Motorcycle License Plate and Driver Face Verification Using Siamese Neural Network Model

[1*]Yeremia Yosefan Pane, [2]Kana Saputra S,
[3]Said Iskandar Al Idrus, [4]Hermawan Syahputra
[1,2,3,4]Department of Computer Science, Faculty of Mathematics and Natural Science,
Universitas Negeri Medan, Indonesia
Email: [1]yeremiayosefan@mhs.unimed.ac.id, [2]kanasaputra@unimed.ac.id,
[3]saidiskandar@unimed.ac.id, [4]hsyahputra@unimed.ac.id

Article Info
Article history:
Received Jul 22th, 2024
Revised Dec 13th, 2024
Accepted Feb 12th, 2025

Keyword:
Face Recognition
Motorcycle License Plate
Siamese Neural Network
Traffic
Vehicle Plate Detection

ABSTRACT
The security and efficiency of vehicle access management systems have become a primary concern for various institutions, including universities, offices, and public facilities. Effective access management not only enhances security but also improves the flow of incoming and outgoing vehicles, reduces congestion, and enhances user experience. This research aims to develop a vehicle plate detection system and driver face recognition using the Siamese Neural Network model to optimize traffic at the gate. The methods used include the application of deep learning algorithms, specifically the Siamese Neural Network, to verify the driver's face and the use of You Only Live Once (YOLO) to detect and recognize vehicle plates in real-time. Data was collected through direct capture with the researcher's camera. The model was trained and tested using a dataset containing images of vehicle license plates and driver faces. The results showed that the developed model was able to detect and recognize the vehicle plate and the driver's face with a fairly high accuracy, namely in the object detection results getting bounding box validation is 1.05 and class loss validation is 0.95, and 0.85 mAP. As well as in training using the Siamese Neural Network, the highest result is 0.82 with a learning rate of 10e-5 with 30 epochs. It is hoped that this system can be one of the innovations that can be applied in government agencies, universities, industries, etc.
Copyright © 2025 Puzzle Research Data Technology

Corresponding Author:
Yeremia Yosefan Pane,
Departement of Mathematics and Natural Sciences,
Universitas Negeri Medan,
Jl. Willem Iskandar / Pasar V, Medan, Sumatera Utara – Indonesia.
Email: yeremiayosefan@mhs.unimed.ac.id
DOI: http://dx.doi.org/10.24014/ijaidm.v8i1.31750

1. INTRODUCTION

In the era of advancing digitalization, the security and efficiency of vehicle access management systems have become a primary concern for various institutions, including universities, offices, and public facilities. Effective access management not only enhances security but also improves the flow of incoming and outgoing vehicles, reduces congestion, and enhances user experience.Traditional systems relying on manual inspection of vehicle documents and visual identification of drivers by security personnel are often inefficient and prone to human error. The time required to manually verify license plates and driver faces leads to long queues, especially during peak hours. Additionally, varying levels of inspection accuracy can create security vulnerabilities. Vehicle license plate recognition technology (Automatic License Plate Recognition or ALPR) and facial verification (Face Recognition) have advanced significantly, offering faster and more accurate solutions. However, the challenge in implementing these technologies lies in integrating them into a single system capable of real-time operation with high accuracy.

Nowadays technology has developed a lot through the rapid advancement of computer technology today. Computers can be used to help humans to facilitate work quickly and effectively. One of them is object recognition and pattern recognition. With the development of artificial intelligence, currently many industries have used the help of this technology for various purposes, one of which is security [1]. The security offered by this technology certainly varies from object recognition or detection using cameras to face recognition. Of course, by using this technology, various industries can use it to get maximum benefits [2].

Previously, research related to vehicle security systems has been carried out by adopting machine learning technology, one of which is research conducted by Rifki, et al who conducted research on the recognition of Number Plates and Drivers' Faces Using Convolutional Neural Network and Absolute Difference Method in Automatic Gate Systems, where face recognition is carried out using the haar-cascade method while the image extraction method uses Histogram of Oriented Gradient (HOG), for the face recognition process using a face recognition library [3]. While the license plate recognition uses the convolutional neural network method. Face testing gets 100% accuracy when the face is without obstructions. For the detection and reading of license plates, the accuracy is 97.1% for accuracy and 94% for plate reading, the method used is good enough in recognizing faces and vehicle plates. Previous research has also been conducted by Farhan Aditama, et al by making a face recognition and verification system using raspberry pi-based transfer learning with a facenet model that uses the Res-Net V1 architecture, where the accuracy results obtained during the training process are 98%, while in realtime results when already on the system obtained 95% results. These results are good enough to be used as a tool for face verification [4].

Siamese Neural Network is one of the developments of a neural network algorithm that compares two input data patterns and the output value is the similarity value between the two patterns [5]. The architecture of this algorithm is designed to adapt to small amounts of data and can overcome problems that often occur in images, such as differences in light, pose, expression, etc [6]. This network can be used to recognize faces by comparing facial features extracted from two input images. So that this algorithm is very suitable for use in the case study that the author brings up.

Therefore, a solution is needed in the form of a machine learning application that can identify vehicle plates and facial recognition to improve the security system and efficiency of vehicle access management systems in the various institutions, including universities, offices, and public facilities environment by applying Artificial Inteligent and OCR

2. RESEARCH METHOD

Research scheme of work, also known as research design or research methodology design, refers to the overall design of a study. It includes the systematic steps and methods that the researcher will follow to answer the research question or achieve the research objectives. Figure 1 is a research work scheme for vehicle license plate and driver face detection using Siamese Neural Network architecture. The scheme explains the stages of taking license plate data and facial images, preprocessing, to the implementation of the Siamese Neural Network model and evaluation of results and analysis. In general, this scheme visualizes the research method step by step so that the research can run in a structured manner until the achievement of the final goal, namely the design of an automatic detection system for the identity of the vehicle and its driver.

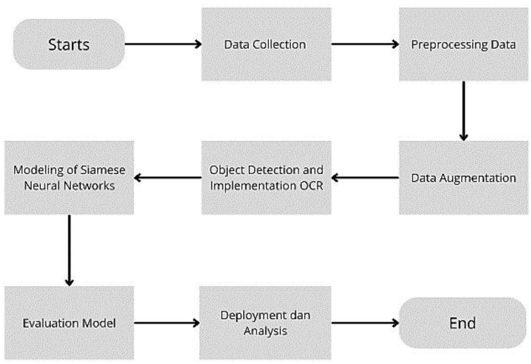

Figure 1. Research flow chart

2.1. Data Collection

Data collection is carried out to obtain data in the form of image data that will be taken for the training process and the data validation process using the machine learning model when the vehicle exits through the main gate. The data collection process was carried out using the researcher's smartphone camera by capturing images of the research object. After data collection, the data will be stored in a directory that has been determined by the researcher. The amount of data obtained is 560 image data, and will be divided into training data and validation data 80:20 respectively. Where training data is 80% of the overall data and 20% for validation data.

2.2. Preprocessing Data

After collecting the data, the researchers will preprocess the data first to minimize the occurrence of image errors. At this stage, researchers will ensure that the images are in accordance with the folder that has been determined by the researcher. This is also often referred to as labeling or the process of labeling each data. Labels are useful for determining the class that will be generated by the model for classification cases. At this stage, the image data that has been collected will be processed to set the image size, so that each data has the same size. In pre-processing the image data for SNNs modelling, image cropping is performed first. Image cropping is performed on the rider's head area as shown in Figure 2. This is done to perform face verification modelling. So it is very important to do before modelling.

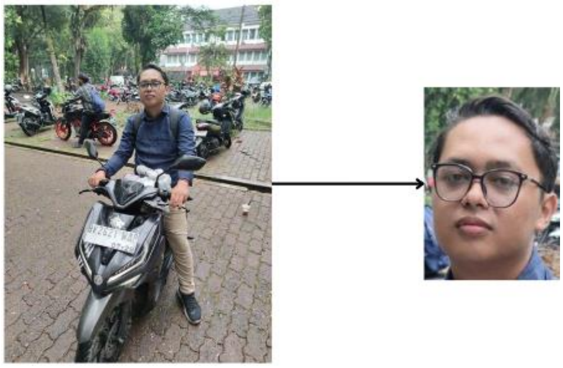

Figure 2. Croping Face Object from Images

Data that has been adjusted based on its class is then labelled. This is done for training and validation of modelling on object detection. In doing data labelling, researchers use the labelimg library that is available in the python programming language as shown in figure 3. Data labelling in labelimg is done by creating bounding-boxes on objects that will be modelled using YOLOv8. In data labelling there will be 2 classes that will be detected, namely faces and vehicle number plates. After data labelling is done using labeling, the bounding box coordinates will be saved in the form of a .txt file extension.

Figure 3. Labeling Data Using labelimg

2.3. Data Augmentation

Data augmentation is an important technique in image data modelling to improve the performance of machine learning models, especially in classification and object detection tasks. It involves modifying existing image data to generate new variations, so that the model is exposed to more data variations and learns to be more generalisable. Modifications can include normalising the pixel values in the image, randomly rotating the image with a maximum angle of 90 degrees clockwise, randomly shifting the image up, down, left, or right, adjusting the lighting, tilting the image, and etc.

2.4. Object Detection and Impelementation OCR
2.4.1. Object Detection

Object detection is an ability possessed by a computer to be able to identify an object or object contained in a video that has been converted into an image or in an image. Object detection is one of the techniques used in computer vision to identify objects. One method that can be implemented to detect an object is You Only Look Once (YOLO) object detection which provides a pre-trained model and easy to use.

YOLO is an object detection algorithm used to identify and classify objects in a single process on images using a covolutional neural network algorithm as the basis for feature extraction. First introduced by Joseph Redmon and his team in 2016, YOLO uses a single neural network to predict bounding boxes and class probabilities directly in one evaluation stage. This approach allows YOLO to provide more accurate results in less time. Since the first version, YOLO has undergone several significant updates that have brought improvements in performance, accuracy, and speed [7].

YOLOv8 is the latest model of the YOLO version that has been released and developed by ultralytics. YOLOv8 consists of three main components backbone, neck and head [7]. Where this main component makes this model has a fairly good accuracy and speed when compared to other models [8]. In its application, YOLO can perform object detection, classification, object segmentation, etc. Where the backbone part serves to extract features, the neck part serves to connect the features extracted by the Backbone. And the head is responsible for predicting objects [9].

2.4.2. Implementation OCR

After object detection, text extraction from the vehicle license plate is then carried out to obtain information about the vehicle. One way to be able to perform text extraction automatically is to use Optical Character Recognition (OCR). Optical Character Recognition or often known as OCR is an application that can read characters contained in an image. In this research, the author uses the easyocr library in implementing OCR. EasyOCR is a Python library that allows developers to do Optical Character Recognition (OCR) easily. OCR is a technology that can recognize and convert text in scanned images or documents into digital text. EasyOCR has supported 80 available languages including Indonesian, so this can be applied in character extraction on vehicle license plates.

2.5. Siamese Neural Network Model

Siamese Neural Network is a neural network that consists of two or more identical sub-networks where both inputs will share the same weights and the same architecture. Each subnetwork processes one of the multiple inputs and outputs a representation (embedding) of that input into a high-dimensional vector space. The main goal of this network is to maximize the similarity between embeddings of similar input pairs and minimize the similarity between embeddings of different input pairs. Siamese Neural Networks are commonly used to perform tasks such as face recognition, image verification, and image matching. In its application to triplet loss, anchor, positive and negative inputs are needed to train the Siamese Neural Network model to measure the similarity value between anchor and positive and anchor and negative [10].

Anchor data is data that will be predicted for similarity, positive data is data that is the original face on each label, while negative data is an image that does not come from positive data, but data that is not an image in the class. In Figure 4 below, we can see one example of connecting anchor, positive and negative data. After connecting the anchor, positive and negative data, modeling can be done using the Siamese Neural Network [11]. Modeling with Siamese Neural Network is done using InceptionResNetV2 architecture which is an extension of the Inception architecture by adding residual linking to improve model performance and accuracy In using the Siamese Neural Network, the input images will be given the same weights and then the embeddings will be calculated so that the similarity distance can be calculated. In calculating the distance, the author uses the calculation of euclidean distances to measure the similarity of the three data. The following can be described for the application of euclidean distances using equation (1).

$$D(A,B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2} \tag{1}$$

Where D is the euclidean distances between input vector A and vector B. and embedding vector A_i and vector B_i are the i-th components of vectors A and B respectively. To train a Siamese network, the data must be given similarity labels such as "same" or "different" for each input pair. The model then learns to generate high similarity predictions for input pairs labelled "same" and low for "different". The forward process consists of parallel feature extraction and calculation of the similarity loss function [12]. Ilustration of Siamese Neural Network Using Triplet Loss can view figure 4.

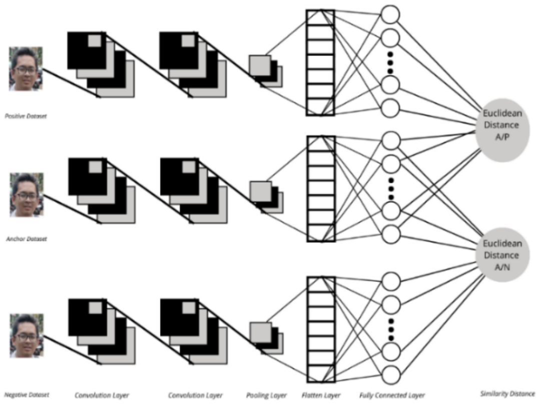

Figure 4. Ilustration of Siamese Neural Network Using Triplet Loss

2.6. Model Evaluation

Machine learning model evaluation plays an important role in assessing the performance of a model that has been built. This process aims to ensure that the model successfully achieves its goals, which can involve tasks such as classification, regression, or clustering [13]. In the context of this evaluation, metrics become the main tool to measure the extent to which the model successfully performs its function. One commonly used metric is accuracy. Accuracy measures how good the model is at making correct predictions compared to the total predictions made. The accuracy metric can be calculated using equation (2).

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{2}$$

Accuracy metrics provide an easy-to-understand overview of how well a machine learning model can correctly predict data. Its high readability makes accuracy a frequently used metric in model performance evaluation, especially in the case of binary classification [14]. This advantage makes accuracy a good choice when a quick and intuitive interpretation is required.

In Siamese Neural Network, the loss function has several types that can be used when evaluating, such as triplet loss, uber loss, constrative loss, etc. The author chooses to take triplet loss as the loss function, as it significantly improves the accuracy of the face recognition system by learning a more representative and structured embedding. In its various applications, triplet loss is often used to verify signatures [15], object similarity [16], [17], to the implementation in natural language processing[18].

Triplet Loss is a function used to learn embedding in Euclidean space. The main purpose of triplet loss is to ensure that the distance between pairs of similar data or positive data is closer than the distance between pairs of dissimilar data or so-called negative data. Triplet loss consists of three main components: anchor which is a reference example, positive which is an example similar to the anchor, and negative which is an example different from the anchor. This loss function works by minimizing the distance between anchor and positive, while maximizing the distance between anchor and negative. Mathematically, triplet loss can be formulated 3.

$$\mathrm{L} = \max\left(0, \mathrm{d}(\mathrm{a}, \mathrm{p}) - \mathrm{d}(\mathrm{a}, \mathrm{n}) + \alpha\right) \tag{3}$$

Where the L value is the triplet loss value obtained from the maximum value between data distances. Next, the value of $\mathrm{d}(\mathrm{a}, \mathrm{p})$ is the distance value between the input anchor vector and positive or in the sense of a value with the same class label. Furthermore, $\mathrm{d}(\mathrm{a}, \mathrm{n})$ is the distance value between the input vector anchor and negative. And then the margin value (α) is added to be able to overcome or enlarge the distance between the anchor and negative values.

2.7. Deployment and Analysis

After the model evaluation process has been carried out and the model has reached the optimal accuracy value, the model that has been built will be saved in the form of a file with the h5 extension. This is useful so that the model can be used anytime when needed, so there is no need to train the model again. Furthermore, the model will be deployed to the system in the form of a website application. Furthermore, the results obtained from the web application will be analyzed in the form of model performance and testing.

3. RESULTS AND ANALYSIS

This section will discuss the results obtained based on the research flow that has been carried out by the author using YOLO to perform object detection on vehicle number plates and driver faces and Siamese Neural Networks to verify driver faces using the InceptionResNetV2 model. After that, the author will present the results of the model evaluation that has been obtained based on hyperparameter tuning. In addition, the author will also provide model performance results from tests that will be carried out when checking manually and using the system.

3.1. Model Training and Evaluation

3.1.1. Object Detection Training and Evaluation

In this research, the author uses YOLOv8l to train the model by using Google Colab as an assisting tool in training the model. In google colab, the author uses an additional GPU, specifically L4GPU to speed up the training process which is available in the pro version of google colab. In this case, the author provides hyperparameter functions on several parameters such as learning rate, epochs, batch size, and drop out. It can be seen in the table 1. the parameters that will be used by the author in conducting training.

We selected these hyperparameter values based on a combination of prior studies and empirical tuning. The YOLOv8l variant was chosen for its higher accuracy potential compared to smaller YOLOv8 models, leveraging the available GPU computation at the cost of more parameters [19]. The input image resolution was set to 640×640 pixels, which is a standard size in YOLO implementations that provides a good balance between detection accuracy and speed. Using this resolution allows the model to capture sufficient detail of faces and plates while keeping inference time reasonable. We set the number of training epochs to 300 to allow the model ample opportunity to learn the features in our dataset; given the moderate size of 560 images (augmented to many more samples), 300 epochs helped the model converge without significant overfitting (as validated by the loss curves). A batch size of 20 was used to maximize GPU memory utilization while avoiding memory overflow – this batch size was a trade-off between gradient stability and hardware limits. We initialized the learning rate at 1×10^{-6}, a relatively low value, to ensure stable training in the beginning. A small learning rate prevents large weight updates that could destabilize training, especially important when fine-tuning a complex model like YOLOv8 on a new dataset[20]. This learning rate could be increased later if needed, but our experiments found that starting low and gradually increasing (if using a scheduler) yielded better results than starting too high. We also employed dropout as a regularization technique during training (e.g., in the classification head of YOLO or in augmented layers if applicable). Dropout randomly disables a fraction of neurons during each training iteration, which helps prevent the model from becoming too specialized to the training data. By incorporating dropout, we aimed to improve the model's ability to generalize to new images, given our relatively limited dataset size. The Hyperparameter Training Model can view table 1.

Table 1. Hyperparameter Training Model

Hyperparameter	Value
Epochs (epoch)	300
Image Size (imgsz)	(640,640)
Batch Size (batch)	20
Learning Rate (lr)	10^{-6}

Subsequent to establishing the aforementioned hyperparameters, the YOLOv8 model was trained on the designated training set, with its performance on the validation set being meticulously monitored. The training process yielded multiple metrics for each epoch.The training and validation loss curves, along with the mAP (mean average precision) curve, are depicted in Figure 5.A thorough examination of Figure 5 reveals that the model converges remarkably well, as evidenced by the consistent decline in both the bounding box regression loss and the classification loss on the validation set during the course of training. The optimal result obtained for the bounding box loss (validation) was 1.05, and for the classification loss (validation), it was 0.95. Lower values indicate superior performance in both localization and classification. As demonstrated in the graph, the model's performance improves with each epoch, as evidenced by the downward trend of the loss curves. This indicates that the model is successfully learning to detect faces and plates with greater accuracy over time. The mAP value on the validation set also exhibited an upward trend, reaching approximately 0.85 (85%) at IoU 0.5 by the conclusion of the training process. mAP 0.5 is a prevalent metric for object detection, quantifying the mean detection precision when the Intersection over Union threshold for a precise detection is 50%. An mAP of 0.85 signifies that the model is accurately detecting and classifying a substantial proportion of faces and plates. This high mAP, in conjunction with the observed decrease in validation losses, serves as a testament to the efficacy of the training approach employed. Notably, the augmented data enabled the model to continue improving and avoid premature stagnation, validating the chosen methodology. The results underscore the importance of a sufficient training epoch, a judiciously tuned learning rate, and data augmentation in facilitating the YOLOv8 model's achievement of commendable performance in detecting the two target object classes. The model exhibited no indications of significant overfitting, as evidenced by the continual decrease in validation loss. This finding suggests that the augmentation and dropout regularization techniques employed were effective. The figure displays the training/validation loss curves for the bounding box and class predictions, and the mAP curve across 300 epochs. The decreasing validation losses and increasing mAP indicate improving model accuracy on the vehicle plate and face detection task as training progresses. Evaluation Model YOLOv8L can view detail in figure 5.

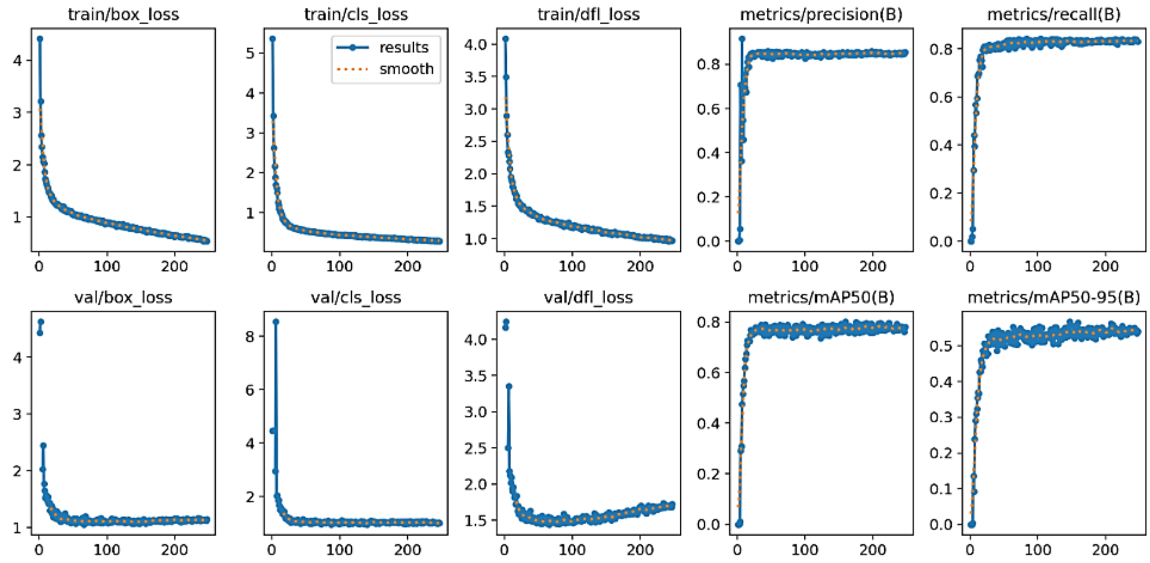

Figure 5. Evaluation Model YOLOv8L

3.1.2. Siamese Neural Network Training and Evaluation

The model training uses the InceptionResNetV2 architecture, which is an extension of the Inception model. The author uses several hyperparameter functions such as epochs, learning rate, adding layers (Dense layer, Dropout layer, BatchNormalization layer), and Callbacks to retrieve the best value of the model based on the training accuracy and training validation obtained. In this study, the author will change the learning rate to determine the difference between the performance of the model. And the author will use 30 epochs in training. The following data presents the results of the model performance evaluation in table 2.

Table 2. Training model evaluation comparison

Learning Rate	Epochs	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Precission	Recall
$10e^{-6}$	30	0.932	0.115	0.824	0.301	0.735	0.72
$10e^{-5}$	30	0.863	0.258	0.775	0.392	0.80	0.82
$10e^{-4}$	30	0.998	0.002	0.882	0.329	0.76	0.76

Table 2 is a comparison of the model evaluation results on hyperparameters with learning rates of $10e^{-6}$, $10e^{-5}$, $10e^{-4}$. From the model evaluation with three different learning rates, learning rate $10e^{-6}$ yields accuracy 0.74, precision 0.735, and recall 0.72, showing stable but slow and suboptimal training. Learning rate $10e^{-5}$ gives the best performance with accuracy 0.82, precision 0.80, and recall 0.82, showing a good balance between convergence speed and stability. Meanwhile, learning rate 10^{e-4} yielded accuracy 0.77, precision 0.76, and recall 0.763, indicating instability and degraded performance. Therefore, a learning rate of 10^{e-5} is the most optimal for this model. Figure 6 is a graph for model training using 3 comparisons of model performance by adjusting the learning rate paramater. The plot compares training/validation accuracy and loss for learning rates $10e^{-6}$, $10e^{-5}$, and $10e^{-4}$. The model with learning rate $10e^{-5}$ achieves the best validation accuracy and a good convergence profile, whereas $10e^{-6}$ learns slower and $10e^{-4}$ shows signs of overfitting despite rapid initial learning. Model Performance Plot can view in figure 6.

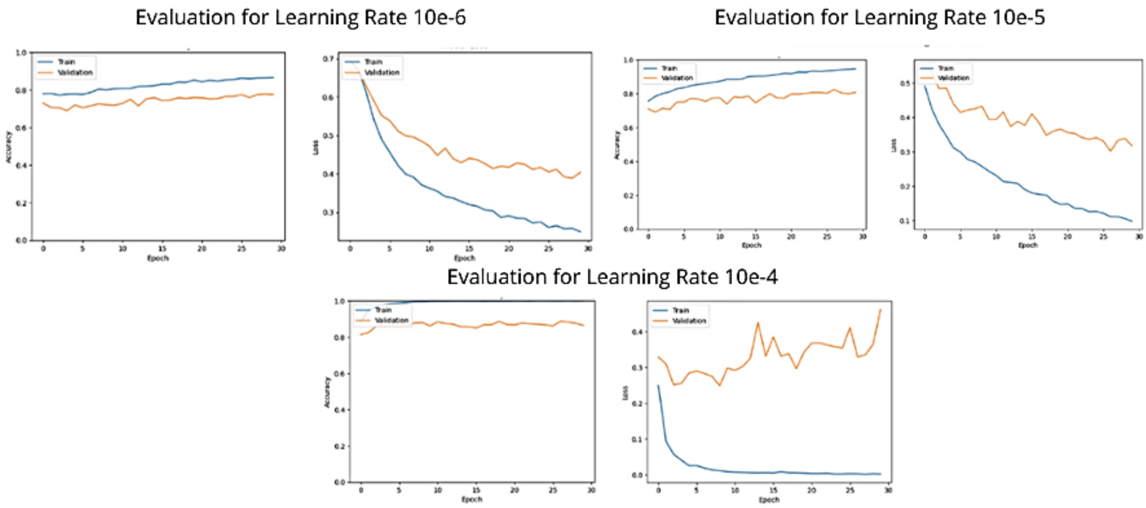

Figure 6. Model Performance Plot

In summary, through hyperparameter tuning we found that training the Siamese network for 30 epochs with a learning rate of $10e^{-5}$ and using the InceptionResNetV2 architecture (with added regularization layers) produces the best result for face verification. The final trained Siamese model (with the 1e-5 learning rate configuration) achieves approximately 77–82% accuracy in verifying whether two face images belong to the same person, with precision 80% and recall 82% on the validation set. This level of performance demonstrates that our methodology for face verification is effective. The clear link between our training methodology and the results can be seen: by choosing an appropriate network architecture and tuning the learning rate, we improved the model’s ability to generalize (as evidenced by balanced precision/recall). The use of triplet loss and data preprocessing (face cropping, etc.) also contributed to learning a good facial representation. The Siamese model is therefore successful in accomplishing the second objective of our system to verifying the driver’s identity by face.

In previous research conducted by Rifki et al who tested the recognition of number plates and faces of motorists using the convolutional neural network algorithm and the abolute difference method, the maximum results were 97% and 94% [3] and research conducted by Aditama and Haryanti in face recognition and verification systems using transfer learning, where the accuracy results obtained were 98% on training and 95% on validation[4]. Where the results of previous research when compared to the results obtained in this research are quite good, considering that this is experimental research. But it does require further research to get maximum results.

3.2. Deployment and Analysis

The results that have been obtained through the model that has been built and stored, will then be loaded again for deployment on the web using Flask. Flask is one of the micro web frameworks provided by Python to be able to build a web. In this case, the author performs deployment by building an API using Flask. In building the interface on the web, the author uses the Bootstrap framework. Bootstrap framework is one of the frameworks to build a website display with the help of javascript as an API function. In Figure 7 is a display that has been built by the author to do the deployment model. On the left display is used to select the menu that will be predicted, namely through uploading images or videos and can select the webcam button for realtime processing. And on the right display is a display for the results of the model. It can be noted when uploading an image and prediction is done, the model can recognize or show a fairly good performance with the distance value between the prediction face and the actual face with a distance value of 0.695. Where if the distance value between the anchor and positive is close to zero, the results are getting better and it can be ascertained that the image is correct. Web Interface for Prediction can view at figure 7.

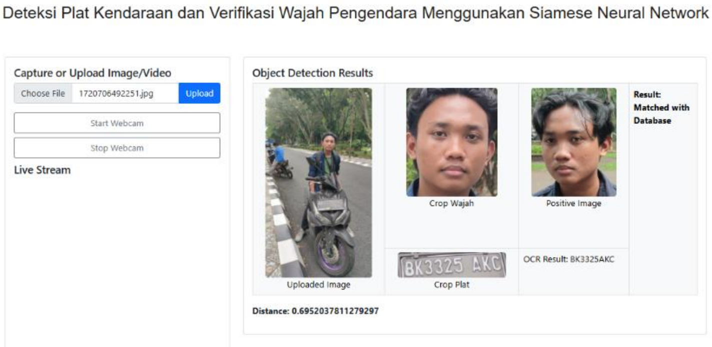

Figure 7. Web Interface for Prediction

4. CONCLUSION

The objective of this research is to develop an integrated system for number plate detection and face verification, with a view to enhancing the security and efficiency of vehicle access management. The evaluation results demonstrated that the YOLOv8 model successfully achieved a mean average precision (mAP) of 0.85 for object detection, while the Siamese Neural Network produced a verification accuracy of approximately 77%, precision 80% and recall 82%. A re-examination of the original objectives indicates that the system has met the targets by reducing verification time and improving identification accuracy, thereby confirming the relevance and appropriateness of the research objectives. However, further development is required to optimise the parameters and assess the system's performance in more diverse operational scenarios.

REFERENCES

[1] R. Nandita Al Zahra, I. Maulana, A. Dwiantoro, and N. Putri Damayanti, "Artificial Intelligence (AI) Imaging untuk Peningkatan Keamanan Tempat Parkir," Semarang, Oct. 2021.
[2] M. Farwati, I. Talitha Salsabila, K. Raihanun Navira, and T. Sutabri, "Analisa Pengaruh Teknologi Artificial Intelligence (Ai) Dalam Kehidupan Sehari-Hari," Jurnal Sistem Informasi & Manajemen, vol. 11, no. 01, pp. 39–46, 2023.
[3] K. Rifki, J. Priambodo, and A. Musthofa, "Pengenalan Plat Nomor dan Wajah Pengendara Menggunakan Convolutional Neural Network dan Metode Absolute Difference pada Sistem Gerbang Otomatis," Jurnal TEKNIK ITS, vol. 10, no. 2, pp. 386–392, 2021.
[4] M. Farhan Aditama and M. S. Haryanti, "Sistem Pengenalan dan Verifikasi Wajah Menggunakan Transfer Learning Berbasis Raspberry PI," Jurnal Teknologi Industri, vol. 12, no. 1, 2023.
[5] M. Toby Suwindra, A. Erlansari, and J. W. Supratman Kandang Limun, "Analisis Kemiripan Jenis Burung Menggunakan Siamese Neural Network Analysis Of Bird Species Similarity Using Siamese Neural Network," 2021. [Online]. Available: http://ejournal.unib.ac.id/index.php/rekursif/193
[6] W. Hayale, P. S. Negi, and M. H. Mahoor, "Deep Siamese Neural Networks for Facial Expression Recognition in the Wild," IEEE Trans Affect Comput, vol. 14, no. 2, pp. 1148–1158, Apr. 2023, doi: 10.1109/TAFFC.2021.3077248.
[7] A. Setiyadi, E. Utami, and D. Ariatmanto, "Analisa Kemampuan Algoritma YOLOv8 Dalam Deteksi Objek Manusia Dengan Metode Modifikasi Arsitektur," Jurnal Sains Komputer & Informatika (J-SAKTI, vol. 7, no. 2, pp. 891–901, 2023.
[8] Q. Aini, N. Lutfiani, H. Kusumah, and M. S. Zahran, "Deteksi Dan Pengenalan Objek Dengan Model Machine Learning: Model YOLO," CESS (Journal of Computer Engineering System and Science), vol. 6, no. 2, pp. 2502–714, 2021.
[9] D. A. Abdurrafi, M. Taqijjuddin Alawiy, and B. M. Basuki, "Deteksi Klasifikasi Dan Menghitung Kendaraan Berbasis Algoritma You Only Look Once (Yolo) Menggunakan Kamera CCTV," SCIENCE ELECTRO, vol. nn, no. 9, 2023.
[10] Y. Zhang et al., "Similarity-based pairing improves efficiency of Siamese Neural Networks for regression tasks and uncertainty quantification," J Cheminform, vol. 15, no. 1, Dec. 2023, doi: 10.1186/s13321-023-00744-6.
[11] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," Feb. 2016, [Online]. Available: http://arxiv.org/abs/1602.07261
[12] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-shot Image Recognition," Toronto, 2015. Accessed: Dec. 01, 2023. [Online]. Available: http://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf
[13] Rifkie Primartha, Algoritma Machine Learning, 1st ed., vol. 1. Bandung: Informatika Bandung, 2021.
[14] N. Dewi and F. Ismawan, "Implementasi Deep Learning Menggunakan Cnn Untuk Sistem Pengenalan Wajah," Faktor Exacta, vol. 14, no. 1, p. 34, Mar. 2021, doi: 10.30998/faktorexacta.v14i1.8989.
[15] M. F. Mayda and A. Musdholifah, "Siamese-Network Based Signature Verification using Self Supervised Learning," IJCCS (Indonesian Journal of Computing and Cybernetics Systems), vol. 17, no. 2, p. 115, Apr. 2023, doi: 10.22146/ijccs.74627.

[16] Y. Pratama, S. T. N. Nainggolan, D. I. Nadya, and N. Y. Naipospos, “One-shot learning Batak Toba character recognition using Siamese Neural Network,” Telkomnika (Telecommunication Computing Electronics and Control), vol. 21, no. 3, pp. 600–612, Jun. 2023, doi: 10.12928/TELKOMNIKA.v21i3.24927.
[17] M. Toby Suwindra, A. Erlansari, and J. W. Supratman Kandang Limun, “Analisis Kemiripan Jenis Burung Menggunakan Siamese Neural Network Analysis Of Bird Species Similarity Using Siamese Neural Network,” 2021. [Online]. Available: http://ejournal.unib.ac.id/index.php/rekursif/193
[18] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” ArXiv, vol. 35, no. 11, Aug. 2019, [Online]. Available: http://arxiv.org/abs/1908.10084
[19] L. Ramos, E. Casas, E. Bendek, C. Romero, and F. Rivas-Echeverría, “Hyperparameter optimization of YOLOv8 for smoke and wildfire detection: Implications for agricultural and environmental safety,” Artificial Intelligence in Agriculture, vol. 12, pp. 109–126, Jun. 2024, doi: 10.1016/j.aiia.2024.05.003.
[20] E. Panja, H. Hendry, and C. Dewi, “YOLOv8 Analysis for Vehicle Classification Under Various Image Conditions,” Scientific Journal of Informatics, vol. 11, no. 1, pp. 127–138, Feb. 2024, doi: 10.15294/sji.v11i1.49038.

BIBLIOGRAPHY OF AUTHORS

Yeremia Yosefan Pane is currently pursuing a Bachelor of Computer Science degree at Medan State University, where he has been enrolled since 2020. He is active in research related to computer science, and focuses his research on data mining, computer vision, and deep learning. His dedication to these fields is shown by his commitment to keep abreast of the latest developments and techniques in computer science.

Kana Saputra S, S.Pd., M.Kom., is an lecturer and researcher who specialises in the fields of artificial intelligence, machine learning and bioinformatics. He is currently employed as a lecturer at the University of North Sumatra, where she has been conducting research in these three fields since 2018. To date, her research has been cited 194 times on Google Scholar.

Iskandar Al Idrus, S.Si., M.Si., is a lecturer and researcher at the University of North Sumatra, specialising in the fields of artificial intelligence, computer engineering and education. He has contributed to both national and international research through his research in these areas.

Dr. Hermawan Syahputra, S.Si., M.Si., is a lecturer and researcher at Medan State University. His research focuses on artificial intelligence, computer vision, pattern recognition and image processing. He has contributed to national and international research projects, thanks to his research in these areas, which has resulted in a significant body of knowledge. As of 2018, his research has been cited 335 times on Google Scholar.