The 1st International Seminar August 2022
The Changing Role of Knowledge and Living Sustainability in ASEAN Community
Website: https://proceeding.unpkediri.ac.id/index.php/ip

Classification of Lung CT-Scan Images for Covid-19 Detection Using Texture Feature Extraction and Naive Bayes Algorithm

[1]Indera Cahyo Wibowo, [2,]*Abd, Charis Fauzan
Computer Science, Universitas Nahdlatul Ulama Blitar
E-mail: [1]inderacw19@gmail.com, [2]abdcharis@unublitar.ac.id
*Corresponding Author

Abstract—Covid-19 diagnostic testing is divided into 2 approaches, namely laboratory-based approaches such as Swab Tests, PCR Tests, and Rapid Tests. To minimize transmission, a second approach is used, namely a non-laboratory approach that uses imaging diagnostic tools such as x-rays and Computed Tomography (CT). It is hoped that the use of this non-laboratory approach can be an alternative for detecting the Covid-19 virus and will also minimize the spread of the covid virus because at the time the diagnosis process takes place there is little direct contact with patients. Chest CT scan has a high sensitivity for diagnosing Covid-19. A CT scan uses the transmission of x-rays through the patient's chest, which is then processed with high-resolution medical images. In this study, the classification of Covid-19 images through CT-Scan images of the lungs using texture feature extraction and the naive Bayes classifier algorithm is a system that enters information in the form of input images and compares them with images in the database. The data set used is 200 CT-Scan images of Covid-19 lungs and 200 CT-Scan images of Non-Covid-19 lungs. The training data uses 150 Covid-19 images and 150 Non-Covid-19 images, while the rest is used as test data. The first method used is feature extraction, namely mean intensity, standard deviation, skewness, energy, entropy, and smoothness. After successfully obtaining the feature, it is calculated using the Naive Bayes classifier algorithm. The results obtained from this study get an Accuracy rate of 54%, Precision of 55.88%.

Keyword—Covid-19 Detection, Feature Extraction, Naive Bayes,

Corresponding Author:
Abd. Charis Fauzan
Computer Science
Universitas Nahdlatul Ulama Blitar
Email: abdcharis@unublitar.ac.id

I. INTRODUCTION

At the beginning of 2020, the world was shocked by the occurrence of an outbreak of an infectious disease with unknown causes, which began with a Chinese report to the World Health Organization (WHO) that there were several pneumonia patients in Wuhan City, Hubei Province, China. Initial allegations are related to a market that sells fish, marine animals, and various other species. The cause of this virus began to be revealed and many experts identified the virus as a coronavirus [1]. Coronavirus is an RNA virus with a particle size of 120-160 nm. This virus initially only infects animals, including bats and camels. However, with time this virus began to spread to humans, the initial assumption was that in China there were humans who ate bat meat until they finally contracted the virus. After this incident, the spread of this virus in humans grew very rapidly, even to several countries outside China. To minimize the spread of the virus, Covid-19 is currently being tested. Where Covid-19 diagnostic testing is divided into 2 approaches. Namely laboratory-based approaches such as Swab tests, PCR tests, and Rapid tests. The second approach is to use a non-laboratory (computing) approach, using imaging diagnostic tools such as x-rays and Computed Tomography (CT). Chest CT scan has a high sensitivity for diagnosing Covid-19. CT scan uses x-ray transmission through the patient's chest, which is then processed with high-resolution medical images [2]. With a system like this, it is proven to be accepted by much medical personnel, because of the high level of prediction and accuracy. This CT-Scan aims to see images of the lungs and respiratory tract. Through the results of the CT-Scan, it is known whether the patient has symptoms of infection or inflammation that occurs in the respiratory tract and lungs. It is hoped that the use of this non-laboratory approach can be an alternative for the detection of the Covid-19 virus and will also minimize the spread of the covid virus because at the time the diagnosis process takes place there is little direct contact with patients [3].

Previous research [16], also used a non-laboratory approach for Covid-19 detection by using data in the form of X-Ray images and the Deep Residual Network method. While in research [17] researchers used data in the form of CT-Scan images and used Multi-Class texture analysis. However, in this study, there are differences in the use of methods, namely using the feature extraction method and the Naive Bayes classifier algorithm. In this study, we will use the texture feature extraction method and the Naive Bayes algorithm. Texture feature extraction is used to extract features from an object in the image that you want to recognize/differentiate from other objects. The features that have been extracted are then used as parameters used to distinguish objects from one another at the identification/classification stage [4]. In this study, the Naive Bayes classifier algorithm will also be used. The naive Bayes algorithm is a classification method that uses probability to predict opportunities based on experience or using existing training data. Naive Bayes has the main characteristic, namely a very strong (naive) assumption of the independence of each condition or event [5].

Although the Naive Bayes algorithm has some similarities with the KNN algorithm which can process large amounts of data. However, Naive Bayes has several advantages, namely to carry out the classification process, it does not require large amounts of training data, and is fast in performing calculations [6]. Based on this, the formulation of the problem in this study is how to classify CT-Scan images of the lungs to detect the Covid-19 virus using the texture feature extraction method and the Naive Bayes classifier algorithm which can perform early detection of Covid-19 cases through image data input. CT scan of the lungs. Texture feature extraction is used to find parameter values in the form of mean intensity, standard deviation, skewness, energy, entropy, and also smoothness. The naive Bayes algorithm is used for the data classification process which is determined as Covid or Non-Covid class. In the implementation of the system, this time the author uses Netbeans 8.0.2 and JDK 1.8.0 to classify data sets that have gone through preprocessing using the Naive Bayes algorithm, and the data set used comes from Tongji Medical College in the form of 200 CT-Scans Covid-19 and 200 Non-Covid-19 CT Scan. This research is expected able to become an alternative system for detecting the Covid-19 virus disease using a non-laboratory (computing) approach. Some of the purposes of making an early detection application system for Covid-19 are to help detect Covid-19 cases early, and to know the process of extracting image features. This research is expected able to become an alternative system for detecting the Covid-19 virus disease using a non-laboratory (computing) approach.

II. RESEARCH METHOD

System Design

Building an intelligent system for early detection of Covid-19 requires a system design for detailed and regular application implementation. Figure 1 is a system design applied to an intelligent system. The intelligent system flow begins with the test data input process, pre-processing with feature extraction in the form of mean intensity, standard deviation, skewness, energy, entropy, and also smoothness. After successfully getting the value from the feature extraction process, then the calculation is carried out using the Naive Bayes classifier method and implemented in the program. For the process of evaluating the system and getting the accuracy value, a comparison is made between the original data set value and the results from the Naive Bayes classifier.

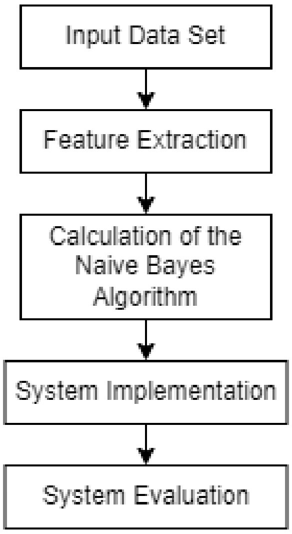

Figure 1. Flowchart of System

Data Set

The study was conducted with the initial step of data collection, the data used is an open source lung CT-Scan image from Tongji Medical College with the URL link https://github.com/UCSD-AI4H/COVID-CT. There are 2 data sets used, namely 200 CT-Scan Covid-19 and 200 CT-Scan Non-Covid-19. This data will later be used for data sets in the database as well as for training data and testing data. Figure 2 is an example of the data to be used, namely a CT-Scan of the lungs.

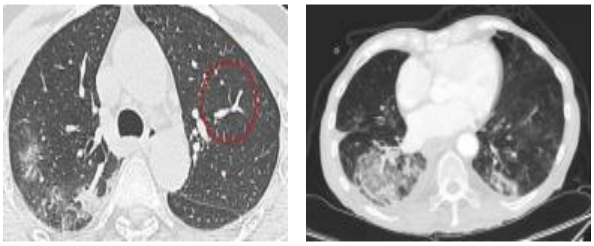

Figure 1. Results CT Scan of the Lungs for Covid (Left) and Non-Covid (Right)

Pre-Processing

Pre-processing performs the feature extraction process, which looks for the average value of intensity (mean), standard deviation, skewness, energy, entropy, and smoothness. The flow of the pre-processing stage is shown in Figure 3.

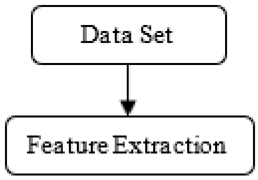

Figure 2. Flowchart of Pre-Processing

The process in pre-processing is the extraction of texture features in the training data, the first feature is the mean intensity (mean). The following is the source code and formula for calculating the mean intensity feature which can be seen in Equation 1.

$$m = \sum_{i=0}^{L-1} i \, . p(i) \qquad (1)$$

The second feature that calculated is the Standard Deviation. The following is the source code and formula for calculating the standard deviation feature which can be seen in Equation 2.

$$\sigma = \sqrt{\sum_{i=1}^{L-1} (i-m)^2 p(i)} \qquad (2)$$

The third feature that calculated is skewness. The following is the source code and formula for calculating the skewness feature which can be seen in Equation 3.

$$skewness = \sum_{i=1}^{L-1} (i-m)^3 p(i) \qquad (3)$$

The fourth feature that counts is Energy. The following is the source code and formula for calculating energy features which can be seen in Equation 4.

$$energy = \sum_{i=0}^{L-1} [p(i)]^2 \qquad (4)$$

The fifth feature that calculated is Entropy. The following is the source code and formula for calculating the entropy feature which can be seen in Equation 5.

$$entropi = -\sum_{i=0}^{L-1} p(i) \, \log_2 (p(i)) \qquad (5)$$

The sixth feature that counts is Smoothness. The following is the source code and formula for calculating the smoothness feature which can be seen in Equation 6.

$$R = 1 - \frac{1}{1+\sigma^2} \qquad (6)$$

Naive Bayes Calculation

Naive Bayes Classifiers is a classification method using probability and statistical methods proposed by British scientist Thomas Bayes, which predicts opportunities based on experience or pre-existing training data. In the naive Bayes classifier calculation process, there are several calculation processes as shown in Figure 4.

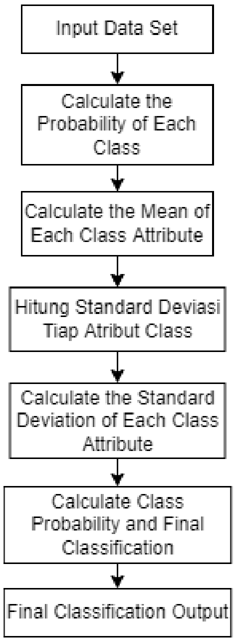

Figure 3. Calculation Stage of Naive Bayes

The first step of Naïve Bayes is Calculating the probability of each class aims to determine the greatest probability of each existing class. Shown in Equation 7.

$$P(E) = \frac{X}{N} \qquad (7)$$

Then, calculating the mean of each class attribute aims to get the average value of each existing class. Shown in Equation 8.

$$m = \sum_{i=0}^{L-1} i \,.\, p(i) \qquad (8)$$

After that, calculating the standard deviation of each class attribute serves to determine the standard deviation of each existing class. Shown in Equation 9.

$$\sigma = \sqrt{\sum_{i=1}^{L-1} (i - m)^2 p(i)} \qquad (9)$$

Then, calculating the normal distribution of each class attribute serves to determine the distribution of each existing class. Shown in Figure 10.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \quad (10)$$

Last, calculating probability values by inputting testing data serves to predict the maximum class value that will be determined as a result of the testing data by multiplying the value of the class probability (Covid & Non-Covid) with the value obtained by the Normal Distribution in each feature.

System Evaluation

Evaluation of the system in this study using the Confusion Matrix. The Confusion Matrix is a table with 4 different combinations of predicted values and actual values. Four terms represent the results of the classification process in the confusion matrix, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). This method represents the results of the classification using a matrix that can be seen in Table 1.

Table 1. Confusion Matrix

Correct Classification		***Classified as***	
		+	-
	+	*True Positive*	*False Positive*
	_	*False Negative*	*True Negative*

True Positives are the number of positive records that were successfully classified as positive, while false positives were positive records that were incorrectly classified as negative. Meanwhile, false negatives are negative records that were incorrectly classified as positive, and true negatives are negative records that have been successfully classified as negative records. The confusion matrix test method can produce calculations with 4 outputs, including:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (11)$$

$$Presisi = \frac{TP}{TP+FP} \times 100\% \quad (12)$$

$$Sensitifitas = \frac{TP}{TP+FN} \times 100\% \quad (13)$$

$$Spesifitas = \frac{TN}{TN+FP} \times 100\% \quad (14)$$

III. RESULT AND DISCUSSION

Pre-Processing

The process in pre-processing is the extraction of texture features in the training data. 6 features are calculated in this discussion, namely, mean intensity (mean), standard deviation, skewness, energy, entropy, and smoothness. In the following, the data from the calculation of feature extraction can be seen in Table 2, Table 3, Table 4, Table 5, Table 6, and Table 7.

Table 2. Feature Extraction Results Average Intensity

Image To -	Feature Extraction Results Average Intensity
1	173.899
2	69,052
3	213.869
4	130.917
5	169,843
6	185,265
7	143.926
8	210.78
9	145.874
10	34,009
...	...
291	124,284
292	129,770
293	137.517
294	126.892
295	143.765
296	106.219
297	151,315
298	138,444
299	162,941
300	140,761

Table 3. Standard Deviation Feature Extraction Results

Image to-	Standard Deviation Feature Extraction Results
1	1.197
2	0.814
3	0.850
4	1.337
5	1,108
6	0.944
7	1.175
8	0.761
9	1,135
10	0.366
...	...
291	1.03

292	1,146
293	1.047
294	1.084
295	0.502
296	0.553
297	1,153
298	1,237
299	1.126
300	1,239

Table 4. Feature Extraction Results*Skewness*

Image to-	Skewness Feature Extraction Results
1	-222,594
2	139,442
3	-537,794
4	152.886
5	-194.228
6	-108,334
7	-48,354
8	-663,774
9	-24.603
10	-168,583
...	...
291	138.896
292	117,710
293	93,826
294	94,058
295	621,487
296	123,295
297	98,344
298	98,978
299	24,956
300	83,423

Table 5. Energy Feature Extraction Results

Image to-	Energy Feature Extraction Results
1	235.506
2	26,377
3	603.539
4	591,899
5	131.535
6	141,302
7	250,763
8	138,228
9	350,562
10	215,405
...	...
291	119.939
292	128,946
293	123.789
294	117.382
295	128,300

296	449,237
297	499.993
298	292.153
299	132,618
300	276.517

Table 6. Entropy Feature Extraction Results

Image to-	Entropy Feature Extraction Results
1	3,364
2	121
3	15.625
4	15,129
5	57,600
6	65,062
7	3,721
8	1,444
9	6,400
10	2,916
...	...
293	51,984
294	47,534
295	55,225
296	9,604
297	11,449
298	4,761
299	9,025
300	4,356

Table 7. Results of *Smoothness*

Image to-	Smoothness Feature Extraction Results
1	0.589
2	0.398
3	0.42
4	0.641
5	0.551
6	0.471
7	0.58
8	0.367
9	0.563
10	0.118
...	...
291	0.515
292	0.568
293	0.523
294	0.540
295	0.201
296	0.221
297	0.571
298	0.605
299	0.559
300	0.606

System Implementation

This Covid-19 Early Detection application was created using the Netbeans 8.0.2 and JDK 1.8.0 applications. The application that has been completed is then tested on the entire testing data. The amount of data used as training data is 300 images, namely 150 Covid-19 images and 150 Non-Covid-19 images. While the testing data is 100, namely 50 Covid-19 images and 50 Non-Covid-19 images. In the appearance of this application itself, 3 sections are using the jTabPane feature in java netbeans, namely, the first jTabPane section to display training data as shown in Table 8, the second jTabPane section for the final classification process as shown in Picture 5 and Table 9, and the last jTabPane section. for the calculation of performance evaluation as in Table 10 and Table 11.

Table 8. Show Feature Extraction From Data Training

Image to-	Feature Extraction Results Average Intensity	Feature Extraction Results Standard Deviation	Feature Extraction Results *Skewness*	Feature Extraction Results Energy	Feature Extraction Results Entropy	Feature Extraction Results *Smoothness*
1	173.899	1.197	-222,594	235.506	3,364	0.589
2	69,052	0.814	139,442	26,377	121	0.398
3	213.869	0.850	-537,794	603.539	15.625	0.42
4	130.917	1.337	152.886	591,899	15,129	0.641
5	169,843	1,108	-194.228	131.535	57,600	0.551
6	185,265	0.944	-108,334	141,302	65,062	0.471
7	143.926	1.175	-48,354	250,763	3,721	0.58
8	210.78	0.761	-663,774	138,228	1,444	0.367
9	145.874	1,135	-24.603	350,562	6,400	0.563
10	34,009	0.366	-168,583	215,405	2,916	0.118
...	...	...	...	...	...	...
291	124,284	1.03	138.896	119.939	49,284	0.515
292	129,770	1,146	117,710	128,946	55.696	0.568
293	137.517	1.047	93,826	123.789	51,984	0.523
294	126.892	1.084	94,058	117.382	47,534	0.540
295	143.765	0.502	621,487	128,300	55,225	0.201
296	106.219	0.553	123,295	449,237	9,604	0.221
297	151,315	1,153	98,344	499.993	11,449	0.571
298	138,444	1,237	98,978	292.153	4,761	0.605
299	162,941	1.126	24,956	132,618	9,025	0.559
300	140,761	1,239	83,423	276.517	4,356	0.606

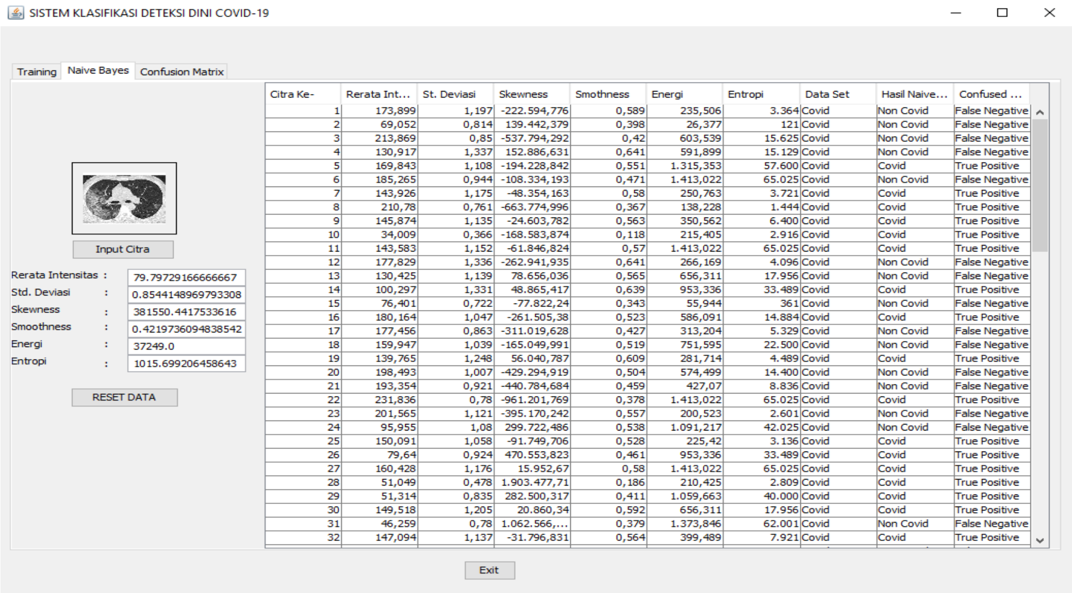

Figure 5. Final Classification

Table 9. Showing Results*Confusion Matrix*

Image to-	Feature Extraction Results Average Intensity	Feature Extraction Results Standard Deviation	Feature Extraction Results *Skewness*	Feature Extraction Results Energy	Feature Extraction Results Entropy	Feature Extraction Results *Smoothness*	DATA SET	NAIVE BAYES RESULTS	*CONF. MATRIX*
1	173.899	1.197	-222,594	235.506	3,364	0.589	*COVID*	*NON COVID*	*FALSE NEGATIVE*
2	69,052	0.814	139,442	26,377	121	0.398	*COVID*	*NON COVID*	*FALSE NEGATIVE*
3	213.869	0.850	-537,794	603.539	15.625	0.42	*COVID*	*NON COVID*	*FALSE NEGATIVE*
4	130.917	1.337	152.886	591,899	15,129	0.641	*COVID*	*NON COVID*	*FALSE NEGATIVE*
5	169,843	1,108	-194.228	131.535	57,600	0.551	*COVID*	*COVID*	*TRUE POSITIVE*
6	185,265	0.944	-108,334	141,302	65,062	0.471	*COVID*	*NON COVID*	*FALSE NEGATIVE*
7	143.926	1.175	-48,354	250,763	3,721	0.58	*COVID*	*COVID*	*TRUE POSITIVE*
8	210.78	0.761	-663,774	138,228	1,444	0.367	*COVID*	*COVID*	*TRUE POSITIVE*
9	145.874	1,135	-24.603	350,562	6,400	0.563	*COVID*	*COVID*	*TRUE POSITIVE*
10	34,009	0.366	-168,583	215,405	2,916	0.118	*COVID*	*COVID*	*TRUE POSITIVE*
...	...	...	...	...	...	...	...	...	...
91	197.183	0.393	-116,969	114.195	45,369	0.134	*NON COVID*	*COVID*	*FALSE POSITIVE*
92	77,314	0.752	931,500	140,648	64.516	0.361	*NON COVID*	*NON COVID*	*TRUE NEGATIVE*
93	66.66	0.382	197.032	134,780	60.025	0.127	*NON COVID*	*NON COVID*	*TRUE NEGATIVE*
94	75,314	0.819	439,746	102,823	38.025	0.402	*NON COVID*	*COVID*	*FALSE POSITIVE*
95	96.835	0.956	185.734	102,823	38.025	0.477	*NON COVID*	*COVID*	*FALSE POSITIVE*

96	183,336	0.919	-246,902	134,780	50.025	0.458	*NON COVID*	*COVID*	*FALSE POSITIVE*
97	138,781	1,140	107,498	383,074	7,396	0.565	*NON COVID*	*NON COVID*	*TRUE NEGATIV E*
98	196.784	0.853	-359,797	138.036	62,500	0.421	*NON COVID*	*NON COVID*	*TRUE NEGATIV E*
99	199,698	1.164	-397.534	156,982	1,764	0.575	*NON COVID*	*NON COVID*	*TRUE NEGATIV E*
100	79,797	0.854	381,550	101.569	37,249	0.422	*NON COVID*	*COVID*	*FALSE POSITIVE*

System Evaluation

In the evaluation of the system this time, data were obtained as in Table 10 and Table 11. Table 10 shows the Confusion Matrix Table and the results obtained are 19 positive data that are predicted to be correct, 15 negative data are predicted to be correct, 35 negative data are predicted to be positive and 31 positive data are predicted to be negative. From the results of the test data of 100 input images, it obtained an accuracy of 54%, a precision of 55.88%, a sensitivity of 38%, and a specificity of 70% which can be seen in Table 11.

Table 10. Recapitulation Confusion Matrix

Confusion Matrix Recapitulation	
True Positive	19
False Positive	15
True Negative	35
False Negative	31

Table 11. Performance evaluation

Performance Result Evaluation	
Accuracy	54%
Precision	55.88%
Sensitivity	38%
Specificity	70%

Discussion

Based on the system trial that has been made, in this study, the accuracy results were lower than in previous studies by obtaining an accuracy value of 54%, precision of 55.88%, sensitivity of 38%, and specificity of 70%. Previous studies obtained a high level of accuracy for example in research [6] with an average percentage of 73%. This study used 60 data divided into 3 classes, namely normal, cancer and effusion. The method used is histogram feature extraction and naive Bayes classifier method. Another study that has a high accuracy is [7] with the highest accuracy being 60%. This study used 80 data which were divided into 4 types of soil, namely litosol soil, alluvial soil and podzolic soil, and humus soil. The first step is preprocessing by resizing the image file to a size of 416 x 312 pixels. The method used is histogram feature extraction and k-nearest neighbor. Research [8] uses a backpropagation artificial neural network for face recognition using a histogram-based feature extraction method. The facial image data uses variations in pixel sizes of 640 x 480 pixels and 600 x 800 pixels. The results obtained from the ANN test using 18 face images were 95%. Research [9] contains the classification of the maturity level of manalagi apples using the Naive Bayes algorithm and digital image feature extraction. The data used are 130 images, consisting of 100 training images and 30 test images with an accuracy of 63%. Research [16] uses data in the form of X-Ray images and the Deep Residual Network method for the detection of Covid-19. The X-Ray image test data used were 42 Covid-19 class images and 234 normal class images with an accuracy rate of 99%. Research [17] used data in the form of CT-Scan images and used Multi-Class texture analysis for Covid-19 detection. The data used are CT scans of the chest used in this design, including CT scans of infected Covid-19 patients, CT scans of infected pneumonia patients, and CT scans of healthy people without detectable chest infections collected from Radiopaedia from The Cancer Imaging Archive website with high accuracy. obtained in this study reached 87.4%. Based on the previous research that has been mentioned previously, this study has a lower percentage of accuracy than previous studies. This is possible because the pixel sizes are not the same in the preprocessing stage. The preprocessing stage in question is feature extraction, namely the process of getting the mean, standard deviation, entropy, energy, skewness, and also smoothness. Accuracy maybe even better if the images in this study are equated. It is hoped that further research will be able to develop this research so that the accuracy of the results obtained is better.

IV. CONCLUSION

This research is to use the feature extraction method and Naive Bayes classifier algorithm to get 54% accuracy, 55.88% precision, 38% sensitivity and 70% specificity. The author realizes that in making this system there are still some shortcomings, so the authors hope that in future research they can develop it by involving the feature extraction process and using methods with better accuracy in image processing to obtain even better results.

REFERENCES

[1] A. Ciputra, A. Susanto, and et al, "Classification of Manalagi Apples with Naive Bayes Algorithm and Digital Image Feature Extraction,"Symmetrical J. Tek. Mechanical, Electrical and Computer Science., vol. 9, no. 1, pp. 465–472, 2018.

[2] I. Ozsahin, B. Sekeroglu, MS Musa, MT Mustapha, and DU Ozsahin, “Review on Diagnosis of COVID-19 from Chest CT Images Using Artificial Intelligence,”Comput.

Maths. Methods Med., vol. 2020, no. Volume 2020, Article ID 9756518, pp. 1–10, 2020, doi: https://doi.org/10.1155/2020/9756518 Research.

[3] T. Elizabeth, “INCREASING THE QUALITY OF LUNG X-RAY IMAGES OF COVID-19 PATIENTS USING THE CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION METHOD,” vol. 4, no. 2, pp. 345–349, 2020.

[4] A. Ultimate, “Image Feature Extraction.” https://pemrogramanmatlab.com/pengolahan-citra-digital/ekstrak-harga-citra-digital/#:~:text=Image feature extraction is a step, the others are in the identification stage,%2F classification.

[5] F. Syarifuddin, M. Misdram, AA Widodo, PS Informatics, and UM Pasuruan, “CLASSIFICATION OF CORONA VIRUS DATA SET USING,” vol. 12, no. 2, pp. 46–52, 2020.

[6] A. Ramdhan, "Classification of Lung X-Rays Using Histogram Feature Extraction and Naive Bayes Classifier Method," 2015.

[7] R. Rudi and D. Avianto, "Implementation of Histogram and K-Nearest Neighbor Feature Extraction for Soil Type Classification in Banjar City, West Java,"J. Buana Inform., vol. 10, no. 2, p. 85, 2019, doi:10.24002/jbi.v10i2.2141.

[8] S. Kusmaryanto, "Backpropagation Neural Network for Facial Recognition Histogram-Based Feature Extraction Method,"J. EECCIS Vol. 8, No. 2, December 2014, vol. 8, no. 2, pp. 193–198, 2014.

[9] H. Bisri, MA Bustomi, and E. Purwanti, “Classification of Lung Image by Feature ExtractionHistograms and Backpropagation Neural Networks,” J. Science And Art Pomits, vol. 2, no. 2, pp. 68–71, 2013.

[10] NI Widiastuti, E. Rainarli, and KE Dewi, "Summary and Support Vector Machine on Document Classification,"J. Infotel, vol. 9, no. 4, p. 416, 2017, doi:10.20895/infotel.v9i4.312.

[11] MI Fajri and L. Anifah, "Detection of Lung Cancer Status on Ct Scan Image Using Fuzzy Logic Method,"Tech. Electro, vol. 7 no. 3, pp. 121–126, 2018.

[12] MS Wibawa, "Comparative Study of Lung Segmentation Methods on Axial Ct-Scan Images,"J. Nas. educator. Tech. information., vol. 7, no. 3, p. 283, 2019, doi:10.23887/janapati.v7i3.15751.

[13] A. Susiloet al., “Coronavirus Disease 2019: Recent Literature Review,” vol. 7, no. 1, pp. 45–67, 2020.

[14] EG Samudro and MA Madjid, "The Government of Indonesia is Facing the Covid-19 National Disaster That Threatens National Security,"J. Resilience Nas., vol. 26, no. 2, p. 132, 2020, doi:10.22146/jkn.56318.

[15] MH Widianto, "Naive Bayes Algorithm," 2019. https://binus.ac.id/bandung/2019/12/algoritma-naive-bayes/#:~:text=Naïve Bayes Classifier is a classification method rooted in theorem Bayes .&text=This algorithm assumes that the attribute,dr " master " is a decision table.

[16] YSUN Hariyani, S. Hadiyoso, and TS Siadari, "Covid-19 Disease Detection Based on X-Ray Imagery Using Deep Residual Networks," vol. 8, no. 2, pp. 443–453, 2020.

[17] AM Yasin, PS Informatics, U. Pembangunan, N. Veteran, and J. Timur, "Design for Covid-19 Disease Detection Using Multi-Class Texture Analysis on Chest CT Scan," pp. 1–4, 2020.