

A Novel Design of Error Backpropagation Algorithm for Ingredient Mixing Process Tamarind Turmeric Herb

Mila Fauziyah^a, Supriatna Adhisuwigno^a, Bagus Fajar Afandi^{a,*}, Anindya Dwi Risdhayanti^a

^a Department of Electrical Engineering, State Polytechnic of Malang, Soekarno Hatta Street No. 9, Malang, 65141, Indonesia

Corresponding author: *1741170092@student.polinema.ac.id

Abstract— The goal of this study is to determine the best picture pattern for the tamarind turmeric herb. So far, the taste and color of tamarind turmeric herb have not been consistent, as they are impacted by maturity, and the amount of Turmeric used. The error backpropagation technique, which is commonly used in Content-based image retrieval systems, will be used to recognize image patterns. The main goal is to capture various portions of the tamarind turmeric herb during the extraction procedure. The camera is used to classify the tamarind turmeric herb product, process it into 5x5 pixels, and average the RGB value to obtain stable RGB values in each category, which are then fed into the Error Backpropagation algorithm. The most appropriate and fastest Error Backpropagation algorithm procedure will be found and implemented in a real-time computer. The first way will be to train the algorithm with ten data by changing neurons, layer, momentum, and learning rate, and the second technique will be to test the algorithm with ten data. The results of the training and testing procedure show that the two hidden layers can recognize 100% of inputs, with three input layers for R, G, and B values, ten neurons in the first and second hidden layers, and one output layer with Learning rate 0.5 and Momentum 0.6 as a parameter. Dark yellow is the best image pattern standard for tamarind turmeric herb, with RGB values in the range from 255, 103, 32 to 255, 128, 48.

Keywords— Tamarind turmeric herb; CBIR systems; error backpropagation; RGB Image.

Manuscript received 5 Feb. 2022; revised 3 Jun. 2022; accepted 8 Aug. 2022. Date of publication 30 Jun. 2023. IJASEIT is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

The culture of the Indonesian people is to consume herbal medicines, one of which is the tamarind turmeric herb. Currently, this culture is becoming a trend due to the Covid-19 Pandemic. Tamarind turmeric herb contains vitamin C, which comes from the color of Turmeric [1]. Vitamin C can maintain the body's immunity to avoid transmission of the covid-19 virus [2]– [4]. Consuming 500 mg of curcumin every day can increase the body's immunity. The tamarind turmeric herb's taste depends on the product's color [5], [6]. The color of the herbal medicines will determine the vitamin C consumed.

One of the success factors in the tamarind turmeric herb can be seen from the color produced, where the yellow color of the tamarind turmeric herbal medicine should not be too bright. Although the composition of Turmeric used is under the standards, the resulting color is always different. This depends on the maturity level of the Turmeric used, but until now, there is no definite standard to see, measure and determine whether the color is in accordance with the

standards given by the business owner. Mixing turmeric and tamarind ingredients on a monitored basis can be an alternative that can help obstacles in the herbal medicine business. The system will check each material's weight and then monitor the resulting color results. Researchers have used artificial neural networks as research material from 1987 until now [7], [8]. Error Backpropagation, the art of the Neural Network, has been tested in many studies[9]–[16]. Image Processing is used to process the data before it is processed by the Artificial Neural Network [17]–[27]. One of the results that can be obtained from Image Processing is the RGB value [28]–[30].

Previous research concluded that BEP could recognize input well. Therefore, in this study, the application of BEP will be carried out to classify tamarind turmeric herb production and as a standard for determining the production of tamarind turmeric herb. So, this research would provide good accuracy and precision to provide standard products even though the level of maturity of Turmeric is different and produced by others. This research will discuss the Backpropagation training and testing algorithm so that the best architecture can recognize colors from production and

provide color standards so that the herbal products have the same results.

II. MATERIALS AND METHOD

A. Tamarind Turmeric Herb

Tamarind turmeric herb is a typical Indonesian drink made from Turmeric and tamarind that spices, and medicinal ingredients. Tamarind turmeric herb contains vitamin C from the color of Turmeric [1]. Several studies revealed that adding turmeric extract significantly affected the parameters of sediment height, antioxidants, color, taste, aroma, and overall acceptance of tamarind turmeric herb [5]. So, the addition of Turmeric has a significant impact on tamarind turmeric herbs production.

B. Error Backpropagation

Backpropagation has multiple units present in one or more hidden layers. The figure below is a backpropagation architecture with n inputs (plus a bias), a hidden layer consisting of p units (plus a bias), and m output units.

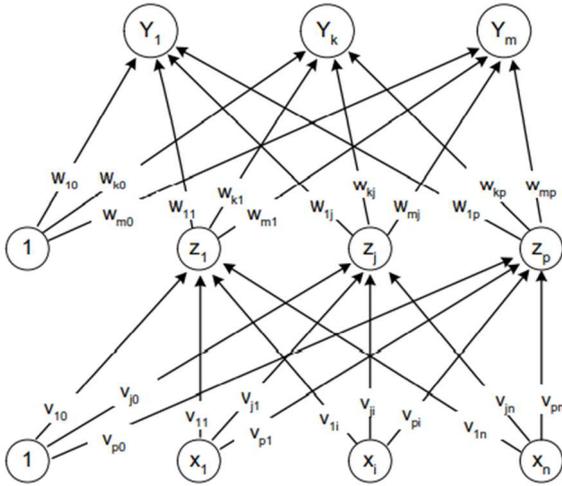


Fig. 1 Error Back Propagation

The training algorithm for a network with one hidden layer is as follows:

Step 0: Initialize all weights with small random numbers.

Step 1: If the termination condition is not met, do steps 2 – 9.

Step 2: For each pair of training data, do steps 3 – 8.

Phase I: Forward propagation.

Step 3: Each input unit receives the signal and forwards it to the hidden unit above it.

Step 4: Count all outputs in hidden units z_j ($j = 1, 2, \dots, p$).

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad (1)$$

$$z_j = f(z_{net_j}) = \frac{1}{1 + e^{-z_{net_j}}} \quad (2)$$

Step 5: Count all network outputs in the unit y_k ($k = 1, 2, \dots, m$).

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad (3)$$

$$y_k = f(y_{net_k}) = \frac{1}{1 + e^{-y_{net_k}}} \quad (4)$$

Phase II: Backward propagation.

Step 6: Calculate factor δ output units based on the error in each output unit y_k ($k = 1, 2, \dots, m$).

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad (5)$$

δ_k is the unit of error that will be used in changing the weight of the layer below it (step 7).

Calculate the rate of change of weight w_{kj} (which will be used later to change the weight w_{kj}) with the Learning rate (Lr).

$$\Delta w_{kj} = Lr \delta_k z_j \quad (6)$$

$$k = 1, 2, \dots, m; j = 0, 1, \dots, p$$

Step 7: Calculate factor δ hidden units based on errors in each hidden unit z_j ($j = 1, 2, \dots, p$).

$$\delta_{net_j} = v_{j0} + \sum_{k=1}^m \delta_k w_{kj} \quad (7)$$

Factor δ hidden unit:

$$\delta_j = \delta_{net_j} f'(z_{net_j}) = \delta_{net_j} z_j (1 - z_j) \quad (8)$$

Phase III: Weight Change.

Step 8: Calculate all changes in weight by adding momentum Mc .

Change in the weight of the line leading to the output unit:

$$w_{kj}(\text{new}) = w_{kj}(\text{old}) + \Delta w_{kj} + Mc (w_{kj}(\text{old}) - w_{kj}(\text{old} - 1)) \quad (9)$$

$$(k = 1, 2, \dots, m; j = 0, 1, \dots, p)$$

Change the weight of the line leading to the hidden unit:

$$v_{ji}(\text{new}) = v_{ji}(\text{old}) + \Delta v_{ji} + Mc (v_{ji}(\text{old}) - v_{ji}(\text{old} - 1)) \quad (10)$$

$$(j = 1, 2, \dots, p; i = 0, 1, \dots, n)$$

After the training is complete, the network can be used for pattern recognition. In this case, only forward propagation (steps 4 and 5) is used to determine the network output.

C. Image Processing

An image is a two-dimensional plane. Mathematically, a continuous function of light intensity in the two-dimensional plane is called an image. The light source illuminates the object, then reflects it. This light reflection can be captured by optical devices such as the human eye and cameras so that the shadow from the object can be recorded. Image processing uses a computer to turn an original image into a better-quality image that humans or machines can easily interpret. Pattern recognition groups numerical and symbolic data (including images) automatically by a computer to recognize an object in the image. A pattern is an entity that is defined and can be identified through its characteristics. These characteristics are used to distinguish one pattern from another. Pattern recognition in this study uses a statistical pattern recognition approach. The statistical approach uses probability science theories and statistics, as shown in the figure below. Its statistical distribution determines the characteristics possessed by a pattern.

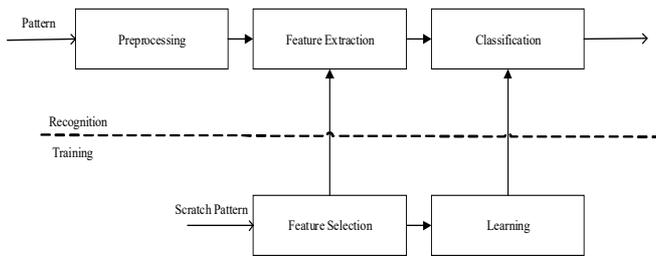


Fig. 2 Image Processing

D. RGB Color Model

The color model provides a standard way of specifying a specific color by defining a 3D coordinate system and a subspace containing all the colors that can be constructed in a given model. Any color that can be specified using the model will correspond to a single point in the subspace it defines. Each color model is oriented towards a specific hardware (RGB, CMY, YIQ), or image processing (HSI) application.

In the RGB model, an image consists of three independent image planes, one in each of the primary colors: red, green, and blue. Specifying a particular color is done by specifying the amount of each of the primary components present.

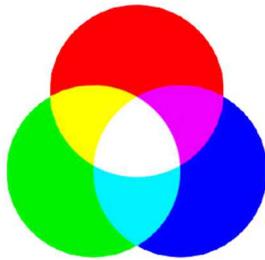


Fig. 3 RGB Color Model

E. Data Normalization

The primary process in the analysis is normalization to compare data from RGB (0-255) value to range (0.1-0.9) because this research uses a sigmoid activation function. It is important to make sure that the data being compared is comparable. This research uses the Min-Max normalization method. In this method, each piece of data is minus by the minimum value, then multiple by 0.8, then divided by the maximum value minus the minimum value, plus 0.1. The formulation of this method is as follows:

$$Value = \left(\frac{0,8 * (dataValue - minValue)}{maxValue - minValue} \right) + 0,1 \quad (11)$$

F. Research Design

The flowchart diagram in Fig. 4 shows the steps and procedures of this paper. In CBIR systems, herb retrieving systems require a query image at the input (Iq) to compare with the information from the database (Ii). If $Iq > Ii$, the system will compute RGB histogram peak and edge.

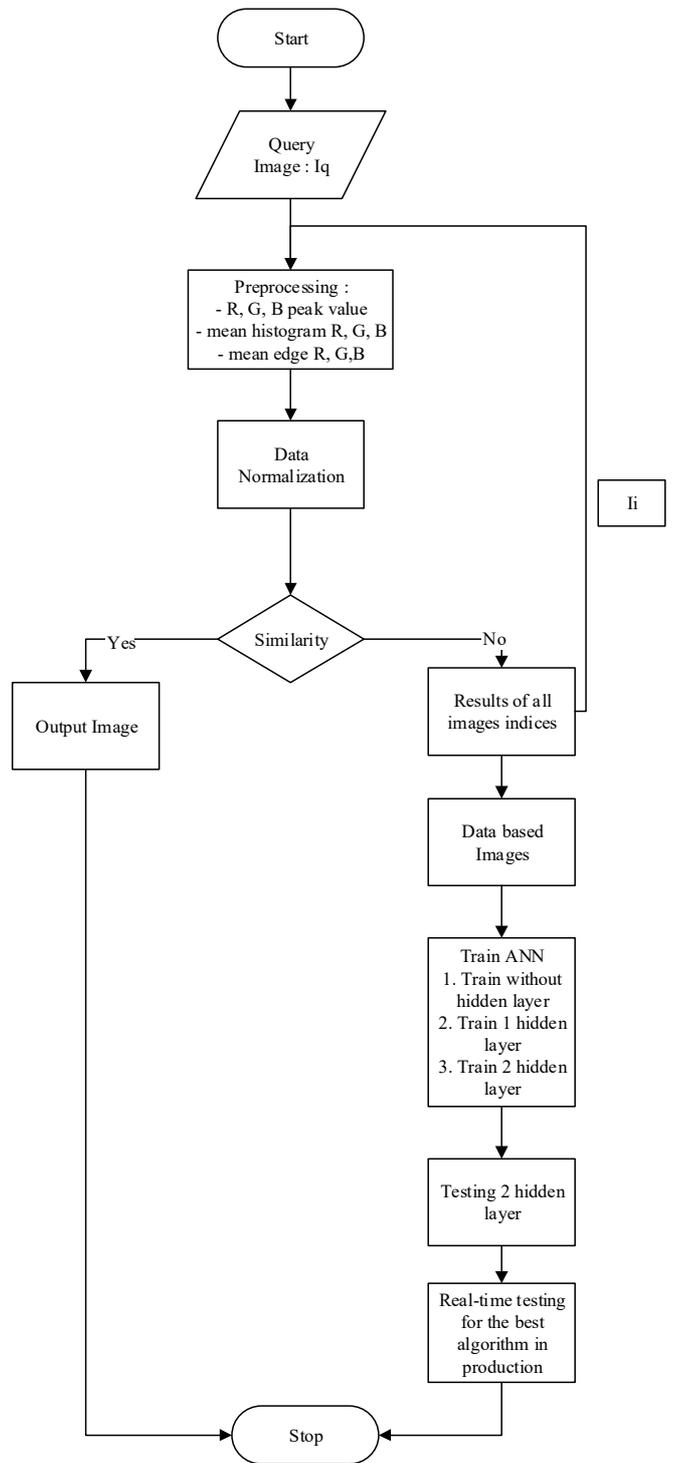


Fig. 4 Research Design

III. RESULTS AND DISCUSSION

A. Image Processing

The research data used is the RGB value of the tamarind turmeric herb production. The research used 20 data (10 testing data and 10 training data) that were classified into 3 colors (dark yellow, yellow, and light yellow).

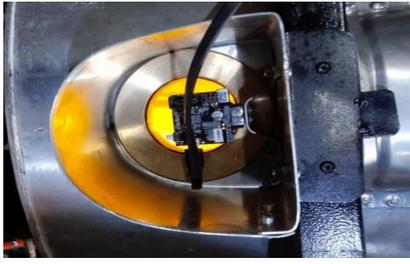


Fig. 5 Image data retrieval

Fig. 5 shows image data retrieval to be compared with the database. All of these data were obtained from observations in the home industry, and the results of color classification are shown in Table I.

TABLE I
IMAGE PROCESSING

Dark Yellow			Yellow			Light Yellow		
R	G	B	R	G	B	R	G	B
255	109	36	255	133	39	255	162	45
255	113	34	255	135	43	255	165	47
255	113	41	255	134	41	255	170	47
255	107	48	255	131	42	255	170	46
255	102	42	255	158	40	255	178	52
255	107	41	255	140	36	255	166	46
255	104	32	255	144	34	255	180	50
255	103	33	255	147	37	255	168	51
255	117	35	255	150	41	255	180	46
255	128	36	255	151	44	255	174	45
255	106	39	255	130	39	255	160	45
255	111	40	255	135	43	255	164	47
255	106	35	255	133	41	255	169	47
255	108	35	255	134	42	255	170	46
255	107	32	255	156	40	255	178	52
255	109	32	255	141	36	255	167	46
255	108	35	255	143	34	255	181	50
255	107	33	255	149	37	255	168	51
255	105	32	255	151	41	255	180	46
255	106	39	255	150	44	255	177	45

The ideal color standard for tamarind turmeric herb is dark yellow with an RGB value of 255, 103, 32 to 255, 128, 48, as shown in Table I.

B. Data Normalization

After processing the image, the RGB value needs to be normalized before being used as input of Back Error Propagation, the normalization using equation 11 is shown in Table below.

TABLE II
DATA NORMALIZATION DARK YELLOW

Dark Yellow		
R	G	B
0.9000	0.4420	0.2129
0.9000	0.4545	0.2067
0.9000	0.4545	0.2286
0.9000	0.4357	0.2506
0.9000	0.4200	0.2318
0.9000	0.4357	0.2286
0.9000	0.4263	0.2004
0.9000	0.4231	0.2035

0.9000	0.4671	0.2098
0.9000	0.5016	0.2129
0.9000	0.4325	0.2224
0.9000	0.4482	0.2255
0.9000	0.4325	0.2098
0.9000	0.4388	0.2098
0.9000	0.4357	0.2004
0.9000	0.4420	0.2004
0.9000	0.4388	0.2098
0.9000	0.4357	0.2035
0.9000	0.4294	0.2004
0.9000	0.4325	0.2224

TABLE III
DATA NORMALIZATION YELLOW

Yellow		
R	G	B
0.9000	0.5173	0.2224
0.9000	0.5235	0.2349
0.9000	0.5204	0.2286
0.9000	0.5110	0.2318
0.9000	0.5957	0.2255
0.9000	0.5392	0.2129
0.9000	0.5518	0.2067
0.9000	0.5612	0.2161
0.9000	0.5706	0.2286
0.9000	0.5737	0.2380
0.9000	0.5078	0.2224
0.9000	0.5235	0.2349
0.9000	0.5173	0.2286
0.9000	0.5204	0.2318
0.9000	0.5894	0.2255
0.9000	0.5424	0.2129
0.9000	0.5486	0.2067
0.9000	0.5675	0.2161
0.9000	0.5737	0.2286
0.9000	0.5706	0.2380

TABLE IV
DATA NORMALIZATION LIGHT YELLOW

Light Yellow		
R	G	B
0.9000	0.6082	0.2412
0.9000	0.6176	0.2475
0.9000	0.6333	0.2475
0.9000	0.6333	0.2443
0.9000	0.6584	0.2631
0.9000	0.6208	0.2443
0.9000	0.6647	0.2569
0.9000	0.6271	0.2600
0.9000	0.6647	0.2443
0.9000	0.6459	0.2412
0.9000	0.6020	0.2412
0.9000	0.6145	0.2475
0.9000	0.6302	0.2475
0.9000	0.6333	0.2443
0.9000	0.6584	0.2631
0.9000	0.6239	0.2443
0.9000	0.6678	0.2569
0.9000	0.6271	0.2600
0.9000	0.6647	0.2443
0.9000	0.6553	0.2412

C. Training

The network is trained without a hidden layer by changing the Lr and Mc values to get the least error value. We can use the Lr 0.5 and Mc 0.6 values from the training results because it has the least MSE. After getting the Mc and Lr values from training without a hidden layer, testing is done by changing the neurons from 1 to 10. The results of training one hidden layer with Mc 0.6 Lr 0.5. From the training results, we can use ten neurons because it produces the least error than other neurons.

After obtaining the number of neurons, retraining is carried out to determine the Lr and Mc values. The results of training one hidden layer with ten neurons. We use the Lr 0.5 and Mc 0.6 values because it has a small MSE and a small number of epochs. Then training for two hidden layers is carried out to compare whether there is a difference between 1 hidden layer and two hidden layers with parameters Mc 0.6, Lr 0.5, and the first hidden network 10. The results of training two hidden layers are shown in Table V.

TABLE V
TRAINING TWO HIDDEN LAYERS WITH PREVIOUS FIRST LAYER NEURONS, LR AND MC BY CHANGING SECOND LAYER NEURONS

Neuron	MSE	Gradient	Epoch
1	4.43e-06	0.00185	7
2	3.56e-06	0.00225	33
3	6.80e-06	0.00396	18
4	4.48e-06	0.00224	9
5	1.36e-06	0.00371	18
6	1.42e-09	0.000730	9
7	7.58e-06	0.00303	10
8	6.38e-07	0.00242	10
9	6.23e-06	0.00679	7
10	8.26e-06	0.00196	19

Time (second)	Explanation	Percentage	Class
1	Reached	100%	10-10-10
1	Reached	100%	10-10-10
1	Reached	100%	10-10-10
1	Reached	100%	10-10-10
1	Reached	100%	10-10-10
1	Reached	100%	10-10-10
1	Reached	100%	10-10-10
1	Reached	100%	10-10-10
1	Reached	93%	10-8-10
1	Reached	100%	10-10-10

Determination of the best network from the performance of the largest network is also viewed from the ability to recognize each class in the classification. So that the best network can be able to recognize all classes with a learning rate of 0.5 and a momentum of 0.6. epoch count of 19 with a training time of 1 second. From the training, it was found that a network with two hidden layers 10-10 has a small MSE value and can recognize 100% input. Then a test will be carried out by comparing one hidden network with ten neurons and two hidden networks with 10-10 neurons. The Lr and Mc parameters are the same, namely 0.5 and 0.6.

D. Testing

Table VI manifests the retrieval results in testing with two hidden layers. The results show that the network has a small MSE value and can recognize 100% input.

TABLE IV
TESTING TWO HIDDEN LAYERS

MSE	Gradient	Epoch
1.69e-09	3.02e05	6
Explanation	Percentage	Class
Reached	100%	10-10-10

E. Real-Time Testing

Fig. 6 shows real-time testing using data training with the best algorithm in simulation before. The results of training in production are shown in Table VII. The test results in Table 8 of 2 hidden layers showed that the network could recognize all colors.

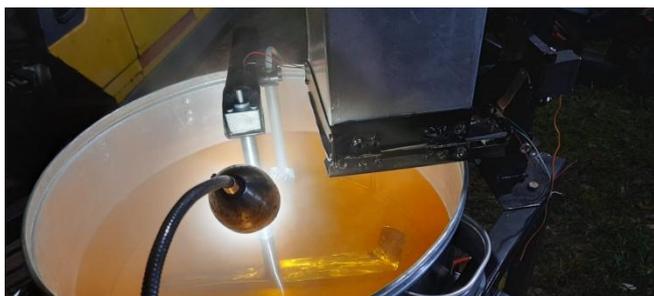


Fig. 6 Real-time testing

TABLE VII
REAL-TIME TESTING THE BEST ALGORITHM IN PRODUCTION

Water Volume	Turmeric Weight	Result
31 L	500 gr	Light Yellow
31 L	1000 gr	Yellow
31 L	1500 gr	Dark Yellow

RGB Value	Explanation
255, 165, 41	Match
255, 150, 39	Match
255, 110, 40	Match

IV. CONCLUSION

From the research above, several conclusions can be drawn. First, training without a hidden layer cannot recognize input at all. Second, training with 1 and 2 hidden layers shows the best architectural results, namely two hidden layers 3-10-10-1 with MSE 8.26e-06. The third, test results from 1 and 2 hidden layers found that two hidden layers can recognize all inputs. Fourth, from the training and testing results, it can be concluded that the best network or architecture for this research is two hidden layers 3-10-10-1 with Lr 0.5 and Mc 0.6. The best color standard for tamarind turmeric herb is dark yellow with an RGB value of 255, 103, 32 to 255, 128, 48.

ACKNOWLEDGMENT

Technical Implementation Unit supports this work in Research and Community Service (UPT-P2M) through Featured Research Programs 2021.

REFERENCES

- [1] F. Y. Kurniawan, M. Jalil, A. Purwatoro, B. S. Daryono, and Purnomo, "Jamu Kunir Asem: Ethnomedicine Overview by Javanese Herbal Medicine Formers in Yogyakarta," *J. Jamu Indones.*, vol. 6, no. 1, pp. 8–15, 2021, doi: 10.29244/jji.v6i1.211.
- [2] S. A. Astuti, F. Juwita, and A. Fajriyah, "Pengaruh Pemberian Kunyit

- Asam terhadap Penurunan Intensitas Nyeri Haid,” *Indones. J. Midwifery*, vol. 3, no. 2, p. 143, 2020, doi: 10.35473/ijm.v3i2.618.
- [3] H. Setyoningsih, Y. Pratiwi, A. Rahmawati, H. M. Wijaya, R. N. Lina, and K. Kudus, “Penggunaan Vitamin Untuk Meningkatkan,” vol. 4, no. 2, pp. 136–150, 2021.
- [4] S. N. Hidayah, N. Izah, and I. D. Andari, “Peningkatan Imunitas dengan Konsumsi Vitamin C dan Gizi Seimbang Bagi Ibu Hamil Untuk Cegah Corona Di Kota Tegal,” *J. ABDINUS J. Pengabd. Nasant.*, vol. 4, no. 1, pp. 170–174, 2020, doi: 10.29407/ja.v4i1.14641.
- [5] L. Fatmawati, Y. Syaiful, and K. Nikmah, “Kunyit Asam (Curcuma Doemstica Val) Menurunkan Intensitas Nyeri Haid,” *Journals Ners Community*, vol. 11, no. 1, pp. 10–17, 2020.
- [6] N. A. Q. A’yunin, U. Santoso, and E. Harmayani, “Kajian kualitas dan aktivitas antioksidan berbagai formula minuman jamu kunyit asam,” *J. Teknol. Pertan. Andalas*, vol. 23, no. 1, pp. 37–48, 2019.
- [7] R. Mangal, A. V. Nori, and A. Orso, “Robustness of neural networks: A probabilistic and practical approach,” *Proc. - 2019 IEEE/ACM 41st Int. Conf. Softw. Eng. New Ideas Emerg. Results, ICSE-NIER 2019*, no. i, pp. 93–96, 2019, doi: 10.1109/ICSE-NIER.2019.00032.
- [8] R. P. Lippmann, “An introduction to computing with neural nets,” *ACM SIGARCH Comput. Archit. News*, vol. 16, no. 1, pp. 7–25, 1988, doi: 10.1145/44571.44572.
- [9] Y. Miyata and S. Nakajima, “Application of back propagation to hospital patient outcomes,” *2020 IEEE 9th Glob. Conf. Consum. Electron. GCCE 2020*, pp. 791–792, 2020, doi: 10.1109/GCCE50665.2020.9291829.
- [10] L. R. Reddy, P. Patel, and S. K. Rajendra, “Utilization of resilient back propagation algorithm and discrete wavelet transform for the differential protection of three phase power transformer,” *2020 21st Natl. Power Syst. Conf. NPSC 2020*, 2020, doi: 10.1109/NPSC49263.2020.9331861.
- [11] F. Guo, L. Zhang, and X. Liu, “An Optimized Back Propagation Neural Network Based on Genetic Algorithm for Line Loss Calculation in Low-voltage Distribution Grid,” *Proc. - 2020 Chinese Congr. CAC 2020*, pp. 663–667, 2020, doi: 10.1109/CAC51589.2020.9327754.
- [12] Y. Ayyappa, A. Bekkanti, A. Krishna, P. Neelakanteswara, and C. Z. Basha, “Enhanced and Effective Computerized Multi Layered Perceptron based Back Propagation Brain Tumor Detection with Gaussian Filtering,” *Proc. 2nd Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2020*, pp. 58–62, 2020, doi: 10.1109/ICIRCA48905.2020.9182921.
- [13] J. Yan and H. Zhu, “Image Based Localization Algorithm Using Similarity Measurements and Backpropagation Neural Network,” *ICEICT 2020 - IEEE 3rd Int. Conf. Electron. Inf. Commun. Technol.*, pp. 379–382, 2020, doi: 10.1109/ICEICT51264.2020.9334204.
- [14] S. Das, A. Wahi, S. Sundaramurthy, N. Thulasiram, and S. Keerthika, “Classification of knitted fabric defect detection using Artificial Neural Networks,” *Proc. 2019 Int. Conf. Adv. Comput. Commun. Eng. ICACCE 2019*, 2019, doi: 10.1109/ICACCE46606.2019.9079951.
- [15] S. Cao and Y. Zhong, “A Methodology of Determining Weight Ratios of Different Question Types Based on Back Propagation Neural Network,” *Proc. 2020 IEEE Int. Conf. Adv. Electr. Eng. Comput. Appl. AEECA 2020*, pp. 164–168, 2020, doi: 10.1109/AEECA49918.2020.9213570.
- [16] R. Mukhaiyar and R. Safitri, “Implementation of artificial neural network: Back propagation method on face recognition system,” *2019 16th Int. Conf. Qual. Res. QIR 2019 - Int. Symp. Electr. Comput. Eng.*, pp. 1–5, 2019, doi: 10.1109/QIR.2019.8898276.
- [17] F. Simmross-Wattenberg *et al.*, “OpenCLIPER: An OpenCL-Based C++ Framework for Overhead-Reduced Medical Image Processing and Reconstruction on Heterogeneous Devices,” *IEEE J. Biomed. Heal. Informatics*, vol. 23, no. 4, pp. 1702–1709, 2019, doi: 10.1109/JBHI.2018.2869421.
- [18] S. Pang *et al.*, “SpineParseNet: Spine Parsing for Volumetric MR Image by a Two-Stage Segmentation Framework with Semantic Image Representation,” *IEEE Trans. Med. Imaging*, vol. 40, no. 1, pp. 262–273, 2021, doi: 10.1109/TMI.2020.3025087.
- [19] O. Huang *et al.*, “MimickNet, Mimicking Clinical Image Post-Processing under Black-Box Constraints,” *IEEE Trans. Med. Imaging*, vol. 39, no. 6, pp. 2277–2286, 2020, doi: 10.1109/TMI.2020.2970867.
- [20] R. Al Mukaddim, R. Ahmed, and T. Varghese, “Subaperture Processing-Based Adaptive Beamforming for Photoacoustic Imaging,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 68, no. 7, pp. 2336–2350, 2021, doi: 10.1109/TUFFC.2021.3060371.
- [21] R. Malladi, G. Kalamangalam, N. Tandon, and B. Aazhang, “Identifying Seizure Onset Zone from the Causal Connectivity Inferred Using Directed Information,” *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 7, pp. 1267–1283, 2016, doi: 10.1109/JSTSP.2016.2601485.
- [22] M. A. A. Mosleh, A. A. Al-Yamni, and A. Gumaei, “An automatic nuclei cells counting approach using effective image processing methods,” *2019 IEEE 4th Int. Conf. Signal Image Process. ICSIP 2019*, pp. 865–869, 2019, doi: 10.1109/SIPROCESS.2019.8868753.
- [23] A. Van Opbroek, H. C. Achterberg, M. W. Vernooij, and M. De Bruijne, “Transfer learning for image segmentation by combining image weighting and kernel learning,” *IEEE Trans. Med. Imaging*, vol. 38, no. 1, pp. 213–224, 2019, doi: 10.1109/TMI.2018.2859478.
- [24] S. Fadaei and A. Rashno, “A Framework for Hexagonal Image Processing Using Hexagonal Pixel-Perfect Approximations in Subpixel Resolution,” *IEEE Trans. Image Process.*, vol. 30, pp. 4555–4570, 2021, doi: 10.1109/TIP.2021.3073328.
- [25] B. Stimpel, C. Syben, F. Schirmacher, P. Hoelter, A. Dorfler, and A. Maier, “Multi-Modal Deep Guided Filtering for Comprehensible Medical Image Processing,” *IEEE Trans. Med. Imaging*, vol. 39, no. 5, pp. 1703–1711, 2020, doi: 10.1109/TMI.2019.2955184.
- [26] R. D. Myers, “Detection Of Skin Cancer Using Image Processing Techniques Chandrasah,” *Science (80-)*, vol. 179, no. 4080, p. 1349, 2016.
- [27] M. Rasamuel, L. Khacef, L. Rodriguez, and B. Miramond, “Specialized visual sensor coupled to a dynamic neural field for embedded attentional process,” *SAS 2019 - 2019 IEEE Sensors Appl. Symp. Conf. Proc.*, 2019, doi: 10.1109/SAS.2019.8705979.
- [28] X. Song, S. Jiang, L. Herranz, and C. Chen, “Learning effective RGB-D representations for scene recognition,” *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 980–993, 2019, doi: 10.1109/TIP.2018.2872629.
- [29] I. Kurniastuti and A. Andini, “Determination of RGB in Fingernail Image As Early Detection of Diabetes Mellitus,” *Proc. - 2019 Int. Conf. Comput. Sci. Inf. Technol. Electr. Eng. ICOMITEE 2019*, vol. 1, pp. 206–210, 2019, doi: 10.1109/ICOMITEE.2019.8920876.
- [30] W. Reinert, “A miniaturized RGB-laser light engine,” *Proc. - 2020 IEEE 8th Electron. Syst. Technol. Conf. ESTC 2020*, 2020, doi: 10.1109/ESTC48849.2020.9229809.