



Implementation of KNN algorithm in classifying diabetic ulcers in patients with diabetes mellitus

Saut Dohot Siregar¹, Yusra Uli Rasita Ginting², Nita Sintami³, Hari Salim Butar-Butar⁴, Riki Marthin Simanjuntak⁵

^{1,2,3,4,5}Prima Indonesia University / Faculty of Science and Technology / Informatics Engineering, Medan, Indonesia

ARTICLE INFO

Article history:

Accepted, Jul 09, 2023

Revised Jul 12, 2023

Accepted Jul 18, 2023

Keywords:

Classification;
Data Science;
Diabetes Mellitus;
Diabetic Ulcer;
K-Nearest Neighbour.

ABSTRACT

The rapid development of the times, humans face various types of diseases that exist, even some of these types of diseases can be fatal and even death, one of the most famous diseases is diabetes mellitus. Diabetic ulcers are caused by diabetes mellitus. The risk of people with diabetes mellitus is 29 times that there will be diabetic ulcer complications, and the potential to experience diabetic ulcers is 15-25% during their lifetime and within 5 years has a recurrence rate of 50-70%. This research uses the K-Nearest Neighbor algorithm with the Gray Level Co-Occurrence model which is the extraction of image or texture features from images. Using the `calculate_glcms(image)` function to calculate some features of the GLCM matrix given an image as input. GLCM: `greycomatrix(image,[1], [0, np.pi/4, np.pi/2, 3 np.pi/4], levels=256, symmetric=True, normed=True)` calculates the GLCM matrix of the image with pixel distance 1 and angle 0, $\pi/4$, $\pi/2$, and $3\pi/4$. Conclusion of this research is the test result matrix, the accuracy is 0.94 (94%), precision 0.928, recall 0.905, F-1 Score 0.907 and support of 66.

This is an open access article under the [CC BY-NC](https://creativecommons.org/licenses/by-nc/4.0/) license.



Corresponding Author:

Saut Dohot Siregar,
Faculty of Science and Technology / Informatics Engineering,
Prima Indonesia University,
Jl. Belanga No.1, Medan, Sumatera Utara, 20113, Indonesia.
Email: Saut.unpri@gmail.com

1. INTRODUCTION

The rapid development of the era, humans face various types of diseases that exist, even some of these types of diseases can be fatal and even death, one of the most famous diseases is diabetes mellitus. According to Dr Damar Upahita, diabetes mellitus is among the deadliest diseases in Indonesia. Diabetes mellitus is a serious disease, diabetes mellitus can even lead to several disease complications including cardiovascular disease, kidney disorders, inflammation, and obesity (Liantoni, 2016). According to the explanation in the World Health Organization (WHO) data, there are 171 million diabetics in the world in 2000 at this time and will experience an increase of 2 times, by 2030 there will be 336 million sufferers (Pawening et al., 2020). Diabetes mellitus can be characterised by a condition of chronic hyperglycaemia caused by abnormalities in insulin secretion. In addition, diabetes mellitus is a disease that cannot be transmitted to

other people (Syarifah et al., 2022). Diabetic ulcers are caused by diabetes mellitus. People with diabetes mellitus have a 29 times risk of diabetic ulcer complications, and the potential to experience diabetic ulcers is 15-25% during their lifetime and within 5 years has a recurrence rate of 50-70% (Laia et al., 2021). Diabetic ulcers can be characterised by chronic wounds. This is a disease that occurs below the ankle and is caused by the process of peripheral neuropathy, peripheral arterial disease that increases morbidity, mortality and reduces the quality of life of patients (Mardisa et al., 2022). In the process of detecting diabetic ulcers, object detection will be carried out in the image of wounds on the soles of the feet through image processing which will recognise diabetic ulcers in patients with diabetes mellitus. In this research, K-Nearest Neighbor (KNN) is the algorithm that will be used in image processing objects. The application of KNN which is a methodology for grouping certain objects and then considering the closest distance from the object (Kurnaiwan, Muhamand Reinaldi Purnamasari & Fuadah, 2021). The purpose of the KNN algorithm is to classify objects based on sample attributes and training, where the results of the tested samples are reclassified mostly in the KNN category (Ikhsan et al., 2020).

In previous research entitled "Classification Of Diabetic Foot Ulcer Using Convolutional Neural Network (CNN) In Diabetic Patient" obtained an accuracy result of 99% using the CNN algorithm. The study conducted several validation training and loss validation so that the stage was carried out six times epoch to get maximum results. Therefore, this research tries to use the KNN algorithm with the Gray Level Co-Occurrence Matrix (GLCM) model which is an extraction of image or texture features from images. By using GLCM, it will be able to measure the level of roughness, uniformity and texture patterns in the image. GLCM will also improve the feature representation and enrich the information used to classify the data. The GLCM stage only requires one time, and is simpler and easier to implement. The advantage of this research is that it is faster because the training is not much and saves storage. Based on the description that has been presented, a research topic was raised with the title "Implementation of the KNN Algorithm in Classifying Diabetic Ulcers in Patients with Diabetes Mellitus".

Of & Of, n.d. (2021) conducted a study entitled SVM vs KNN For Classification Of Histopathological Images Of Varicose Ulcer getting the results that tests conducted with the KNN algorithm obtained an accuracy rate of 80.95%. In this study, using several methods starting with feature extraction. The feature extraction includes texture, homogeneity, energy, entropy, contrast, average, similarity, and variation features. Then the next method is classification in the form of a KNN classifier and SVM classifier. The last method is the experimental results that finally the difference in ulcer stage is classified using KNN classifier and SVM classifier.

Huang et al., (2021) conducted a study entitled Accurate Diagnosis Of Endoscopic Mucosal Healing In Ulcerative Colitis Using Machine learning And Machine Learning getting the results that tests conducted with the KNN algorithm obtained an accuracy rate of 93.8%. In this study, using several methods starting with resources which means searching for datasets. Then the next method is database and image separation and preprocessing. After that, the next method is image classification through DLML-CAD concept DLML-. The last method is training and flow calculation in order to get significant results.

Broti et al., (2022) conducted a study entitled Analysis Toward Detection System Of Skin Diseases (Impetigo, Melanoma, Diabetic Foot Ulcer, And Seborrhic Dermatitis) getting the results that tests conducted with the KNN algorithm obtained an accuracy rate of 93.37%. In this study, there are four methods which include image pre-processing, segmentation, feature extraction and classification tasks. The images constituting the dataset were collected from various medical websites then the images were resized into two parts, 67% of the total images have been used for training and 33% for testing. Another 1250 sets of images have been used for validation.

Khorasani et al., (2020) conducted a study entitled Detecting Ulcerative Colitis From Colon Samples Using Efficient Feature Selection And Machine Learning to get the results that tests conducted with the KNN algorithm obtained an accuracy rate of 92.31%. In this study, the method used first is data pre-processing which includes steps such as calculating the expression value per gene by taking the average expression value of all probes mapped to the same gene, handling missing values with the KNN imputation method from the "missingpy" library in Python27. Then the next method is model generation which aims to create a model to distinguish using dimensionality reduction through the perturbation theory feature selection method. The last method is bio disc ml which is a bio marker discovery software that uses machine learning methods to analyse biological data sets.

Reddy et al., (2021) conducted a study entitled Exploiting Machine Learning Algorithms To Diagnose Foot Ulcers In Diabetic Patients getting the results that the test conducted with the KNN algorithm obtained an accuracy rate of 93.16%. In this study, the method used is data pre-processing performed on the loaded data set. Then the percentage division of 80% is used for training and 20% for testing with 107 and 26 respectively. Then three existing algorithms and one proposed algorithm are implemented in programming individually.

Harahap et al., (2022) conducted a study entitled Classification Of Diabetic Foot Ulcer Using Convolutional Neural Network (CNN) In Diabetic Patients getting the results that tests conducted with the CNN algorithm obtained an accuracy rate of 99%. In this research, the method used is pre-processing the image test data by rotating, adding contrast and removing noise in the image. Image test data with testing stages carried out on the dataset collection is grouped and divided into 547 training image sets, 171 testing sets and 137 validation sets.

Based on previous research studies that have been described, this research uses the KNN algorithm with the GLCM model to get better accuracy and a shorter process. By classifying 1040 datasets that will distinguish into normal and abnormal classes. Therefore, how does it differ from the current research is the model used. It is hoped that the model used is easier, does not take more time and get better results.

2. RESEARCH METHOD

In this study using the type of experimental research where researchers implemented the KNN algorithm in classifying diabetic ulcers in patients with diabetes mellitus. This research uses deep learning methods in processing images that will later be used as research objects. Deep learning will process images that will be classified by KNN to determine whether the object image is a normal or abnormal wound (diabetic ulcer). This research carried out several stages of work procedures. The work procedures carried out are as follows:

a. Dataset

Datasets are collections of data that have been organised and are generally patent for each input. In this study, a dataset is needed in the form of several images of objects to be studied, namely the soles of the feet of people with diabetic ulcers, the source of the dataset can be obtained from https://www.researchgate.net/publication/353818646_Diabetic_Foot_Ulcer_DFU_datase t which is the object of this research.



Figure 1. Example of Diabetes Mellitus Wound Object on the sole of the foot

b. Creating Base Dir

Base dir is a variable that indicates the path or base directory for the diabetes dataset. This stage combines the previously calculated data with the help of the number_label dictionary and produces visualisations in the form of bar charts with appropriate titles and labels.

c. Object Classification

In this research, objects with diabetic ulcers are classified using the KNN algorithm.

d. Results

The final stage where the result of the wound object detection shows the classification of the wound type where the object will be classified as a diabetic ulcer or not.

3. RESULTS AND DISCUSSIONS

Entering the results stage, where the research is almost complete but needs to perform some tests to ensure the programme can function properly. Testing the programme to ensure any errors that would later become potential defects are detected, the researcher will immediately overcome quickly and do the final completion of the programme. This research obtained results from the data processing, testing and discussion of the proposed model to classify diabetic ulcers in patients with diabetes mellitus.

3.1. Creating a base dir

Base dir is a variable that indicates the path or base directory for the diabetes dataset. The value "content" at the beginning of the path indicates that the dataset is located in the main directory or the current context directory in the environment used, such as Google Colab or Jupyter Notebook. Then, the "DATASET" directory is a subdirectory within the main directory. The directory indicates that the diabetes dataset is located within the "DATASET" sub-directory. The last step of creating the base dir is to create the "diabetes" directory which is the second sub-directory located within the "DATASET" sub-directory. This indicates that the specific diabetes dataset is located within the "diabetes" sub-directory. The dataset was obtained from the website https://www.researchgate.net/publications/353818646_Diabetic_Foot_Ulcer_DFU_datas et which can be obtained online.

The code is used to count the number of files in each sub-directory in base_label. In addition, the code also counts the total number of files in all sub-directories and prints the result. It then defines an empty dictionary number_label = {}. This dictionary will be used for stores the number of files in each sub-directory. The variable total_files is initialised with a value of 0. It will be used to calculate the grand total of files in each subdirectory. The last step is to print the message "Total Files:" followed by the current

total_files in each iteration of the loop. Outside the loop, after it has finished counting the number of files in all sub-directories, it then prints the message "number of all files present in the dataset:" followed by the value of total-files.

3.2. Visualising Images and Labels

The code provided uses matplotlib to create a bar chart showing the number of images for each label in the dataset. This stage combines the pre-calculated data with the help of the number_label dictionary and generates a visualisation in the form of a bar chart with appropriate titles and labels. Then it is necessary to import the matplotlib module first by using `import matplotlib.pyplot as plt` before running the programme. It has certainly executed the previous code that counts the number of files in each subdirectory and stores it in the dictionary number_label. The result of the number of images per label can be seen in Figure 2.

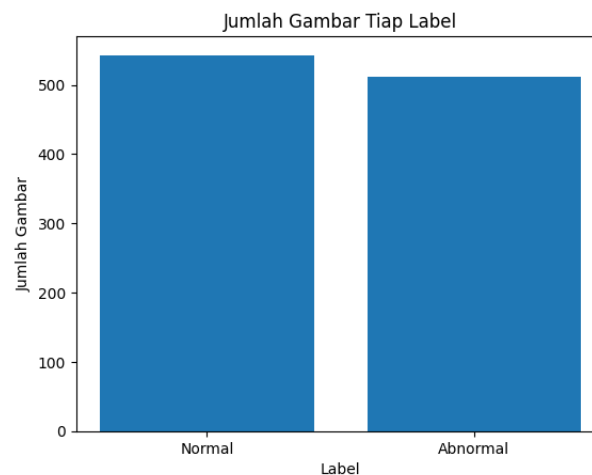


Figure 2. Number of Images per Table

3.3. Object Classification

This stage of the research is used to display sample images from each class in the dataset. The explanation of this stage is in the form of: (a) `Img_each_class = 1` is a variable that determines the number of sample images to be displayed for each class. This instance is set to 1, which means one image from each class will be displayed. (b) `Img_samples = {}` is a dictionary that will be used to store the garmar of each class (c) `Classes = list(number_label.keys())` retrieves the list of classes from the previously created dictionary number_label

Inside the loop, it will iterate through each class, get a list of file names from each class, and then limit the number of files retrieved to `img_each_class`. Inside the nested loop, it iterates through the list of file names obtained in the previous step. At each iteration, it constructs a complete image path with and stores the path in the `img_samples` dictionary with class as the key. After collecting sample images from each class, it proceeds with a loop to display each image. Thus, this code will display sample images one by one from each class in the dataset, with the title corresponding to the class. After running the previous code that counts the number of files in each sub-directory and stores it in the dictionary number_label. In addition, it imported the `matloplib.pyplot` and `matplotlib.image` modules. Then count how many images in the dataset directory, and obtained a total of 512 normal dataset images and 543 abnormal datasets, an example of a normal sample image can be seen in Figure 3. and examples of abnormal sample images can be seen in Figure 4.

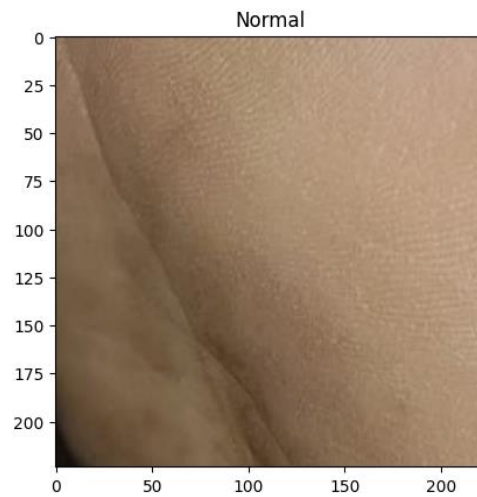


Figure 3. Picture of Normal Class Sample

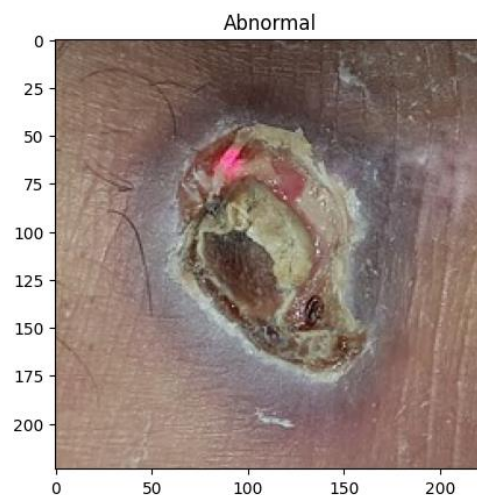


Figure 4. Picture of Abnormal Class Sample

3.4. Creating Function for KNN Classification

This stage will import some modules for KNN first. Then it uses the `calculate_glcms(image)` function to calculate some features of the Gray Level CoOccurrence matrix (GLCM) given the image as input. GLCM: `greycomatrix(image,[1], [0, np.pi/4, np.pi/2, 3 np.pi/4], levels=256, symmetric=True, normed=True)` calculates the GLCM matrix of the image with pixel distance 1 and angle 0, $\text{np.pi}/4$, $\text{np.pi}/2$, and $3\text{np.pi}/4$. Then, the GLCM matrix is calculated using the `greycomatrix()` function of `skimage.feature`. KNN will classify an image based on the closest distance to other objects or neighbours, KNN illustration can be seen in Figure 3.4.

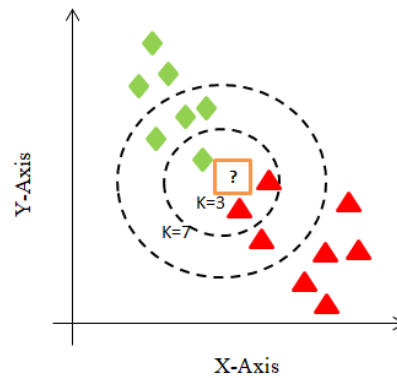


Figure 5. KNN Illustration

There are several functions used to calculate some properties of GLCM: (a) Contrast = `greycoprops(glcm,'energy').ravel()` calculates the energy of the GLCM matrix using the `greycoprops()` function with the argument 'contrast'. `ravel()` is used to convert the result into a 1D array. (b) Dissimilarity = `greycoprops(glcm, 'dissimilarity').ravel()` calculates the dissimilarity of the GLCM matrix using the `greycoprops()` function with the argument 'dissimilarity'. (c) Homogeneity = `greycoprops(glcm, 'homogeneity').ravel()` calculates the homogeneity of the GLCM matrix using the `greycoprops()` function with the 'homogeneity' argument. (d) Energy = `greycoprops(glcm, 'energy').ravel()` calculates the energy of the GLCM matrix using the `greycoprops()` function with the argument 'energy'. (e) Correlation = `greycoprops(glcm, 'correlation').ravel()` calculates the correlation of the GLCM matrix using the `greycoprops()` function with the argument 'correlation'. (f) Features = `np.hstack([contrast, dissimilarity, homogeneity, energy, correlation])` combines all the GLCM features into one feature vector using the `np.hstack()` function. Each of the previously calculated GLCM features (contrast, dissimilarity, homogeneity, energy, and correlation) will be the elements in the feature vector.

`Return features` serves to return the calculated feature vector. In addition, using the `calculate_glcms()` function, the programme calculates the GLCM features of an image and uses them for further analysis, such as classification or pattern detection in the image. This function is used after importing the necessary modules, such as `skimage.feature` and `numpy`. Then the images from the two different categories of normal and abnormal are loaded as grayscale images using OpenCV's `cv2.imread()` function. After that, there will be two empty lists that will be used to store the images from the "Normal" and "Abnormal" categories respectively.

Through the loop for `i` in `range(1,101)` the programme will iterate from 1 to 100 according to the number of images in each category. Furthermore, the images will be modelled with a grey scale and the images will also be loaded in the form of a numpy array. The loaded images are then added to the `normal_images` or `abnormal_images` list according to the image category. After finishing loading all images from the "Normal" and "Abnormal" categories, the `normal_images` and `abnormal_images` lists will be returned as the output of the function.

Implementation is done to extract GLCM features from normal and abnormal diabetes images, and store the features in a list of features and labels in a list of labels. Features and labels are two empty lists that will be used to store GLCM features and labels respectively.

`X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_state=42)` splits the dataset into training and test data. `features` is the list of features to be split, `labels` is the list of labels to be split, `test_size=0.2` determines the proportion of test data (20% of the total dataset), and `random_state=42` is used to set the

seed so that the split can be reproduced with the same result each time the code is run. `return X_train, X_test, y_train, y_test` returns four variables containing the split training and test data. By using the `split_dataset()` function, the program can split the feature and label dataset into training and test data. The variable `X_train` will contain training data features, `X_test` will contain test data features, `y_train` will contain training data labels, and `y_test` will contain test data labels. The programme can also use this data to train the model and test its performance.

This research uses a function to perform classification using the KNN algorithm. This function uses the `KNeighborsClassifier` from the `sklearn.neighbors` module to perform classification. Each part of the function is `knn = KNeighborsClassifier(n_neighbors=3)` initialises the `KNeighborsClassifier` object with the parameter `n_neighbors=3`, which specifies the number of neighbours to be used in the classification process. It then changes the value of `n_neighbors` as needed. By using the `knn_classification()` function, researchers can train the KNN model using `X_train` and label-train `y_train` data, and predict and calculate accuracy on `X_test` and label-test `y_test` data. Researchers can also adjust the value of `n_neighbors` and conduct further analysis of the prediction results and model accuracy. The `plot_confusion_matrix()` function is used to evaluate the model prediction results. Then, the `classification_report` function takes two arguments, namely `y_test` which contains the original labels of the test data and `y_pred` which contains the prediction results of the model. The classification report will include various evaluation metrics such as precision, recall, f1-score, and support for each class. From the test result matrix, the accuracy is 0.94 (94%), precision 0.928, recall 0.905, F-1 Score 0.907 and support 66. The results of the confusion matrix plot can be seen in Figure 6.

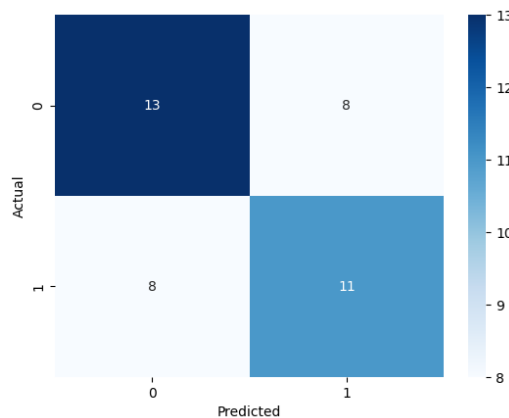


Figure 6. Confusion Matrix Plot

Based on the research of Mawaddah Harahap (2022) [16], entitled classification of diabetic foot ulcer using convolutional neural network (CNN) in diabetic patients, the accuracy result is 99% using CNN algorithm. It can be concluded that the process in KNN uses fewer models and does not require more models, namely GLCM. In CNN, it is necessary to train as many as 6 models consisting of VGG19, Resnet50V2, Resnet101V2, Resnet 152V2, InceptionRestNetV2, and MobileNetV2 to get maximum results and high accuracy. Based on the results of the research that has been undertaken, the following are things that are concluded by researchers in writing research reports are obtaining an accuracy value of 0.94 (94%), precision 0.928, recall 0.905, F-1 Score 0.907 and support of 66. Can be seen in Table 1 for comparison of results from previous titles.

Table 1. Comparison of Results

Algorithm	Accuracy	Precision	Recall	F-1 Score	Support
KNN	0.94	0.928	0.905	0.907	66
CNN	0.99	1.00	0.986	0.993	72

After obtaining the accuracy results of this study, a comparison was made with 5 previous journals that used the KNN algorithm that classified ulcers. The following accuracy comparison can be seen in Table.

Table 2. Comparison of KNN Accuracy Results

Research Title	Accuracy □ Percentage
SVM Vs KNN For Classification Of Histopathological Images □ Of Varicose Ulcer	80.95%
Accurate Diagnosis Of Endoscopic Mucosal Healing In □ Ulcerative Colitis Using Machine Learning And Machine □ Learning	93.8%
Analysis Toward Detection System Of Skin Diseases □ (Impetigo, Melanoma, Diabetic Foot Ulcer, And Seborrhic □ Dermatitis)	93.37%
Detecting Ulcerative Colitis From Colon Samples Using □ Efficient Feature Selection And Machine Learning	92.21%
Exploiting Machine Learning Algorithms To Diganose Foot □ Ulcer In Diabetic Patients	93.16%
Implementation of the KNN Algorithm in Classifying □ Diabetic Ulcers in Patients with Diabetes Mellitus	94%

4. CONCLUSION

This research implements the KNN algorithm in classifying diabetic ulcers in patients with diabetes mellitus which aims to help medical personnel be more modern and better in utilising technology. Based on the results of the research that has been undertaken, the following are the things concluded by the researcher in writing the research report is the results of the research conducted obtained an accuracy value of 0.94 (94%), precision 0.928, recall 0.905, F-1 Score 0.907 and support of 66. Then, the obtaining a comparison between CNN and KNN in terms of methods and accuracy. The implementation of the KNN algorithm in classifying diabetic ulcers is expected to increase the ease of delivering information and save time and effort in medical science and it is expected to facilitate and provide efficiency and acceleration of performance due to the application of technology in the medical world. The limitation of this research is that the dataset is not much, so it is hoped that further research will use larger data and with other algorithms. In addition, this research only uses one variable, namely diabetic ulcers. so, it is hoped that further research will use more variables.

REFERENCES

- Broti, T., Siddika, A., Hossain, N., Rituparna, S., & Sakib, N. (2022). *Analysis Toward Detection System of Skin Diseases (Impetigo , Melanoma , Diabetic Foot Ulcer , and Seborrhic Dermatitis)*. August.
- Fauzi, A. A., Utamingrum, F., & Ramdani, F. (2020). Road surface classification based on LBP and GLCM features using KNN classifier. *Bulletin of Electrical Engineering and Informatics*, 9(4), 1446–1453. <https://doi.org/10.11591/eei.v9i4.2348>
- Haesri, B., Fitriyah, H., Hannats, M., & Ichsan, H. (2022). *Klasifikasi Kesiapan Panen Tanaman Hidroponik Bayam Hijau menggunakan Metode Pengolahan Citra dan K-Nearest Neighbours berbasis Raspberry Pi*. 6(12), 5931–5939.
- Harahap, M., Anjelli, S. K., Sinaga, W. A. M., Alward, R., Manawan, J. F. W., & Husein, A. M. (2022). Classification of diabetic foot ulcer using convolutional neural network (CNN) in diabetic patients. *Jurnal Infotel*, 14(3), 196–202. <https://doi.org/10.20895/infotel.v14i3.796>
- Huang, T. Y., Zhan, S. Q., Chen, P. J., Yang, C. W., & Lu, H. H. S. (2021). Accurate diagnosis of endoscopic mucosal healing in ulcerative colitis using deep learning and machine learning.

- Journal of the Chinese Medical Association*, 84(7), 678–681.
<https://doi.org/10.1097/JCMA.0000000000000559>
- Ikhsan, D., Utami, E., & Wibowo, F. W. (2020). Metode Klasifikasi Mutu Greenbean Kopi Arabika Lanang Dan Biasa Menggunakan K-Nearest Neighbor Berdasarkan Bentuk. *Jurnal Ilmiah SINUS*, 18(2), 1. <https://doi.org/10.30646/sinus.v18i2.456>
- Khorasani, H. M., Usefi, H., & Peña-Castillo, L. (2020). Detecting ulcerative colitis from colon samples using efficient feature selection and machine learning. *Scientific Reports*, 10(1), 1–9. <https://doi.org/10.1038/s41598-020-70583-0>
- Kurnaiwan, Muhamand Reinaldi Purnamasari, R., & Fuadah, R. Y. N. (2021). Sistem Klasifikasi Sel Darah Putih Berbasis Pengolahan Citra Digital dan Jaringan Saraf Tiruan White Blood Cell Classification System Based on Digital Image Processing and Neural Network. *E-Proceeding of Engineering*, 8(1), 1–6.
- Laia, M., Hondro, R. K., & Zebua, T. (2021). Implementasi Pengolahan Citra dengan Menggunakan Metode K-Nearest Neighbor Untuk Mengetahui Daging Ayam Busuk dan Daging Ayam Segar. 8(2), 39–49. <https://doi.org/10.30865/jurikom.v8i2.2818>
- Larijani, M. R., Asli-Ardeh, E. A., Kozegar, E., & Loni, R. (2019). Evaluation of image processing technique in identifying rice blast disease in field conditions based on KNN algorithm improvement by K-means. *Food Science and Nutrition*, 7(12), 3922–3930. <https://doi.org/10.1002/fsn3.1251>
- Liantoni, F. (2016). Klasifikasi Daun Dengan Perbaikan Fitur Citra Menggunakan Metode K-Nearest Neighbor. *Jurnal ULTIMATICS*, 7(2), 98–104. <https://doi.org/10.31937/ti.v7i2.356>
- Mardisa, R., Nasution, I. S., & Siregar, K. (2022). Klasifikasi Kualitas Fisik Kopi Beras Arabika Menggunakan Pengolahan Citra Dengan Metode K-Nearest Neighbor (K-NN). *Jurnal Ilmiah Mahasiswa Pertanian*, 7(2), 514–522. <http://www.jim.unsyiah.ac.id/JFP/article/view/19896%0Ahttp://www.jim.unsyiah.ac.id/JFP/article/download/19896/9640>
- Masker, D., & Wajah, P. (2022). *Principal Component Analysis ... (Krismawan & Rachmawanto)*. 12(1), 382–389.
- Muhammad, D. I., Ermatita, E., & Falih, N. (2021). Penggunaan K-Nearest Neighbor (KNN) untuk Mengklasifikasi Citra Belimbing Berdasarkan Fitur Warna. *Informatik: Jurnal Ilmu Komputer*, 17(1), 9. <https://doi.org/10.52958/iftk.v17i1.2132>
- Of, L., & Of, M. (n.d.). SVM V S KNN FOR C LASSIFICATION OF H ISTOPATHOLOGICAL I MAGES. 3(2), 19–28.
- Pawening, R. E., Shudiq, W. J., & Wahyuni, W. (2020). KLASIFIKASI KUALITAS JERUK LOKAL BERDASARKAN TEKSTUR DAN BENTUK MENGGUNAKAN METODE k-NEAREST NEIGHBOR (k-NN). *COREAI: Jurnal Kecerdasan Buatan, Komputasi Dan Teknologi Informasi*, 1(1), 10–17. <https://doi.org/10.33650/coreai.v1i1.1640>
- Puspita, R., & Widodo, A. (2021). Perbandingan Metode KNN, Decision Tree, dan Naïve Bayes Terhadap Analisis Sentimen Pengguna Layanan BPJS. *Jurnal Informatika Universitas Pamulang*, 5(4), 646. <https://doi.org/10.32493/informatika.v5i4.7622>
- Reddy, S. S., Mahesh, G., & Preethi, N. M. (2021). Exploiting Machine Learning Algorithms to Diagnose Foot Ulcers in Diabetic Patients. *EAI Endorsed Transactions on Pervasive Health and Technology*, 7(29), 1–14. <https://doi.org/10.4108/eai.24-8-2021.170752>
- Saputro, W., Sumantri, D. B., Tinggi, S., Komputer, I., & Karya Informatika, C. (2022). Implementasi Citra Digital Dalam Klasifikasi Jenis Buah Anggur Dengan Algoritma K-Nearest Neighbors (Knn) Dan Data Augmentasi Digital Image Implementation in Grape Classification With K-Nearest Neighbors (Knn) Algorithm and Augmentation Data. *Journal of Information Technology and Computer Science (INTECOMS)*, 5(2), 248–253.
- Sarimole, F. M., & Syaeful, A. (2022). Classification of Durian Types Using Features Extraction Gray Level Co-Occurrence Matrix (GlcM) and K-Nearest Neighbors (Knn). *Journal of Applied Engineering and Technological Science*, 4(1), 111–121. <https://doi.org/10.37385/jaets.v4i1.959>
- Shakya, D. S. (2020). Analysis of Artificial Intelligence based Image Classification Techniques. *Journal of Innovative Image Processing*, 2(1), 44–54. <https://doi.org/10.36548/jiip.2020.1.005>
- Shidiq, F. (2021). Penerapan Metode K-Nearest Neighbor (KNN) Untuk Menentukan Ikan Cupang Dengan Ekstraksi Fitur Ciri Bentuk Dan Canny. *Innovation in Research of Informatics (INNOVATICS)*, 3(2), 39–46. <https://doi.org/10.37058/innovatics.v3i2.3093>
- Subhannur Rahman, Yumi Baida Rahmah. (2020). Literature Review: Kebutuhan Spiritual Pada

- Pasien Dengan Ulkus Diabetik. *Nursing Science Journal (NSJ)*, 1(2), 6–11. <https://doi.org/10.53510/nsj.v1i2.25>
- Syarifah, A., Riadi, A. A., & Susanto, A. (2022). 417-1274-1-Pb. 7(1), 27–35.
- Utia Detty, A., Fitriyani, N., Prasetya, T., & Florentina, B. (2020). Karakteristik Ulkus Diabetikum Pada Penderita Diabetes Melitus The Characteristics of Diabetic Ulcer in Patients with Diabetes Mellitus. *Juni*, 11(1), 258–264. <https://doi.org/10.35816/jiskh.v10i2.261>