IJAIT (International Journal of Applied Information Technology)
http://journals.telkomuniversity.ac.id/ijait/

Generating Information of URL Based on Web Scraping Using YOLOv3 Face Recognition Technology

Lulud Annisa Ainun Mahmuddah [a, b, c], Suryo Adhi Wibowo [a, b, c, *], Gelar Budiman [c]

[a] *The University Center of Excellence for Advanced Intelligent Communications, Telkom University, Indonesia*
[b] *Image Processing and Vision Laboratory, Telkom University, Indonesia*
[c] *School of Electrical Engineering, Telkom University, Indonesia*

luludaam@student.telkomuniversity.ac.id, suryoadhiwibowo@telkomuniversity.ac.id, gelarbudiman@telkomuniversity.ac.id

ARTICLE INFO

Received June 28th, 2021
Revised February 28th, 2022
Accepted June 7th, 2022
Available online July 1st, 2022

Keywords
face recognition; web scraping; You Only Look Once (YOLO); object detection

ABSTRACT

Artificial Intelligence (AI) is a system developed to learn and apply human intelligence. Some technologies produced from the development of Al are face recognition and web scraping. Face recognition is used for identifying or verifying the identity of an individual using their face. The result of a face recognition process can be used to collect information on the internet with a web scraping technique. This paper proposes a face recognition model and web scraping system using You Only Look Once (YOLO) object detection method and Request library written in Python. The face recognition model performed fine-tuning in two hyperparameters, which are learning rate and step training. The proposed model for face recognition is using custom datasets that contain 8000 images divided into 5 classes and evaluated using the Mean Average Precision (mAP) performance parameter, while the web scraping system is evaluated using the precision rate parameter. From the test results, the best configuration was obtained at a learning rate of 0.0001 and step training of 10K. The highest mAP that is achieved is 0.90 with a recall and precision value of 0.75 for each, while the average precision rate is 0.87. The results of this paper are expected to contribute to the development of biometric security technology.

Acknowledgment

This research was supported by the Directorate of Research and Community Service PPM, Telkom University, and was supported by Basic Research Grant funded by the Ministry of Research and Technology and was supported by The University Center of Excellence Grant funded by the Ministry of Education, Culture, Research and Technology.

* Corresponding author at:
The University Center of Excellence for Advanced Intelligent, Telkom University
Jl. Telekomunikasi No. 1, Terusan Buah Batu, Bandung, 40257
Indonesia
E-mail address: suryoadhiwibowo@telkomuniversity.ac.id

ORCID ID:
First Author: 0000-0001-8710-5523

https://doi.org/10.25124/ijait.v5i02.3910
Paper_reg_number IJAIT000050205 2022

1. Introduction

Artificial Intelligence (AI) has been an interesting topic for the past few decades. AI is a system developed to learn and apply human intelligence [1]. Computer Vision is a branch of AI. Computer Vision algorithms can teach machines how to simulate human vision [2], therefore the ability to detect an object is needed. Object detection is a technique used in digital image processing to detect human faces, buildings, trees, cars, or other objects. The main purpose of object detection is to determine whether there is an object in an image.

Face recognition is a development technology from computer vision that is used to identify human faces. One of the reasons for the recent increased interest in face recognition is the need for identity verification in the digital world [3]. There are several methods that can be used for face recognition, including Convolutional Neural Network (CNN) [4], Region Based CNN (R-CNN), Faster R-CNN [5], and You Only Look Once (YOLO) [6].

Besides computer vision, one of the AI developments is data science. Data Science is a field that encompasses anything related to data cleansing, preparation, and analysis [7]. Web scraping is one of the developments in data science technology. Web scraping is a program that can extract data from a web page. There are several tools that can be used to do web scraping including Scrapy, BeautifulSoup, Selenium, and Request-HTML library. This paper designed a web scraping system based on human face identification. This system requires a web scraping system that can retrieve relevant information and is equipped with object detection methods that have high accuracy and can detect an object in real-time.

An object detection system that can detect object in real-time and have high accuracy is required to build face recognition technology. In previous research [8], The Region-Based CNN (R-CNN) method is slower in detecting an object because the convolutional network process is carried out for each proposed object without sharing computation and takes 47 seconds (on the GPU) to detect one image. Meanwhile, the Faster R-CNN method can be trained to do a share convolutional network so that it can increase the speed and accuracy of object detection [9]. However, the two methods are still not optimal for detecting objects in real-time [10].

The web scraping system requires tools that can work effectively in collecting information on the internet. Selenium is a framework used for software testing that also functions as a scraper, especially on dynamic websites [11]. However, the scraping process using Selenium takes longer than sending web requests with HTTP Requests while BeautifulSoup is assisted by the requests module to send web requests when doing the web scraping.

This paper proposes a web scraping system using Request-HTML library and YOLO method for object detection in face recognition technology. The Requests-HTML library can send web requests and extract the information from the intended web. YOLO implements a single neural network in the entire images during training and testing time [6] which is increasing the speed of object detection system in detecting objects and is very suitable to be applied to real-time object detection systems. The system will retrieve information in the form of a web URL according to the keywords generated from the face recognition process.

2. Basic Theory

2.1. You Only Look Once (YOLO)

The You Only Look Once (YOLO) object detection algorithm is the first one-stage detector based on CNN. YOLO is an object detection algorithm targeted for real-time processing. YOLO uses a single neural network to predict bounding box and class probability directly from the input image. YOLO divides the input image into 7×7 grid cells, each cell produces 2 predicted bounding boxes, then there will be 98 predicted bounding boxes. Then, each predicted bounding box will produce two confidence score values, namely a confidence score that describes the object's presence and a confidence score from the detected class prediction. Figure 1 shows YOLO system model.

YOLO performs an object detection process on each grid cell simultaneously, this is the reason why YOLO is a very fast algorithm for detecting objects [10]. There are many predicted bounding boxes that have a low confidence score of the 98 predicted bounding boxes. For object detection to work optimally, it is necessary to adjust the threshold value. YOLO will take predicted bounding boxes with the best confidence scores and eliminate predicted bounding boxes whose confidence score is below the threshold value. The process of eliminating predicted bounding boxes with low confidence scores is called Non-Max Suppression (NMS).

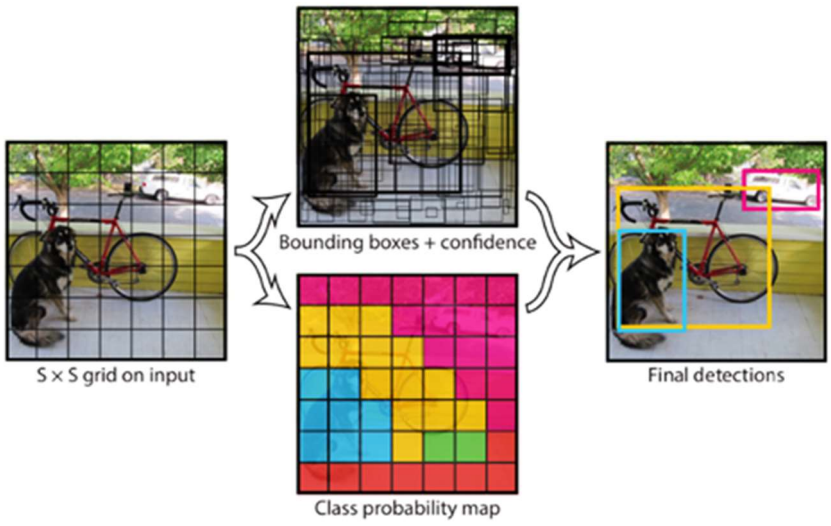

Figure 1 YOLO System Model [7]

2.2. Network Architecture

This paper is using the YOLOv3 architecture which was developed from the YOLOv1. YOLOv3 uses Darknet53 as a feature extractor and independent logistic classifier for multi-labeled classification.

Independent logistics classifier will predict the score of each class, then the threshold is used to perform multi-label classification for objects detected in the input image.

Darknet53 has 53 convolutional layers and 53 additional convolutional layers for object detection, so there are 106 fully convolutional layers. YOLOv3 performs object detection at 3 scales at 3 different layers as shown in Figure 2 [12], to solve a problem in the previous version, YOLOv1, which had difficulty detecting small objects. The 3 layers used to detect objects are layer 82, layer 94, and layer 106. The Network will down sample each layer.

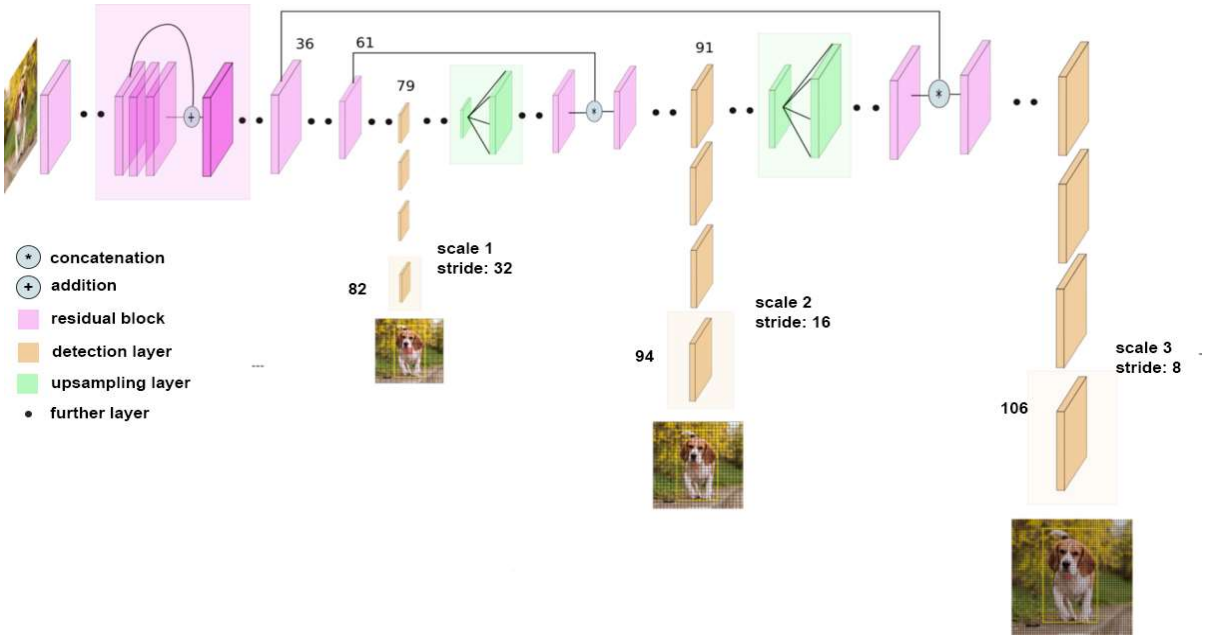

Figure 2 YOLOv3 Architecture [10]

2.3. Web Scraping

A web scraper is software that simulates human browsing on the web to collect detailed information data from different websites [13]. The scraper starts from one or more Uniform Resource Locator (URL), then downloads the content from that URL page and extracts the other URL pages it needs on the web page and puts it in the queue [14]. This process is repeated until the scraper decides to stop when the required conditions are fulfilled.

2.4. Request-HTML

Web sites are written using Hypertext Markup Language (HTML), which means that each web page is a structured document. As mentioned before, we can collect information from a web page using a web scraper. Request-HTML is a library in Python that allows users to do web scraping. This library will send a request to a web page to get information on that web page by parsing the HTML [15]. Figure 3 shows how to import Request-HTML library to a python program.

```
1  from requests_html import HTMLSession
2  session = HTMLSession()
3
4  r = session.get('https://python.org/')
5
```

Figure 3 Importing Request-HTML Library

3. Methodology

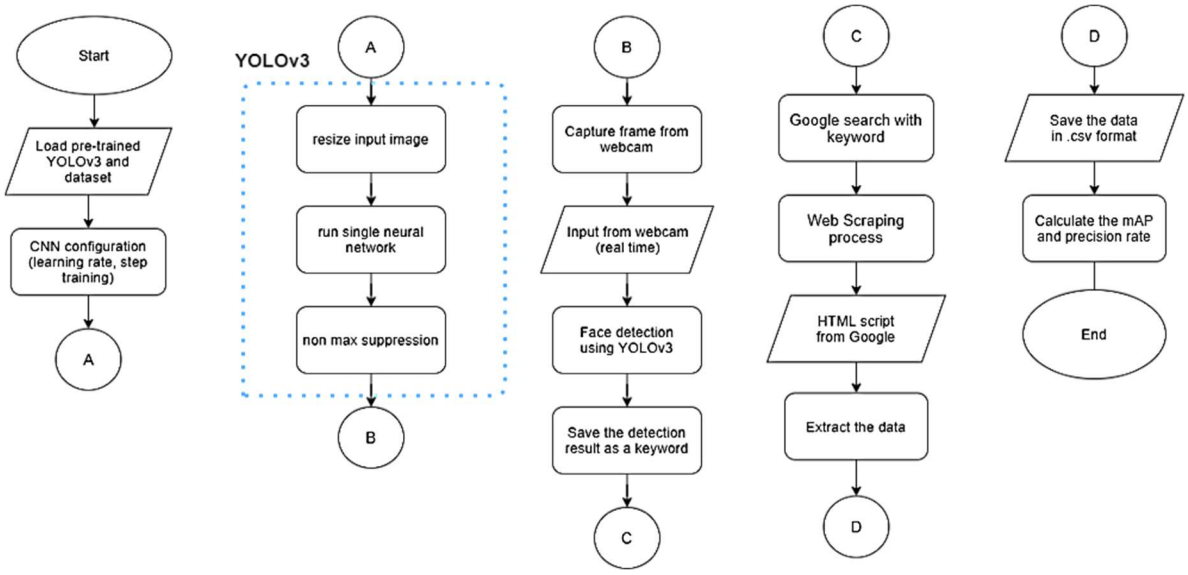

Figure 4 The Flow Diagram for Proposed Model

This research was building a model of face recognition that works in real-time using the YOLOv3 algorithm. The identity generated by the face recognition process was then used as a keyword for an Internet search using web scraping techniques. This research has a scheme as shown in Figure 4.

3.1. System Model Configuration

This research proposes a face recognition model using pre-trained weights YOLOv3 and custom datasets. YOLOv3 is a CNN architecture that has been trained using the COCO dataset which has 80 classes. The datasets used in this paper have 5 classes that consist of face images of 5 different people. Images were taken at a 60 cm distance with various facial conditions. The datasets are divided into three parts including training data, validation data, and testing data images. The systematic datasets are shown in Table 1.

Table 1 Systematic Datasets

Distance (cm)	Training Data	Validation Data	Testing Data	Facial Condition
60	1500	750	500	Wearing nothing
60	1500	750	500	Wearing glasses
60	1500	750	500	Wearing mask
60	1500	750	500	Wearing glasses and mask

The face recognition model performed fine-tuning in two hyperparameters, which are learning rate and step training. Fine-tuning is used to find the most optimal model configuration. The configuration of the face recognition system model is shown in Table 2.

Table 2 Used Hyperparameters of Face Recognition Model

Hyperparameters	Configuration A	Configuration B
Learning rate	0.001	0.0001
Step training	500.2K	10K
Batch size	64	64

Configuration A is the original configuration of YOLOv3, while Configuration B is the modification of the original one. Learning rate defines how fast the model updates its weights. A large learning rate will accelerate the training progress rate, but it may also cause the training to get stuck at a local minimum and fluctuating losses. While a small learning rate will slow down the training progress but are also more desirable when the loss keeps getting worse [16]. Step training or iteration is defined as the number of batches of data passed through the network. Increasing the number of steps in training may cause overfitting to the model. The training should be stopped when the error rate of validation data is minimum. A batch is the number of training samples or examples in one iteration.

3.2. System Workflow

From Figure 3. the system is divided into 3 steps including face detection using YOLOv3, web scraping process, and evaluation of the system.

3.2.1. Face Detection Using YOLOv3

The system uses a webcam to capture video in RGB color space in real-time which will be used for the face detection process. The captured video is then extracted into frames to make it easier for the system to detect objects. After the video input data is converted into an image, the system will perform face detection with YOLO. YOLOv3 resizes the image into 416×416 during the object detection process to speed up the training process. YOLOv3 divides the input image into 13×13 grid cells, so the 416×416 resolution is chosen because it is divisible by 13×13. Then, YOLO runs a single neural network on the image. YOLO uses the Non-Maximum Suppression (NMS) to select the best-predicted bounding box. The output face detection is a bounding box. The parameter of the bounding box that will be used for the web scraping process is the class name. An Illustration of YOLO detection process is shown in Figure 5.

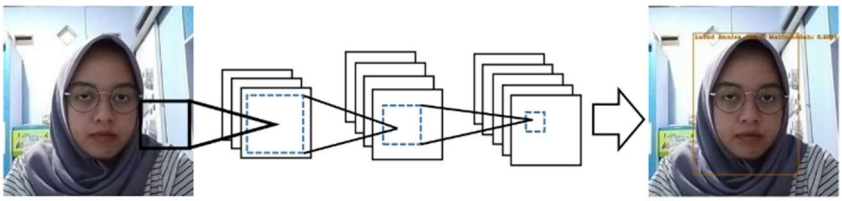

Figure 5 Illustration of YOLO Detection Process.

3.2.2. Web Scraping Process

After the face is detected by the system, the class name on the detected face will be stored as a keyword. Then the system will search for information on Google using these keywords.

title	link
Kiki Widianto Profil | Facebook	https://id-id.facebook.com/public/Kiki-Widianto
Kiki Widianto | Facebook	https://id-id.facebook.com/people/Kiki-Widianto/1000018600479
Kiki Widianto | Facebook	https://id-id.facebook.com/people/Kiki-Widianto/1000021561505
Kiki Widiyanto | Facebook	https://id-id.facebook.com/kiki.widiyanto.7

Figure 6 Web Scraping Results

The scraping process will start from www.google.com. Then, the system will parse the webpage content to collect specific information using the request-HTML library. The system designed will only take 9 URLs from Google search results for each keyword. So, the total amount of crawled URLs is 45 URLs. This paper retrieves information on website names and their URL link. The collected information is saved into a structured table in .csv format as shown in Figure 6.

3.2.3. Evaluating System

To evaluate the system, tests are performed on mAP and precision rate parameters. Mean Average Precision (mAP) is a metric used to evaluate performance in object detection models. This paper uses IoU threshold = 0.5, so the mAP that is calculated is mAP@0.5. This threshold was chosen because it can produce a good detection, if we use a threshold below 0.5, it can cause an error in the detection system. mAP value can be calculated using the average of Average Precision (AP). To get the AP value, it is necessary to calculate the precision and recall for each class. mAP value can be calculated using Equation 1 and Equation 2 [17].

$$mAP = \frac{1}{n}\sum_{k=1}^{k-n} AP_k \quad \text{Equation 1}$$

$$AP = \int_0^1 p(r)dr \quad \text{Equation 2}$$

where *n, p,* and *r* are defined as the number of classes, precision, and recall. AP represents the area under the precision-recall curve.

Precision and recall values can be calculated using the confusion matrix. The confusion matrix is visualized in a matrix that describes the performance of the classification model on test data that the true value is already known. Figure 7 shows the confusion matrix.

The recall parameter is used to calculate the sensitivity of the system, while the precision parameter is used to calculate the accuracy of the system in detecting objects. Equation 3 and Equation 4 shows how to calculate recall and precision using a confusion matrix.

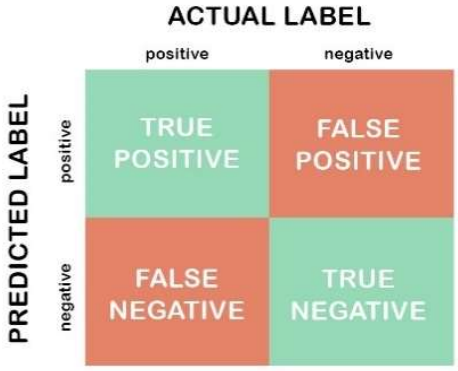

Figure 7 Confusion Matrix

$$R = \frac{TP}{TP + FN} \quad \text{Equation 3}$$

$$P = \frac{TP}{TP + FP} \quad \text{Equation 4}$$

Where *R, P, TP, FP*, and *FN* are defined as recall, precision, true positive, false positive, and false negative. True negative (TN) metric does not apply for recall

and precision calculation in object detection because there are numerous possible predictions that should not be detected in an image. As a result, TN includes all probable incorrect detections that were missed.

Precision and precision rate are the different parameters. The precision rate parameter shows the level of relevance between the crawled web and the topic being searched. Precision rate can be calculated using Equation 5 [18].

$$Pr = \frac{R}{N} \quad \text{Equation 5}$$

where R and N are defined as the number of relevant crawled web and the total crawled web.

4. Result and Discussion

The designed system has two configurations that are described in Table 2. Each configuration is evaluated using mAP and precision rate parameters.

4.1. Mean Average Precision (mAP)

To calculate the mAP value, the system performed two test scenarios including testing on precision and recall parameter. Precision and recall value can be obtained from the confusion matrix of the test result that will be detailed in Table 3.

Table 3 Confusion Matrix of Test Results

Configuration	Confusion Matrix		
	True Positive	False Positive	False Negative
A	841	1186	1159
B	1508	492	492

4.1.1. Scenario 1: Testing on Recall Parameter

Testing on the recall parameter is used to determine the level of sensitivity of the system model in detecting objects. The recall parameter calculates the success of the system in recovering information (true positive rate). The recall value of the model is shown in Figure 8.

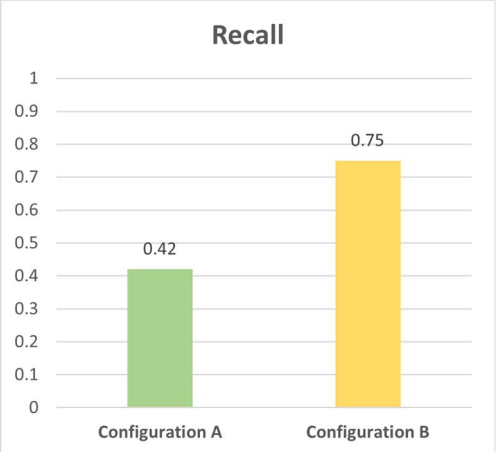

Figure 8 Recall of The Proposed System Model

In Figure 8, the recall value is increasing from Config. A to Config. B. This was caused by the fine-tuning process on the learning rate and steps in Config. B.

4.1.2. Scenario 2: Testing on Precision Parameter

Testing on the precision parameter is used to determine the accuracy of the system for detecting objects in test data images. The ideal precision value is closer to 1. This means the system detects more true positive objects than false positive objects. In Figure 9, we can see that the precision value is increasing from 0.42 to 0.75. This was also caused by the fine-tuning process.

After calculating precision and recall values, the mAP value can also be calculated. Figure 10 shows the mAP value of the proposed system model.

Configuration B obtained a higher mAP value than Configuration A, which means Config. B works more optimally. Config. A has 500.2 K steps of iteration and a 0.001 learning rate value. A very large number of iterations can cause overfitting of the system. Instead of learning, the system will only memorize the training data, while a large learning rate will cause a fluctuated loss. Figure 11 shows the results of the face recognition system using the systematic datasets shown in Table 1.

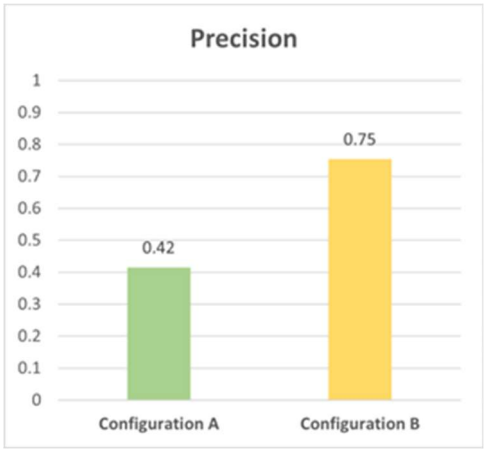

Figure 9 Precision of The Proposed System Model

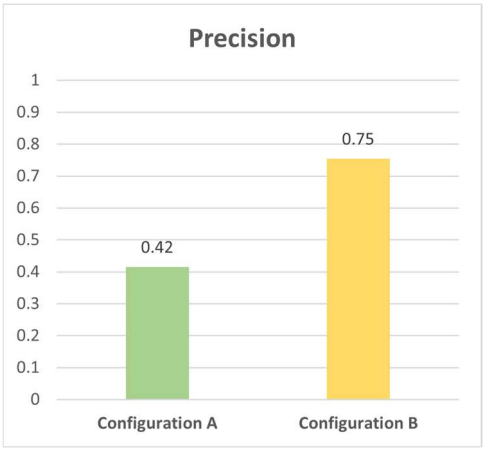

Figure 10 mAP of The Proposed System Model

4.2. Precision Rate

After the face recognition process is completed, the designed system will evaluate the web scraping system on the precision rate parameter. The system will search for information on the internet using the output from the face recognition process as keywords. Testing was conducted with 9 webs crawled for each class, bringing the total web crawled to 45. This test aims to determine the level of accuracy of the system in collecting information that is on the internet. The web

can be said to be relevant if it has information that directly leads to the used keywords.

Figure 11 Result of Face Recognition on The Proposed System Model

In Table 4, the highest precision rate value is 1 with the keywords Aprilla Firdausya Nugraha and Kiki Widianto. While the worst precision rate is 0.56 with the keyword Adhi Satriya Andrian. The precision rate value for the whole system is 0.87, this value is obtained by finding the average value of the precision rate.

Table 4 Precision Rate of Web Scraping

Keyword	Relevant Web	Crawled Web	Precision Rate
Adhi Satriya Andrian	5	9	0.56
Aprilla Firdausya Nugraha	9	9	1.00
Ivan Bagastama	8	9	0.89
Kiki Widianto	9	9	1.00
Lulud Annisa Ainun Mahmuddah	8	9	0.89
		Average Precision	0.87

5. Conclusions

In summary, this paper proposed a face recognition model and web scraping system using custom datasets. The proposed system model has obtained a mAP value of 0.90 and a precision rate of 0.87. The best configuration for the face recognition model is Configuration B which has 10K steps of iterations and a 0.0001 learning rate value. Based on the test results, the system performance improves when the learning rate of 0.001 is changed to 0.0001 and step training of 500.2K is changed to 10K, which means a large learning rate can cause a fluctuated loss while a very large number of iterations can cause overfitting to the model. For future experiments, it would be interesting if this model could be developed to perform on more varied face datasets such as adding the blurriness to the image, setting some different distances for capturing images, and covering the face with other objects so that this model could be implemented in more complex cases.

Bibliography

[1] X. Liu, "Artificial intelligence and modern sports education technology," *Proc. - 2010 Int. Conf. Artif. Intell. Educ. ICAIE 2010*, pp. 772–776, 2010, doi: 10.1109/ICAIE.2010.5641441.

[2] X. Li and Y. Shi, "Computer vision imaging based on artificial intelligence," *Proc. - 2018 Int. Conf. Virtual Real. Intell. Syst. ICVRIS 2018*, pp. 22–25, 2018, doi:

10.1109/ICVRIS.2018.00014.

[3] W. Wójcik, K. Gromaszek, and M. Junisbekov, “Face Recognition: Issues, Methods and Alternative Applications,” *Intech*, pp. 8–28, 2016.

[4] M. Coskun, A. Ucar, O. Yildirim, and Y. Demir, “Face recognition based on convolutional neural network,” *Proc. Int. Conf. Mod. Electr. Energy Syst. MEES 2017*, vol. 2018-January, pp. 376–379, 2017, doi: 10.1109/MEES.2017.8248937.

[5] H. Jiang and E. Learned-Miller, “Face Detection with the Faster R-CNN,” *Proc. - 12th IEEE Int. Conf. Autom. Face Gesture Recognition, FG 2017 - 1st Int. Work. Adapt. Shot Learn. Gesture Underst. Prod. ASL4GUP 2017, Biometrics Wild, Bwild 2017, Heteroge*, pp. 650–657, 2017, doi: 10.1109/FG.2017.82.

[6] 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.

[7] S. Kumar, N. Dhanda, and A. Pandey, “Data Science - Cosmic Infoset Mining, Modeling and Visualization,” *2018 Int. Conf. Comput. Charact. Tech. Eng. Sci. CCTES 2018*, pp. 1–4, 2019, doi: 10.1109/CCTES.2018.8674138.

[8] R. Girshick, “Fast R-CNN,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[9] S. Ren, K. He, G. Ross, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[11] G. S. Kalra, R. S. Kathuria, and A. Kumar, “YouTube Video Classification based on Title and Description Text,” *Proc. - 2019 Int. Conf. Comput. Commun. Intell. Syst. ICCCIS 2019*, vol. 2019-Janua, pp. 74–79, 2019, doi: 10.1109/ICCCIS48478.2019.8974514.

[12] A. Kathuria, “What’s new in YOLO v3?,” 2018. [Online]. Available: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b. [Accessed: 18-Jun-2021].

[13] R. Diouf, E. N. Sarr, O. Sall, B. Birregah, M. Bousso, and S. N. Mbaye, “Web Scraping: State-of-the-Art and Areas of Application,” *Proc. - 2019 IEEE Int. Conf. Big Data, Big Data 2019*, pp. 6040–6042, 2019, doi: 10.1109/BigData47090.2019.9005594.

[14] Y. Ren, “web scraping in python using SCRAPY,” *2018 15th Int. Conf. Serv. Syst. Serv. Manag.*, pp. 1–6, 2018.

[15] K. Reitz, “Request-HTML: HTML Parsing for Humans (writing Python 3)!,” 2017. [Online]. Available: https://requests.readthedocs.io/projects/requests-html/en/latest/. [Accessed: 10-Feb-2021].

[16] Y. Wu *et al.*, “Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks,” *Proc. - 2019 IEEE Int. Conf. Big Data, Big Data 2019*, pp. 1971–1980, 2019, doi: 10.1109/BigData47090.2019.9006104.

[17] D. Sharma, “Evaluation and Analysis of Perception Systems for Autonomous Driving,” KTH Royal Institute of Technology, 2020.

[18] M. S. Safran, A. Althagafi, and D. Che, “Improving relevance prediction for focused Web crawlers,” *Proc. - 2012 IEEE/ACIS 11th Int. Conf. Comput. Inf. Sci. ICIS 2012*, pp. 161–166, 2012, doi: 10.1109/ICIS.2012.61.