

THE DETECTION OF FACE RECOGNITION AS EMPLOYEE ATTENDANCE PRESENCE USING THE YOLO ALGORITHM (YOU ONLY LOOK ONCE)

Oleh:

Diana Puspita Sari ¹⁾

A. Haidar Mirza ²⁾

Universitas Bina Darma, Medan ^{1,2)}

E-mail:

dianapuspita719@gmail.com ¹⁾

haidar.mirza06@gmail.com ²⁾

ABSTRACT

face detection is a fundamental and important process in the field of facial recognition. The aim of this face detection was to determine the presence and mark the position of the face through an image called a bounding box. The problem that we often encounter in attendance machines is that it is often difficult and takes a few seconds to perform facial recognition with a low level of accuracy. Therefore, an attendance system that is faster and more accurate in recognizing faces is needed with the aim of increasing better accuracy. This research was undertaken by applying the You Only Look Once (yolo) algorithm with several test scenarios to see the performance generated by the system, because yolo is one of the fastest and most accurate methods for object detection and even exceeds 2 times the capabilities of other algorithms. The yolo (you only look once) algorithm is an architecture of deep learning and an algorithm developed to detect an object in real-time. Regarding to the classification performance measurement from the training data, it indicated that the accuracy value has reached 90% thus it can be concluded that the system can work wellt.

Keywords: *Yolo, Deep Learning, Algorithm*

ABSTRAK

Deteksi wajah merupakan proses mendasar dan penting dalam bidang pengenalan wajah. Deteksi wajah ini bertujuan untuk mengetahui keberadaan dan menandai posisi wajah melalui citra yang disebut dengan bounding box. Masalah yang sering kita jumpai pada mesin absensi seringkali sulit dan membutuhkan waktu beberapa detik untuk melakukan pengenalan wajah dengan tingkat akurasi yang rendah. Oleh karena itu, diperlukan sistem absensi yang lebih cepat dan akurat dalam mengenali wajah dengan tujuan untuk meningkatkan akurasi yang lebih baik. Penelitian ini dilakukan dengan menerapkan algoritma You Only Look Once (yolo) dengan beberapa skenario pengujian untuk melihat performa yang dihasilkan oleh sistem, karena yolo merupakan salah satu metode pendeteksian objek yang tercepat dan paling akurat bahkan melebihi 2 kali kemampuan algoritma lainnya. Algoritma yolo (Anda hanya melihat sekali) adalah arsitektur pembelajaran mendalam dan algoritma yang dikembangkan untuk mendeteksi objek secara real-time. Sedangkan untuk pengukuran kinerja klasifikasi dari data training menunjukkan bahwa nilai akurasi telah mencapai 90% sehingga dapat disimpulkan bahwa sistem dapat bekerja dengan baik.

Kata Kunci: *Yolo, Deep Learning, Algoritma*

1. INTRODUCTION

The development of technology is increasing very rapidly in this day, there are many new and cutting-edge technologies in various facilities. The development of identification systems in the community or agency is very fast, one of which is the facial feature identification system of someone who has high speed and accuracy is facial recognition.

One part of the human body that is unique is the face. Everyone in the world has a different facial contour. Therefore, the face is used by everyone to be a marker of his identity so that it is easily recognized by others. In this modern era, technology is developing very rapidly. Face recognition and face detection is a technique used to perform the face recognition process on a computer.

In some agencies, attendance activities are carried out manually where absent participants need to sign the attendance column that has been provided as proof of attendance. Even though there have been many technologies provided by commercial companies by providing machine products as attendance such as fingerprint. However, the agency will bear the burden of a fairly large cost by procuring the machine, especially with the addition of maintenance costs.

However, in the modern era like now, there are many agencies or companies that use technology that encourages human work easier, one of the examples is the attendance machine that has been used by Smart integrated system companies. However, problems that are often encountered in attendance machines are often difficult and take a few seconds to perform facial recognition with a low level of accuracy and are not appropriate so that

employees have difficulty in taking attendance.

Dealing with some of the problems above, an attendance system is needed and it is faster in recognizing faces and accurately in the attendance data collection process (attendance). Therefore, the researcher is interested in making a facial recognition system for employee attendance by applying the Yolo Algorithm (You Only Look Once) which is one part of the deep learning method because yolo has a higher level of accuracy and processing speed.

Previously, it has been done in the form of image processing for face recognition using the YOLO algorithm, among others, the implementation of the you only look once (yolo) algorithm for detection of victims of natural disasters by applying yolov3 and yolov3-tiny obtained an accuracy of 92.9%, then research on attendance management system with facial recognition and GPS features using yolo on the android platform with an average accuracy value of 93.26%.

YOLO (You Only Look Once) is one of the fastest and most accurate methods for object detection, even exceeding 2 times the capabilities of other algorithms. You Only Look Once is an algorithm developed to detect an object in real-time You Only Look Once is an algorithm developed to detect an object in real-time. YOLO uses an artificial neural network (ANN) approach to detect objects in an image. This network divides an image into several regions and predicts each bounding box and probability for each region. These bounding boxes are then compared with each predicted probability. In this study, the author uses the YOLOv3 method because it has several improvements in detecting objects and a higher accuracy value than

the previous version. It is expected that this study can provide good results in detecting faces by applying the YOLO method which can detect objects more quickly.

2. RESEARCH METHODS

2.1. Research Steps

In this study, the SEMMA method was used in system development and the observation method in the data collection process. Semma stands for Sample, Explore, Modify, Model and

Assess. The abbreviation refers to the process of performing a data mining object. There are five stages for the development of this system. The following are the stages of the Semma method used for this research:

1. Data collection stage (sample)
2. Data description stage (explore)
3. Data transformation stage (modify)
4. Data modeling stage (model)

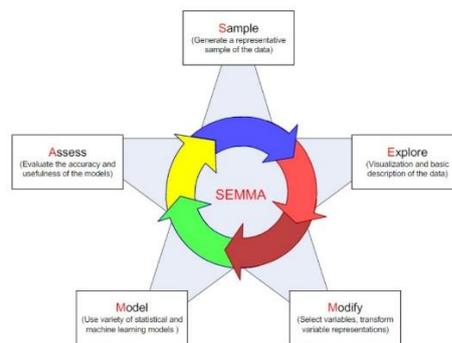


Figure 1. Stages of the semma method

2.2. Research Description

The descriptions in this study are as follow:

a. Data collection stage (sample)

In this step, the data, tools and materials needed for research were collected. Data collection was carried out using the observation method where in this case, the researcher visited the research location to directly observe various things or conditions that existed in the field. The source of data used in this study comes from employee image data taken directly at the Smart integrated system company. The data used in this study are facial data with random facial poses and free poses. The input data in this study are photos in .jpg format.

b. Data description stage (explore)

At this stage, exploration was undertaken by explaining the data in full regarding data description and data visualization. The data description will explain what the big picture of the data information is. Data visualization will show information visually from the data.

c. Data transformation stage (modify)

At this stage is to modify the data by creating, selecting and transforming variables to focus on the model selection process.

d. Data modeling stage (model)

Modified data will be modeled with a software to automatically search for reliable data combinations to predict the desired results. Existing data will be annotated data. Data annotation aims to label the image by providing a bounding box. Then,

conduct data training which aims to train the computer by processing the annotated data to form a characteristic as a consideration to reach a prediction. The next step is to implement the You Only Look Once (YOLO) algorithm for detection. Implementation is done by implementing the design results into software using the python programming language.

- e. Data evaluation stage (assess)

At this stage, an evaluation of the modeling that has been made will be carried out. Tests are carried out in real-time by using a webcam. Tests are carried out to determine the level of object accuracy with the new model that has been trained. Researchers will get results based on data that has been processed in the first stage of data collection to the modeling stage, the final results regarding the level of accuracy, precision and confusion matrix.

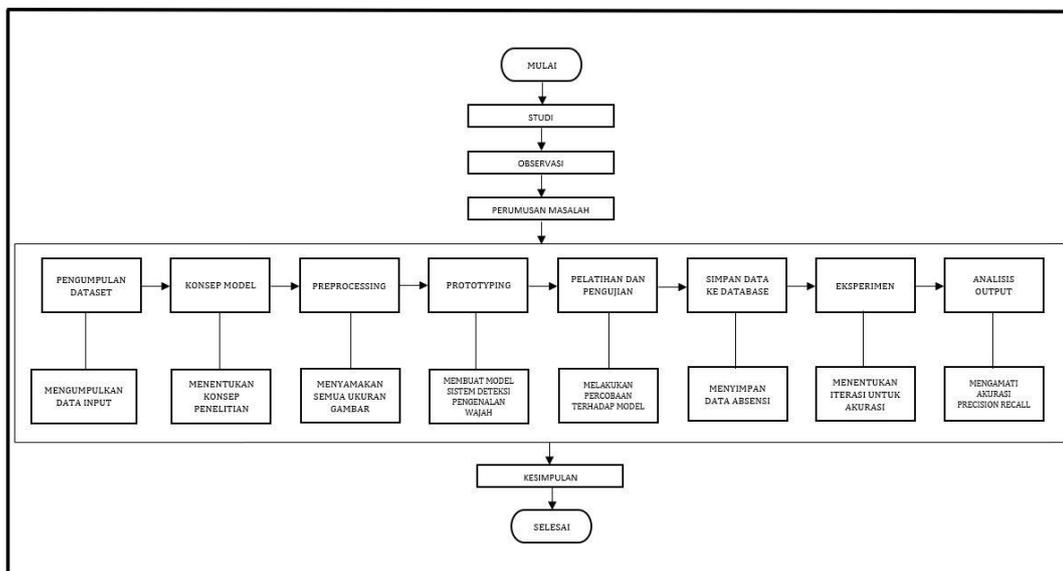


Figure 2. Thinking framework

3. RESULT AND DISCUSSION

3.1. Sample

The first stage that the researcher does is designing the system, it was undertaken by collecting a dataset of employee images that would be used for train data and test data. The data used in this study were facial data with random

facial poses and free poses. The faces would be entered into the system dataset. The dataset collection was taken directly at the smart integrated system company with input data in the form of photos in .jpg format.



Figure 3. Example of employee image dataset

3.2. Explore

After collecting data, the attribute selection was carried out on the initial data obtained. This selection aims to select and sort out what attributes should be taken or discarded.

3.3. Modify

This stage modifies the data by creating, selecting, and changing variables to focus on the model selection process by changing the image size in order to get new data with the same image size.

3.4. Model

This face recognition system was developed in the python and pycharm programming languages as software. Pycharm is an IDE application for doing programming specifically for the python programming language. Python-MySQL library is required to be able to read and process data that will be stored in a presence database using a MySQL server.

To start training using Pycharm, first create a file and then import the following libraries which will be needed during testing.

```

from_future_import print function

import pickle
import click
import os
import re
import face_recognition.api as face_recognition
import multi_processing
import intertools
import sys
import PIL.Image
import numpy as np

```

Figure 4. The syntax section of the library library

```

        # Stop the program if reached end of video
        if not has_frame:
            print('[i] ==> Done processing!!!')
            print('[i] ==> Output file is stored at: ' + os.path.join(ops_output_dir,
output url_server = "http://172.20.10.3:8000/"
cv2.wa url_get_data_image = url_server + "api/get_all_pegawai"
        url_isi_data_absensi = url_server + "api/absen_masuk"
        url_server_image = url_server + "wajah/"

        import requests

# Sets the input to the network
        net.setInput(blob)

        # Runs the forward pass to get output of the output layers
        outs = net.forward(get_outputs_names(net))

        # Remove the bounding boxes with low confidence
        faces = post_process(frame, outs, CONF_THRESHOLD, NMS_THRESHOLD)
        print(faces)

        cropped_face=[]
        for f in faces:
            if (f._len_() > 0):
                cropped = frame[f[1]:f[1]+f[3] ,f[0]:f[0]+f[2]]

                # print(cropped)
                if (len(cropped)!=0):
                    cv2.imshow('Image', cropped)
                    cropped_face.append(cropped)
                    [nama,jarak] = real_test_image(cropped, known_names,
known_face_encodings, 0.6, True)
                    if (nama != nama_sebelumnya and nama!=""):
                        cv2.imwrite('test.jpg',frame)
                        #proses kirim data ke serve
                        myobj = {'nip': nama}
                        files = {'foto': open('test.jpg', 'rb')}
                        x = requests.post(url_isi_data_absensi,files=files,
data=myobj)

                        print(x.text)
                        nama_sebelumnya = nama
                        response = x.json()
                        nama = nama + " | " + response['message']
                        draw_predict_name(frame, nama + "|" + str(jarak), f[0], f[1], f[0]
+ f[2], f[1] + f[3])
                        print('[i] ==> # detected faces: {}'.format(len(faces)))
                        print('#' * 60)

                        # initialize the set of information we'll displaying on the frame
                        info = [
                            ('number of faces detected', '{}'.format(len(faces)))
                        ]
                        for (i, (txt, val)) in enumerate(info):
                            text = '{}: {}'.format(txt, val)
                            cv2.putText(frame, text, (10, (i * 20) + 20),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.7, COLOR_RED, 2)
                        cv2.imshow(wind_name, frame)
                        key = cv2.waitKey(1)

```

The syntax of the part of the system related to face recognition is shown in the following figure: Figure The syntax part of the face recognition system

The system will search for the nearest face, if the face can be predicted it will display the identity of the employee which will then be stored into the system database according to the date and time during the attendance process, where the attendance process is carried out directly via a webcam. The data can be seen on the attendance website which can only be accessed by the admin.

```

# the process of finding the closest
# print(distances)
index_terdekat = np.where(distances == np.amin(distances))
print(index_terdekat)
print(image_to_check+ ' diprediksi ' + known_names[index_terdekat[0][0]])

if not unknown_encodings:
    # print out fact that no faces were found in image
    print_result(image_to_check, "no_persons_found", None, show_distance)

def real_test_image(image_to_check, known_names, known_face_encodings, tolerance=0.6,
show_distance=False):
    # unknown_image = face_recognition.load_image_file(image_to_check)
    unknown_image = image_to_check

```

Figure 6. syntax of the process of finding the closest face

The syntax for connecting the system to the website uses APIs so that data is automatically stored into the website as shown in the following figure:

Figure 7. Syntax of linking through api

3.5. Assess

a. Evaluation of facial recognition test result

In this section, the data on the results of testing the facial recognition system as an employee attendance presence using the You Only Look Once (yolo) algorithm where what is tested is

the performance of the system that has been designed along with the prediction results of face recognition sent to attendance management on the website. Testing is done in real-time using a webcam camera. The following are the results of the tests that have been carried out:

```

wind_name = 'face detection using YOLOv3'

cv2.namedWindow(wind_name, cv2.WINDOW_NORMAL)
output_file = ''

if args.image:
    if not os.path.isfile(args.image):
        print("[!] ==> Input image file {} doesn't exist".format(args.image))
        sys.exit(1)

    cap = cv2.VideoCapture(args.image)
    output_file = args.image[:-4].rsplit('/')[-1] + '_yoloface.jpg'
elif args.video:

    if not os.path.isfile(args.video):
        print("[!] ==> Input video file {} doesn't exist".format(args.video))
        sys.exit(1)

    cap = cv2.VideoCapture(args.video)

    output_file = args.video[:-4].rsplit('/')[-1] + '_yoloface.avi'
else:

```

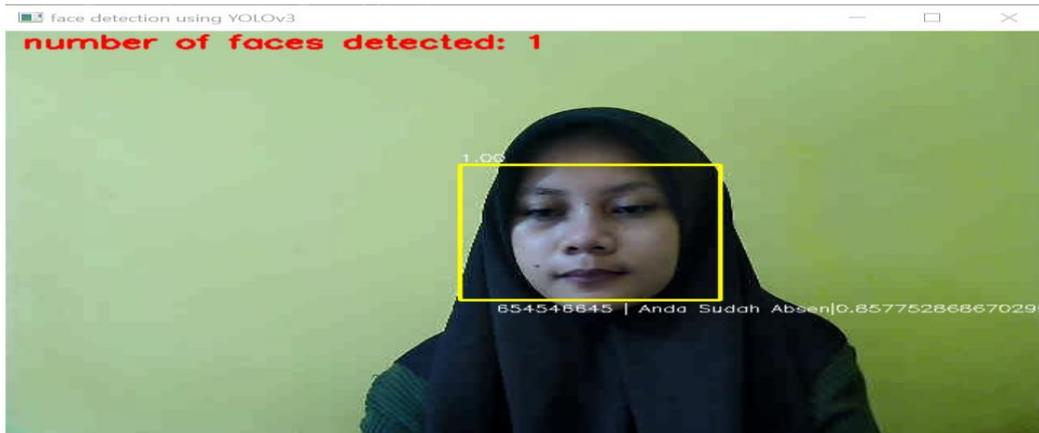


Figure 8. Test Results

From the test results display the nip and information that has successfully performed attendance, and then the data will be entered into the attendance data website as shown in Figure 9 below:

Figure 9. Website attendance data

Figure 9 is the result of the execution of the website creation program for attendance data. The design is made with several menu pages that can be used both for recording and monitoring the results that have been sent by the server

b. Evaluation of Confusion matrix

To know how accurate the results of the system that have been carried out, a trial is carried out first through evaluation and validation by using a confusion matrix calculation consisting of accuracy, precision and recall. Based on the test results of the

system that has been made, it is obtained :

		<i>Observed</i>	
		<i>True</i>	<i>False</i>
<i>Predicted Class</i>	<i>True</i>	<i>True Positive (TP)</i>	<i>False Positive (FP)</i>
	<i>False</i>	<i>False Negative (FN)</i>	<i>True Negative (TN)</i>

Figure 10. Table of confusion matrix

Thus, accuracy, precision, and recall can be formulated as follows :

- 1) Precision $\frac{TP}{TP+FP}$
- 2) Recall $\frac{TP}{TP+FN}$
- 3) Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$

Table 1. Result of *Confusion matrix*

	True	False
True	229	14
false	12	15

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{229}{229+14} = 0,94 = 94\%$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{229}{229+12} = 0,95 = 95\%$$

$$\text{Accuracy} = \frac{229+15}{229+12+15+14} = \frac{244}{270} = 0,90 = 90\%$$

So, based on the results of the confusion matrix table calculations, the results of the system that have been created are 90% accuracy, 94% precision and 95% recall. With high accuracy, precision and recall values obtained, it can be considered that the system has good performance.

4. CONCLUSION

Dealing with the result, it has been described in the previous chapters that the YOLO Algorithm can perform face detection well with high accuracy values of 90%, precision of 94% and recall of 95%. With high accuracy, precision and recall values obtained, it can be considered that the system has good performance because the accuracy results are above 85%.

5. REFERENCES

Y. M. Poysancin and A. N. Utomo, "Rancang Bangun Sistem Deteksi Wajah Dengan Metode Viola-

Jones Untuk Mengidentifikasi Identitas Seseorang," *Incomtech*, vol. 8, no. 2, pp. 69–76, 2019, [Online]. Available: <https://ejournal.istn.ac.id/index.php/incomtech/article/download/547/404/>.

M. K. Hasan, M. S. Ahsan, Abdullah-Al-Mamun, S. H. S. Newaz, and G. M. Lee, "Human face detection techniques: A comprehensive review and future research directions," *Electron.*, vol. 10, no. 19, 2021, doi: 10.3390/electronics10192354.

D. Garg, P. Goel, S. Pandya, A. Ganatra, and K. Kotecha, "A Deep Learning Approach for Face Detection using YOLO," *1st Int. Conf. Data Sci. Anal. PuneCon 2018 - Proc.*, pp. 1–4, 2018, doi: 10.1109/PUNECON.2018.8745376.

T. Elektro, U. Sam, R. Manado, and J. K. B. Manado, "Aplikasi Pengenalan Wajah Untuk Sistem Absensi Kelas Berbasis Raspberry Pi," *J. Tek. Inform.*, vol. 15, no. 3, pp. 179–188, 2020, doi: 10.35793/jti.15.3.2020.31290.

C. N. Ihsan, "Klasifikasi Data Radar Menggunakan Algoritma Convolutional Neural Network (CNN)," *DoubleClick J. Comput. Inf. Technol.*, vol. 4, no. 2, pp. 115–121, 2021.

Y. Pratama, R. Errissya, S. Akwan, and I. Irawan, "Application of YOLO (You Only Look Once) V.4 with Preprocessing Image and Network Experiment," *Int. J. Informatics Comput. Sci.*, vol. 5, no. 3, pp. 280–286, 2021, doi: 10.30865/ijics.v5i3.3386.

- M. Sarosa *et al.*, “IMPLEMENTASI ALGORITMA YOU ONLY LOOK ONCE (YOLO) UNTUK IMPLEMENTATION OF YOU ONLY LOOK ONCE (YOLO) ALGORITHM FOR,” vol. 8, no. 4, pp. 787–792, 2021, doi: 10.25126/jtiik.202184407.
- Y. Hartiwi, E. Rasywir, Y. Pratama, and P. A. Jusia, “Sistem Manajemen Absensi dengan Fitur Pengenalan Wajah dan GPS Menggunakan YOLO pada Platform Android,” vol. 4, pp. 1235–1242, 2020, doi: 10.30865/mib.v4i4.2522.
- J. Liu and X. Wang, “Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network,” *Front. Plant Sci.*, vol. 11, no. June, pp. 1–12, 2020, doi: 10.3389/fpls.2020.00898.
- X. Jiang, T. Gao, Z. Zhu, and Y. Zhao, “Real-time face mask detection method based on yolov3,” *Electron.*, vol. 10, no. 7, pp. 1–17, 2021, doi: 10.3390/electronics10070837.
- J. Redmon and A. Farhadi, “Yolo V2.0,” *Cvpr2017*, no. April, pp. 187–213, 2017, [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/9789812771728_012.
- J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” 2018, [Online]. Available: <http://arxiv.org/abs/1804.02767>.
- T. I. Hermanto and Y. Muhyidin, “Analisis Data Sebaran Bandwidth Menggunakan Algoritma Dbscan Untuk Menentukan Tingkat Kebutuhan Bandwidth Di Kabupaten Purwakarta,” *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 5, no. 2, pp. 130–137, 2020, doi: 10.36341/rabit.v5i2.1388.
- A. N. Syahrudin and T. Kurniawan, “Input Dan Output Pada Bahasa Pemrograman Python,” *J. Dasar Pemrograman Python STMIK*, no. January, pp. 1–7, 2018.
- H. Yun, “Prediction model of algal blooms using logistic regression and confusion matrix,” *Int. J. Electr. Comput. Eng.*, vol. 11, no. 3, pp. 2407–2413, 2021, doi: 10.11591/ijece.v11i3.pp2407-2413.
- B. P. Pratiwi, A. S. Handayani, and S. Sarjana, “Pengukuran Kinerja Sistem Kualitas Udara Dengan Teknologi Wsn Menggunakan Confusion Matrix,” *J. Inform. Upgris*, vol. 6, no. 2, pp. 66–75, 2021, doi: 10.26877/jiu.v6i2.6552.
- M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto, “Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking,” *IEEE Access*, vol. 8, pp. 90847–90861, 2020, doi: 10.1109/ACCESS.2020.2994222.