accuracy and precision, with a precision of 92% and an accuracy of 88% for detecting children's faces. It highlights integrating object detection techniques within a user-friendly application to address privacy concerns in today's digital era. However, the model faced challenges in handling unseen data and complex or overlapping facial patterns, indicating the need for a more diverse dataset and robust training strategies; thus, future research could focus on improving model robustness with diverse datasets, exploring additional anonymisation techniques to enhance user experience and data security, and extending the application's ability to work on videos.

REFERENCES

Annur, C. M. (2023). *Jumlah Pengguna Instagram Indonesia Terbanyak ke-4 di Dunia*. Databooks. https://databoks.katadata.co.id/teknologi-telekomunikasi/statistik/f38041b68c2f889/jumlah-pengguna-instagram-indonesia-terbanyak-ke-4-di-dunia

Ardiansyah, A., Qodri, K., Banna, D., & Al-Baihaqi, M. (2024). Pemanfaatan Sam dan Yolov8 Untuk Deteksi dan Segmentation MRI Tumor Otak. *TEKNIMEDIA: Teknologi Informasi Dan Multimedia*, *5*, 82–89. https://doi.org/10.46764/teknimedia.v5i1.192

Chumachenko, K., Gabbouj, M., & Iosifidis, A. (2022). *Chapter 11 - Object detection and tracking* (A. Iosifidis & A. B. T.-D. L. for R. P. and C. Tefas (Eds.); pp. 243–278). Academic Press. https://doi.org/https://doi.org/10.1016/B978-0-32-385787-1.00016-6

Endra, Y., & Ratri, D. D. M. (2021). *Foto Bayi Artis Dicatut dalam Sindikat Jual Beli Anak, Audi Marissa Marah!* Matamata. https://www.matamata.com/seleb/2021/09/05/195718/foto-bayi-artis-dicatut-dalam-sindikat-jual-beli-anak-audi-marissa-marah?ref=detail-terkini-2#.

Fahrezi, M., & Widiyanto, E. (2024). Implementasi YOLOv8 Dalam Penghitung Masuk Dan Keluar Manusia Pada Gedung. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *11*, 335–345.

Gyawali, D., Pokharel, P., Chauhan, A., & Shakya, S. C. (2020). Age Range Estimation Using MTCNN and VGG-Face Model. *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–6. https://doi.org/10.1109/ICCCNT49239.2020.9225443

Hari Dunia Anti Perdagangan Orang 2024, Menteri PPPA: Lawan dan Akhiri Segala Bentuk Perdagangan Orang. (2024). KemenPPPA. https://www.kemenpppa.go.id/page/view/NTMzNQ%3D%3D?

Hari Dunia Anti Perdagangan Orang 2024. (2024). IOM UN Migration Indonesia. https://indonesia.iom.int/id/news/hari-dunia-anti-perdagangan-orang-2024-menciptakan-lingkungan-migrasi-yang-aman-untuk-melawan-perdagangan-orang

Kim, J., Sung, J.-Y., & Park, S. (2020). Comparison of Faster-RCNN, YOLO, and SSD for Real-Time Vehicle Type Recognition. *2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, 1–4. https://doi.org/10.1109/ICCE-Asia49877.2020.9277040

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Livingstone, S., Blum-Ross, A., & Zhang, D. (2024). *What do parents think, and do, about their*

children's online privacy? https://eprints.lse.ac.uk/87954/1/Livingstone_Parenting Digital Survey Report 3_Published.pdf

Maulana, I., Rahaningsih, N., & Suprapti, T. (2024). Analisis Penggunaan Model Yolov8 (You Only Look Once) Terhadap Deteksi Citra Senjata Berbahaya. *Jati*, *7*(6), 3621–3627. https://doi.org/10.36040/jati.v7i6.8271

Mustapha, M. F., Mohamad, N. M., Osman, G., & Ab Hamid, S. H. (2021). Age Group Classification using Convolutional Neural Network (CNN). *Journal of Physics: Conference Series*, *2084*(1), 012028. https://doi.org/10.1088/1742-6596/2084/1/012028

Permana, R., Saldu, H., & Maulana, D. I. (2022). Optimasi Image Classification Pada Jenis Sampah Dengan Data Augmentation dan Convolutional Neural Network. *J.Sist.Inf.Inform.*, *5*(2), 111–120. https://doi.org/10.47080/simika.v5i2.1913

Rahaditya Kusuma, N. T., I Made Agus Dwi Suarjaya, & Vihikan, W. O. (2024). Implementasi Metode YOLOv8 Pada Mobile Apps Untuk Klasifikasi Kain Endek Bali. *Decode*, *4*(3), 787–797. https://doi.org/10.51454/decode.v4i3.610

Rasjid, A. A., Rahmat, B., & Sihananto, A. N. (2024). Implementasi YOLOv8 Pada Robot Deteksi Objek. *Journal of Technology and System Information*, *1*(3), 9. https://doi.org/10.47134/jtsi.v1i3.2969

Setiyadi, A., Utami, E., & Ariatmanto, D. (2023). Analisa Kemampuan Algoritma YOLOv8 Dalam Deteksi Objek Manusia Dengan Metode Modifikasi Arsitektur. *J-SAKTI (Jurnal Sains Komputer & Informatika)*, *7*(2). https://tunasbangsa.ac.id/ejurnal/index.php/jsakti/article/view/694

Sharma, N. (2023). *Understanding and Applying F1 Score: AI Evaluation Essentials with Hands-On Coding Example*. Arize. https://arize.com/blog-course/f1-score/

Solawetz, J. (2020). *What is Mean Average Precision (mAP) in Object Detection?* Roboflow. https://blog.roboflow.com/mean-average-precision/

Srivastava, S., Divekar, A. V., Anilkumar, C., Naik, I., Kulkarni, V., & Pattabiraman, V. (2021). Comparative analysis of deep learning image detection algorithms. *Journal of Big Data*, *8*(1), 66. https://doi.org/10.1186/s40537-021-00434-w

Takase, Y. (2023). *Introducing YoloV8: Operation and comparison to previous versions of the object detection model*. RevComm Tech Blog. https://tech.revcomm.co.jp/yolov8-introduction-en

Terven, J. R., & Esparza, D. M. C. (2023). A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond. *ArXiv*, *abs/2304.00501*. https://api.semanticscholar.org/CorpusID:257913644

Thaneeshan, R., Thanikasalam, K., & Pinidiyaarachchi, A. (2022). Gender and Age Estimation From Facial Images using Deep Learning. *2022 7th International Conference on Information Technology Research (ICITR)*, 1–6. https://doi.org/10.1109/ICITR57877.2022.9993277

Ting, K. M. (2010). Confusion Matrix. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (p. 209). Springer US. https://doi.org/10.1007/978-0-387-30164-8_157

United Nation Office on Drugs and Crime. (2024). United Nations. https://www.unodc.org/unodc/en/press/releases/2024/December/unodc-global-human-trafficking-report_-detected-victims-up-25-per-cent-as-more-children-are-exploited-and-forced-labour-cases-spike.html

Wisak, S. A., Safirah, N. A., & Kaesmetan, Y. R. (2024). Identifikasi Jenis Mangga Berdasarkan Ciri Daun Menggunakan Metode CNN. *J.Sist.Inf.Inform.*, *7*(2), 121–129. https://doi.org/10.47080/simika.v7i2.3295

Zhang, Y.-J. (2023). Computer Vision Overview. In Y.-J. Zhang (Ed.), *3-D Computer Vision: Principles, Algorithms and Applications* (pp. 1–35). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-7580-6_1