International Journal of Allied Research in Engineering and Technology (IJARET)
Volume.13, Number 1; January-2022;
ISSN: 2836-5631| Impact Factor: 5.52
https://zapjournals.com/Journals/index.php/ijaret
Published By: Zendo Academic Publishing

A NOVEL APPROACH TO COFFEE BEAN SORTING: YOLO ALGORITHM IMPLEMENTATIO

[1]S. Divvala, [2]R. Girshick, and [3]A. Farhadi

Article Info

Keywords: Coffee bean selection, YOLO algorithm, Raspberry Pi, automatic sorting, image understanding

Abstract

The coffee industry's growing demand for high-quality coffee beans has spurred the need for efficient and accurate coffee bean sorting methods. Traditional manual selection by humans lacks consistency due to factors like human error, lack of training, and long working hours. On the other hand, existing sorting machines can only select beans based on size, neglecting essential characteristics like shape that contribute to coffee flavor. To address these limitations, we propose an automatic coffee bean selector using the YOLO (You Only Look Once) algorithm integrated with a Raspberry Pi microcontroller.

The YOLO algorithm, known for its real-time object detection capabilities at 45 FPS, has demonstrated promising results in video analysis and image understanding tasks. We adapt this algorithm to detect coffee beans based on their shape and size, optimizing the coffee bean selection process. The Raspberry Pi, a versatile microcomputer, provides a suitable platform for running the YOLO algorithm, utilizing Python as the programming language, which aligns with the algorithm's requirements.

In this study, we develop a coffee bean selection system, named "Green Beans," to automate the process and reduce the dependency on manual labor. By integrating the YOLO algorithm with the Raspberry Pi, we aim to achieve high accuracy and efficiency in coffee bean sorting. This automation becomes crucial as the industry witnesses increasing demands, putting a strain on human resources using the manual method.

We conduct comprehensive testing of the YOLO-based coffee bean selector to evaluate its performance in detecting coffee beans. The results of the testing reveal the algorithm's effectiveness in identifying coffee beans accurately and efficiently. By leveraging the power of computer learning and modern technology, we bridge the

[1] Department of Electrical Engineering, Faculty of Engineering, Universitas Malikussaleh, Aceh, Indonesia
[2] Department of Electrical Engineering, Faculty of Engineering, Universitas Malikussaleh, Aceh, Indonesia
[3] Department of Electrical Engineering, Faculty of Engineering, Universitas Malikussaleh, Aceh, Indonesia

gap between manual and machine-based coffee bean selection, paving the way for improved coffee production quality.

1. Introduction

Following the development of the times both in the field of science and technology, which is growing, and the plant product processing industry is also multiplying, sorting is one of the stages in the processing of production and cultivation to determine the best quality [1]. Coffee consumption and demand for high-quality coffee beans have increased over the years [2]. The most essential things in selecting coffee beans based on physical vision include shape and size [3]. In the coffee industry, two methods are commonly used to determine and choose coffee bea: manually and using a machine. The first method is done manually by humans using their eyes and hands to see the shape of the coffee beans and select them by hand to determine the best beans, but this manual selection makes the quality of the coffee beans not the same; this inequality factor is influenced by long working hours, lack of training and human error. In addition, this manual method takes quite a long time and can be inefficient if you have a lot of seeds to choose from. In the second method, a sorting machine is used to select beans based on their size, but this machine cannot select based on shape which will potentially damage the taste quality of the coffee beans because the way this tool works is only to vibrate the coffee beans placed on it, therefore there is no way the right way to choose the best quality coffee beans[4]. With the development of computer learning, many algorithms have improved the relationship between video analysis and image understanding. All of these algorithms work differently on their network architectures, but have the same goal, which is to detect many objects in a complex image. In this study using the YOLO algorithm because this algorithm can detect an object in real time with a speed of 45 FPS[5]. The Raspberry Pi is a microcomputer developed by the Raspberry Pi Foundation in England. This microcomputer was created to teach basic computer science and programming to students around the world[6]. The programming language on this Raspberry Pi uses the Python programming language, which is the same programming language as the programming language used in the YOLO algorithm. At this time the coffee bean selection system still uses the manual method. This will take a lot of time and human effort. Thus, an automatic coffee bean selector is needed because the number of requests that continues to increase will burden human labor. In addition, the manual method used to select coffee beans still has low accuracy because it is very dependent on human precision. The purpose of this study was to build a coffee bean selector tool (Green Beans) using the YOLO algorithm based on a microcontroller which aims to find out how the YOLO algorithm is applied to identify coffee beans and to be able to find out the results of testing the YOLO algorithm to detect coffee beans.

2. Literature Review

2.1. Coffee

Coffee plants belong to the genus coffee with the Rubiaceae family. The family has many genera, namely Gardenia, Ixora, Cinchona and Rubia. The genus Coffee includes nearly 70 species, but only two species are cultivated on a large scale worldwide, namely Arabica coffee (Coffee Arabica) and Robusta Coffee (Coffee canephora var. robusta). Meanwhile, around 2% of total world production comes from two other coffee species, namely coffee liberica (coffee liberica) and coffee ekselsa (coffee excelsa) which are grown on a limited scale, mainly in West Africa and Asia[7]. There are four known types of coffee, namely Arabica coffee, Robusta coffee, Liberika coffee and Ekselsa coffee. Coffee groups that are known to have economic value and are commercially traded are Arabica coffee and Robusta coffee. This type of Arabica coffee has a high taste quality and lower caffeine content compared to Robusta so the price is more expensive[8]. The taste of Robusta coffee is inferior to Arabica coffee, but Robusta coffee is resistant to leaf rust disease. Therefore, the area of robusta coffee plants in Indonesia is larger than the area of Arabica coffee so that the production of robusta coffee is more. Arabica coffee planting area is limited to highland land above 1000m above sea level to avoid coffee leaf rust[9].

2.2. You Only Look Once (YOLO)

You Look Only Once (YOLO) is a deep learning that can be used for object recognition.deep learning models end-to-end designed for fast object detection, developed by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi in 2015 in their paper entitled "You Look Only Once: Unified, Real-Time Object Detection"[10]. The detection system is carried out using a repurpose classifier or localizer to perform detection. The area with the highest image score will be considered as an object[11]. YOLO applies a neural network to an image, then divides the image into regions and predicts the bounding box and probabilities for each bounding box. The probability for each bounding box is then calculated to classify it as an object or not. YOLO can perform object recognition in real-time at a speed of 45 frames per second. This approach involves a single convolutional neural network which in its detection divides the input into a grid of cells and each cell directly predicts the bounding box and object classification[12]

2.3. Deep Learning (Deep Learning)

Deep learning is part of machine learning (Machine Learning), and machine learning is part of artificial intelligence. Deep learning or deep learning is machine learning inspired by the structure of the human brain[13]. Deep learning tries to draw conclusions, much like humans analyze data based on sequential logical structures. To achieve this goal, deep learning uses a multilayer structure called a neural network (Neural Network)[14].

2.4. Machine Learning

Machine Learning is a study that applies algorithms to computer systems to be able to complete certain tasks without explicit instructions[15]. Machine Learning or machine learning means an analytical method that can handle large amounts of data by developing personal computer solving procedures [16] [17]. Machine Learning tries to imitate how human processes or intelligent beings can generalize. the hallmark of Machine Learning is the process of training, learning or training. Therefore, machine learning requires data to learn which is referred to as training [18].

2.5. OpenCV

Open Computer Vision (OpenCV) is an application programming (API) library familiar in image processing in computer vision. Computer vision is a branch of image processing that allows computers to look like humans. With this vision, the computer can recognize the observed object, while some examples of the implementation of computer vision are face recognition, face detection, face/object tracking, and line tracking. In OpenCV there are programming languages for C, C++, Python, and Java that can run on Windows, Mac, Linux, and Android designed for real-time applications, good acquisition functions for images or videos [19].

2.6. Raspberry PI

Raspberry Pi is a computer mini the size of a credit card developed by the Raspberry Pi Foundation in England for computing teaching basic in schools[20]. The Raspberry Pi is a microcomputer module that also has the digital same as on the microprocessor board. Ram with a capacity of 1 GB, port type B and port for LAN. The name Raspberry Pi is taken from the name of a fruit, to be precise Raspberry fruit Python which is the name of a language. Python is used as the main programming language on the Raspberry Pi, but other programming languages can be used on the Raspberry Pi.

2.7. Webcam

Webcam or web camera is a low-resolution digital camera that functions to take pictures or video with the help of a personal computer and is used for various purposes such as video calling chats, surveillance cameras, and as video conferencing. The process of capturing images on a webcam is the same as the process of capturing images on an ordinary digital camera, which is carried out by two types of light sensors that have different ways of working: Charge Couple Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS)[21].

2.8. Python

Python is a multipurpose interpretive programming language with a design philosophy focused on code readability. Python is claimed to be a language that combines capabilities, capabilities, with code syntax that very clear, and comes with great standard library functionality as well comprehensive. Pythons supports multi-paradigm programming[22]. Python can be used for various purposes software development and can run on various platforms operating system Python is distributed under several different licenses version. But in principle Python can obtained and used freely, even for personal purposes commercial[23]

3. Method

3.1. Design Stage

1. Study of literature, which is where the author conducts a literature review and looks for references related to the issues discussed so as to ensure that this research can be carried out and reduce errors in research.
2. Identification of needs, in this stage the author identifies and looks for the needs needed to solve the problems discussed. Identification of needs that will be prepared includes hardware and software design.
3. Design and engineering, namely designing and designing the problems that are obtained.
4. Functional testing, namely carrying out functional and system performance testing which aims to determine the level of success in building a coffee bean selector, this test consists of camera testing and raspberry pi testing.
5. Overall System Testing, which is testing by running the system that has been built by looking at the level of success of the system as a whole and if there is a system that is not going well, it will be evaluated and retested.
6. Results and discussion, in this stage the author analyzes the test data so that he finds a new idea to draw a conclusion.
7. Conclusion, namely drawing conclusions from the results of research that has been done.

3.2. Identification of Needs

This control need is very important in this research, while the required control needs consist of a Raspberry PI and a Webcam Camera and Requirements Programs are used as commands to run the system so that it works properly and correctly. In doing programming, the language used is Python. **3.3. Preparation of Design**

The preparation of the design of this tool begins with the presentation of the block diagram of the tool, then proceed with the preparation of hardware and software designs.

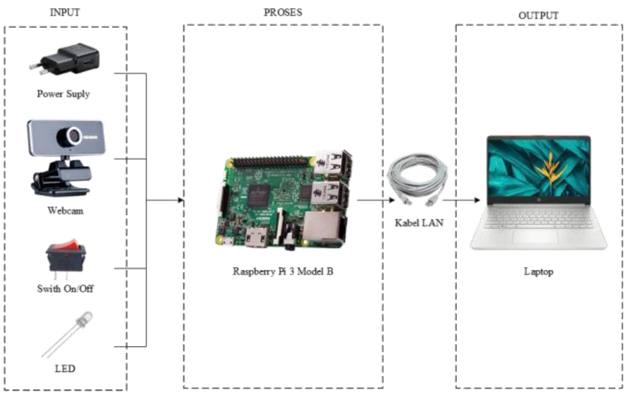

Fig. 1: system block diagram

1. Power Supply is a source of electrical power in the device to activate several components.
2. webcam functions to take pictures of coffee beans that have been placed in the container.
3. The LED functions to help the lighting on the camera.
4. The indicator light serves to provide information if the power supply has been connected to a voltage source.
5. The On/Off switch functions as a switch to activate the LED, Raspberry pi and Power Supply.

6. Raspberry PI 3 serves as the tool used to run the YOLO algorithm.
7. The LAN cable is used to connect the Raspberry pi 3 with a laptop.
8. The laptop functions to display the results of the YOLO algorithm process carried out on Raspberry PI 3 where we will find out which coffee beans are perfect and which are imperfect.

4. Result and Discussion

4.1. Detection of Perfect Coffee Beans

Table 1. Results of Detection of Perfect Coffee Beans

No	Testing	Many Beans	Average Value Percentage of Testing
1	Trial 1	6	76.166667
2	Trial 2	5	70
3	Trial 3	4	68.5
4	Trial 4	4	66.75
5	Trial 5	4	73.25
6	Trial 6	5	75.6
7	Trial 7	3	89.333333
8	Trial 8	4	89
9	Trial 9	5	83.8
10	Trial 10	3	73
	Average Test Score		76.54

From table 1 above it can be seen for the perfect coffee bean category in experiment 1 the average percentage value is 76.17% with 6 coffee beans , in experiment 2 the average percentage value decreased from the first experiment of 6.17%, namely to 70% with 5 coffee beans, then in experiment 3 the average percentage value decreased from experiment 2 of 1.5%, namely me to 68.5% with 4 coffee beans, in experiment 4 the percentage value decreased from experiment 3 by 1.75%, namely to 66.75% with 4 coffee beans, in experiment 5 the percentage value increased from experiment 4 by 8.5 %, namely to 75.25% with 4 coffee beans, in experiment 6 the percentage value increased from the fifth experiment by 0.35%, namely to 75.6% with 5 coffee beans, in experiment 7 the percentage value increased from the previous experiment of 13 .73%, namely to 89.33% with 3 coffee beans, then in experiment 8 the percentage value decreased from the previous experiment of 0.33%, namely to 89% with 4 coffee beans, in the 9th experiment the percentage value decreased again by 5 .2%, namely to 83.8% with 5 coffee beans, and finally in experiment 10 the percentage value decreased from the previous experiment of 10.8%, namely to 73% with 3 coffee beans.

4.2. Detection of Imperfect Coffee Beans

Table 2. Results of Detection of Imperfect Coffee Beans

No	Testing	Many Beans	Average Value Percentage of Testing
1	Trial 1	3	3
2	Trial 2	69.33333333	69.33333333
3	Trial 3	5	5
4	Trial 4	74.6	74.6
5	Trial 5	6	6
6	Trial 6	64.16666667	64.16666667
7	Trial 7	5	5
8	Trial 8	82.6	82.6
9	Trial 9	5	5

10	Trial 10	76	76
	Average Test Score		73.40166667

From table 2 it can be seen that the category of imperfect beans in the first experiment got an average percentage value of 69.33% with 3 coffee beans, in experiment 2 the average percentage value increased by 5.27%, namely 74.6% with 5 coffee beans, in experiment 3 the average percentage value decreased by 10.43%, namely to 64.17% with 6 coffee beans, in experiment 4 the percentage value increased by 18.43%, namely 82.6 % with 5 coffee beans, in experiment 5 the percentage value decreased by 6.6% j adi 76% with 5 coffee beans, in experiment 6 the percentage value decreased by 2.4% to 73.6% with 5 coffee beans, in experiment 7 the percentage value decreased by 7.8% to 65.8% with 5 coffee beans coffee, in the 8th experiment the percentage value increased by 18.45% to 84.25% with 4 coffee beans, in the 9th experiment the percentage value decreased by 11.25% to 73% with 4 coffee beans, in the 10th experiment the percentage value decreased decreased by 2.33% to 70.67% with 6 coffee beans.

4.3. Detection of Undetected Coffee Beans

Table 3. Results of Detection of Undetected Coffee Beans

No	Testing	Many Beans	Average Value Percentage of Testing
1	Trial 1	1	100
2	Trial 2	0	0
3	Trial 3	0	0
4	Trial 4	0	0
5	Trial 5	0	0
6	Trial 6	0	0
7	Trial 7	0	0
8	Trial 8	1	100
9	Trial 9	1	100
10	Trial 10	0	0
	Average Test Value		30

In table 3 it can be seen the percentage value of coffee beans that are not detected by the YOLO Algorithm only in experiment 1 to get a percentage value of 100% with 1 coffee bean, in experiment 8 to get a percentage value of 100% with 1 coffee bean and experiment 9 to get a percentage value of 100% with 1 coffee bean. The YOLO algorithm cannot detect some coffee beans because the coffee beans are too close together

4.4. Detection of Coffee Beans Detected by Two Categories

Table 4. Results of Detection of Beans Detected by Two Categories

No	Testing	Many Beans	Average Value Average Percentage Testing
1	Trial 1	0	0
2	Trial 2	0	0
3	Trial 3	0	0
4	Trial 4	1	25
5	Trial 5	1	26
6	Trial 6	0	0
7	Trial 7	2	37
8	Trial 8	1	59
9	Trial 9	0	0
10	Trial 10	1	51

Average Test Value	19.8

In table 4 it can be seen that the coffee beans detected in two categories by the YOLO Algorithm are in experiment 4 with an average percentage value of 25% with 1 coffee bean, experiment 5 with an average percentage value of 26% with 1 coffee bean, trial 7 with an average percentage value of 37% with 2 coffee beans, experiment 8 with an average percentage value of 59% with 1 coffee bean and trial 10 with an average value a percentage of 51% with 1 coffee bean. The YOLO algorithm classifies coffee beans into two categories due to errors during the labeling process and the unique shape factor of the coffee beans.

4.5. Coffee Bean Detection Time

The time range needed by the YOLO Algorithm in each experiment is as follows

Table 5. Detection time for each trial

No	Trial	Time		Total
		Start	Stop	
1	Trial 1	23:01:02	23:02:14	00:01:12
2	Trial 2	23:04:18	23:05:24	00:01:06
3	Trial 3	23:07:15	23:08:20	00:01:05
4	Trial 4	23:10:19	23:11:22	00:01:03
5	Trial 5	23:13:01	23:14:04	00:01:03
6	Trial 6	23:16:12	23:17:15	00:01:03
7	Trial 7	23:18:41	23:19:45	00:01:04
8	Trial 8	23:21:30	23:22:31	00:01:01
9	Trial 9	23:24:19	23:25:20	00:01:01
10	Trial 10	23:26:44	23:27:46	00:01:02

In table 5 it can be seen the time used by the YOLO Algorithm to detect coffee beans in each experiment. In experiment 1 the time was used to detect 10 coffee beans with a time span of 1 minute 12 seconds, in experiment 2 the time was used to detect 10 coffee beans with a time span of 1 minute 6 seconds, then in experiment 3 the time was used to detect 10 coffee beans, namely 1 minute 5 seconds, in experiments 4, 5 and 6 the time used to detect 10 coffee beans is the same, namely 1 minute 3 seconds, in experiment 7 the time used to detect 10 coffee beans is 1 minute 4 seconds, in experiment 8 and 9 the time used to detect 10 coffee beans is 1 minute 1 second and in experiment 10 the time used to detect 10 coffee beans is 1 minute 2 seconds. All of these experiments are calculated when entering commands on the Raspberry Pi terminal and are said to have been detected when the coffee bean image appears on the display screen. The length of time for detecting the YOLO Algorithm using the Raspberry Pi 3 is due to the hardware factor of the Raspberry Pi 3 which does not support the process of detecting the YOLO Algorithm, where the YOLO algorithm requires assistance from the GPU to carry out the detection process so that the time and accuracy of the YOLO Algorithm can be more optimal.

5. Conclusion

The conclusions that can be drawn from the results of the tests that have been carried out are as follows:

1. It can be concluded in this study that the YOLO algorithm can maintain an accuracy rate of detection success of 75% overall with categories namely perfect coffee beans, imperfect coffee beans perfect and some undetectable coffee beans as well as some two categories of coffee beans detected.
2. The detection results used the YOLO algorithm for 10 trials with an average percentage value of 76.54% for the perfect coffee bean category of 43 coffee beans, then the average percentage of imperfect coffee beans was 73.40% with a lot of 48 coffee beans and 3 coffee beans were not detected with an average percentage value of 1% and 6 coffee beans included in the two categories with an average percentage value of 19.8%. In this study there are also differences in the time span caused by the Raspberry Pi 3 hardware which does not support the process of detecting the YOLO Algorithm, where the YOLO algorithm requires

assistance from the GPU to carry out the detection process so that the time and accuracy of the YOLO Algorithm can be more optimal. In this section you should present the conclusion of the paper. Conclusions must focus on the novelty and exceptional results you acquired. Allow a sufficient space in the article for conclusions. Do not repeat the contents of Introduction or the Abstract. Focus on the essential things of your article.

References

Y. Kansrini, D. Febrimeli, and P. W. Mulyani, "Tingkat Adopsi Budidaya Yang Baik (Good Agriculture Practices) Tanaman Kopi Arabika Oleh Petani Di Kabupaten Tapanuli Selatan," *Paradig. Agribisnis*, vol. 3, no. 1, p. 36, 2020, doi: 10.33603/jpa.v3i1.3957.

E. Winarni, R. D. Ratnani, and I. Riwayati, "Pengaruh jenis pupuk organik terhadap pertumbuhan tanaman kopi," *Momentum*, vol. 9, no. 1, pp. 35–39, 2013.

M. García, J. E. Candelo-Becerra, and F. E. Hoyos, "Quality and defect inspection of green coffee beans using a computer vision system," *Appl. Sci.*, vol. 9, no. 19, 2019, doi: 10.3390/app9194195.

A. Fatih, "Desain dan simulasi mesin sortir biji kopi kering dengan sistem gerakan engkol," *J. Crankshaft*, vol. 4, no. 1, pp. 19–28, 2021, doi: 10.24176/crankshaft.v4i1.5901.

M. Azhad, "Vehicle Detection and Tracking using YOLO and DeepSORT," pp. 23–29, 2021.

I. P. Idehen *et al.*, "Development and Testing of a 5G Multichannel Intelligent Seismograph Based on Raspberry Pi," 2022.

N. Rosniar, I. Perdana, and S. F. Hamama, "Klasifikasi Jenis Serangga dan Peranannya pada Tanaman Kopi di Kampung Kenawat – Bener Meriah," *Semin. Nas. Multi Disiplin Ilmu UNAYA*, pp. 264–272, 2019. [8] Standar Nasional Indonesia, "Biji kopi," 2008.

A. U. Fisik, "Standart umum pengujian mutu pada biji kopi," pp. 1–23.

W. Liu *et al.*, "SSD : Single Shot MultiBox Detector."

M. L. Aziz, "Perancangan Sistem Deteksi Objek Secara Real-Time Menggunakan Metode YOLO pada Robot ALMubarok_MK4," vol. 2020, p. 196, 2020.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once : Unified , Real-Time Object Detection." [13] "Mengenal Teknologi Artificial Intelligence, Machine Learning, dan Deep Learning - PDF Free Download.pdf." .

S. Song, T. Liu, H. Wang, B. Hasi, C. Yuan, and F. Gao, "Using Pruning-Based YOLO v3 Deep Learning Algorithm for Accurate Detection of Sheep Face," pp. 1–16, 2022.

C. R. Gunawan, N. Nurdin, and F. Fajriana, "Design of A Real-Time Object Detection Prototype System with YOLOv3 (You Only Look Once)," *Int. J. Eng. Sci. Inf. Technol.*, vol. 2, no. 3, pp. 96–99, 2022, doi: 10.52088/ijesty.v2i3.309.

F. Hutter, *Automated Machine Learning*. .

S. Permana, M. Andriani, and D. Dewiyana, “Production Capacity Requirements Planning Using The Capacity Method Requirement Planning,” *Int. J. Eng. Sci. Inf. Technol.*, vol. 1, no. 4, 2021, doi: 10.52088/ijesty.v1i4.165.

A. Roihan, P. A. Sunarya, and A. S. Rafika, “Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper,” *IJCIT (Indonesian J. Comput. Inf. Technol.*, vol. 5, no. 1, pp. 75–82, 2020, doi: 10.31294/ijcit.v5i1.7951.

M. R. Saxena, A. Pathak, A. P. Singh, and I. Shukla, “Real time object detection using machine,” vol. 11, no. 1, pp. 16–19, 2019.

A. Gabelly and F. Pradana, “WEB GUI UNTUK MENGONTROL TIRAI,” pp. 6–11.

J. Ali, “SISTEM SECURITY WEBCAM DENGAN MENGGUNAKAN MICROSOFT VISUAL,” vol. 1, no. 2, pp. 46–58, 2016.

D. N. Zuraidah, M. F. Apriyadi, A. R. Fatoni, M. Al Fatih, and Y. Amrozi, “Menelisik Platform Digital Dalam Teknologi Bahasa Pemrograman,” *Teknois J. Ilm. Teknol. Inf. dan Sains*, vol. 11, no. 2, pp. 1–6, 2021, doi: 10.36350/jbs.v11i2.107.

M. Mahasin and I. A. Dewi, “Comparison of CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0 Backbones on YOLO V4 as Object Detector,” *Int. J. Eng. Sci. Inf. Technol.*, vol. 2, no. 3, pp. 64–72, 2022, doi: 10.52088/ijesty.v2i3.291.