Journal of Artificial Intelligence and Engineering Applications

Web sites: https://ioinformatic.org/

October 15, 2025. Vol. Part 5. Not. 1; e-ISSN: 2808-4519

Detection of Surra Disease in Microscopic Images of Horse Blood Using the K-Nearest Neighbor (KNN) Method

Indra Saputra Landujawa*[1], Fajar Hariadi2, Raynesta Mikaela Indri Malo [3]

[1,2,3] Sumba Informatics Engineering Study Program Wira Wacana Christian University
indrasaputralandujawa@gmail.com[1*]

Abstract

Surra's disease is an infectious disease caused by the parasite Trypanosoma evansi and attacks farm animals such as horses, cattle, and buffaloes. This disease can cause serious health problems and even death, resulting in significant economic losses for farmers. This study aims to develop an automatic detection system for Surra disease in horses using the K-Nearest Neighbor (KNN) classification method based on blood microscopic images. The research process includes image preprocessing stages (conversion to grayscale and resize), extraction of texture features using the Gray Level Co-occurrence Matrix (GLCM) method, and classification using the KNN algorithm. The extracted features include contrast, homogeneity, energy, and correlation. Model evaluation was carried out using a confusion matrix and evaluation metrics such as accuracy, precision, recall, and f1-score. Based on the results of the evaluation, the KNN model with a value of k = 1 was able to achieve an accuracy of 85%, with a high recall in the Infected class of 0.95, and an average f1-score of 0.85. These results show that GLCM and KNN methods are effectively used in detecting Surra's disease on blood images, especially in recognizing visual patterns that indicate infection. The developed system is expected to be a fast, efficient, and practical early detection solution in the field, especially in livestock areas such as East Sumba.

***Keywords**: K-Nearest Neighbor, GLCM, Surra Disease Detection, Blood Microscopic Imaging, Horses*

1. Introduction

East Sumba Regency is one of the districts in East Nusa Tenggara Province that has considerable potential in the livestock sector, especially for superior products such as horses. Horses are one of the livestock that are widely raised by the people of East Sumba, because horses are livestock that have many benefits for human life [1]. Apart from being a means of transportation, horses are also used by the community for sports, agriculture, defense support and even work, food, meat, and milk [2]. As production increases, the need for animal protein also increases in line with the growing public awareness of the importance of raising horse livestock. Therefore, the health of livestock managed by farmers is very important to meet nutritional needs and increase income for livestock owners [3]. Livestock health is one of the important aspects of maintenance management, and it also has a great influence on the productivity and sustainability of the livestock business. The health of livestock has a significant impact on its production if farmers pay attention to the health of their livestock. However, during the maintenance process, health problems often interfere with horse livestock, usually caused by diseases that directly affect the health of livestock, one of which is surra disease.

Sura disease is one of the infectious diseases that causes most of the horses affected by this disease to die. Surra is a disease caused by Trypanosoma evansi [4]. Surra disease in East Sumba Regency began to be found in August 2010 which attacked horses and buffaloes, the East Sumba Regency Animal Husbandry Office (2012) reported that Surra cases on Sumba Island, East Nusa Tenggara Province occurred in 2010 – 2011. The case resulted in 4268 cattle (1608 horses, 2464 buffaloes, 196 cows) infected with Sura disease. Currently, the diagnosis of diseases in horses at the Waingapu Livestock Office, East Sumba is still carried out through physical examination and laboratory analysis, which requires a lot of time and resources as well as special skills such as veterinarians, so the detection of diseases can take a long time because it depends on the services of doctors. With technological advancements, especially in the fields of artificial intelligence and image processing, there are now new and more efficient methods to detect diseases in livestock. One method that shows a lot of potential is the K-Nearest Neighbor (KNN) algorithm, which is a machine learning-based classification technique. The KNN method works by comparing new data features with data that is already known to the class, allowing disease detection to be carried out more quickly and practically.

The K-Nearest Neighbor (KNN) algorithm can be used to analyze data from horse blood sample images and detect the presence of disease. With the use of KNN, the detection process can be carried out faster and more accurately, making it easier for farmers to take preventive measures earlier if their livestock show symptoms of the disease. The use of these algorithms can help in early detection of diseases, improve livestock health, and reduce potential economic losses. In addition, the KNN algorithm has also been shown to be effective in the classification of medical images, as in a study conducted by Halim & Anraeni, showing that KNN is able to classify X-ray images of pneumonia patients with up to 96% accuracy [5], and the research conducted by Safitri et al., in the classification of grape leaves also has quite good accuracy [6]. So it has the potential to be applied to various types of medical imaging, including microscopic images of animal blood. Given the importance of horse health in the livestock industry, as well as the high mortality rate due to Sura disease in areas such as East Sumba, this study aims to detect Sura Disease on Microscopic Images of Horse Blood Using the K-Nearest Neighbor (KNN) Method. It is hoped that the results of this study can provide positive benefits for farmers and increase efficiency in livestock health management, especially in efforts to prevent and control Surra disease in the East Sumba region.

2. Library Studies

2.1. Horse Breeding

Horses (Equus caballus) are a type of even-hoofed mammals from the Equidae family that have long been domesticated by humans for various purposes, both as a means of transportation, food sources, and cultural symbols. This animal is known for its strength, high endurance, and intelligence, so it is often used in agricultural activities, cart pulling, and transportation in rural areas. In addition, horses are also used as racetracks in sports and recreation, and have high economic value, especially in areas that still preserve horse racing culture such as Sumba and Jeneponto. Another benefit of horse farming is that it is a source of animal protein through its meat and milk. Horse meat has lower fat content and high iron content than other red meats, while horse milk contains lactose and short-chain fatty acids that are beneficial for human digestive health [5]. In some parts of Indonesia, horses also have strong symbolic and cultural value, being used in traditional ceremonies, dowries, or qurbani in death ceremonies, making them an important part of the social structure of society [7].

2.2. Surra Disease (*Trypanosoma* Evansi)

Sura disease or trypanosomiasis in vertebrate animals is caused by Trypanosoma evansi, a protozoa of the family Trypanosomatidae. In Indonesia, surra has been designated as one of the strategic infectious animal diseases. This sura disease is responsible for the economic losses it causes. Losses due to surah in Asia are reported to reach 1.3 billion US dollars per year (CIVAS, 2014). Trypanosoma evansi initially developed in camels, then spread to many domestic mammals and wild animals. Flies of the genus Tabanus and Stomoxys play an important role in the mechanical transmission of trypanosoma agents evansi. Transmission can also occur through wound contamination with the blood of infected animals [4].

2.3. K-NEAREST NEIGHBOR (KNN)

The K-Nearest Neighbor (KNN) algorithm is one of many classification methods that uses learning data from previous classifications to classify other data. KKN which does not build a model but only studies data that has previously been classified makes KNN often referred to as a lazy learning technique. The K-Nearest Neighbor algorithm included in supervised learning will classify the results of the new instance query, where the majority of the proximity distance of the categories in the KNN will be used as the basis for the classification. In KNN, the class group with the closest vector distance will be the basis for choosing a new class for a data. In KNN, the k value means the closest k-data from the test data. In other words, k is the amount of data or neighbors who are closest to an object. There are several ways to determine the distance between training data and new data, one of which is by using Euclidean Distance. Euclidean distances can be calculated using equations [8].

2.4. Gray Level Co-occrrence Matrix (GLCM)

Gray Level Co-occurrence Matrix (GLCM) is a texture analysis method used in digital image processing to describe the spatial relationships between pixels in an image. This method was introduced by Haralick in 1973 and belongs to the second-order statistical approach, as it takes into account pairs of pixel intensity values in a given distance and direction. GLCM works by calculating how often a particular gray value pair appears in a specific relative position, such as horizontal, vertical, or diagonal. Commonly used features include contrast, homogeneity, energy, and correlation

2.5. Confusion Matrix

Naive Bayes The Confusion Matrix is a crucial tool in machine learning model evaluation because it provides a deeper understanding of model performance than just looking at accuracy. Confusion Matrix is a technique used in data mining and machine learning to calculate how well a model can predict the labels of a data [9]. There are four values produced in the confusion matrix table, including True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN).

	Positive	Negative
Positive	TP	FP
Negative	FN	MR

Figure 1: Confusion Matrix

3. Research Methodology

In this study, it was carried out using KNN in detecting circumcision disease based on microscopic images of horse blood.

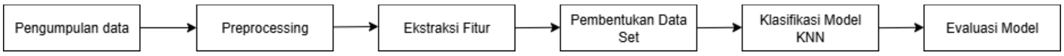


Figure 2: Research Flow

3.1. Data Collection

Data collection was carried out at the East Sumba Livestock Service in Waingapu in two ways, namely interviews and documentation.

3.2. Preprocessing

The preprocessing stages carried out in this study include image conversion from RGB format to grayscale and image size change to 64×64 pixels. Image conversion from RGB to grayscale aims to simplify the color information in the image without compromising important information related to the structure or pattern of the object being observed. It can also reduce data complexity and speed up the computational process. Once the image is converted to grayscale, the next stage is to resize the image to 64×64 pixels. This resize process aims to equalize the dimensions of all images in the dataset so that the model can recognize patterns consistently. The 64×64 size was chosen because it is small enough to speed up the computational process, but it is still able to retain the visual information relevant to the classification process.

3.3. Feature Extraction

Once the preprocessing stage is complete, which includes converting the image from RGB to grayscale format and resizing the image to 64×64 pixels, the next step is the feature extraction process. Feature extraction is an important stage in image processing because it aims to extract relevant information or characteristics from the image so that it can be used in the classification process. In this study, the feature extraction method used is the Gray Level Co-occurrence Matrix (GLCM). This method was chosen because it is able to statistically describe the texture of the image through the relationship between pixels in a certain gray level. The results of this process are in the form of feature values such as contrast, correlation, energy, and homogeneity, which will then be used as inputs at the classification stage.

3.4. Data Set Formation

3.5. KNN Classification

After the extraction process of image texture features using the Gray Level Co-occurrence Matrix (GLCM) method, five main features were obtained for each image, namely contrast, correlation, energy, homogeneity, and entropy. These features represent the texture characteristics of each image and are then arranged in the form of numerical vectors. These vectors form a dataset that is then used in the classification process. At this classification stage, the K-Nearest Neighbor (KNN) algorithm is used, which is a distance-based classification method that determines the class of a test data based on the majority of the class of the nearest K training data in the feature space. In this study, the K value used was set at 3, so that each class prediction was made by considering the three closest neighbors who had the smallest distance to the test data. The distances between feature vectors are calculated using the Euclidean distance metric, and the classification results will show whether the image falls into the category of "Normal" or "Infected".

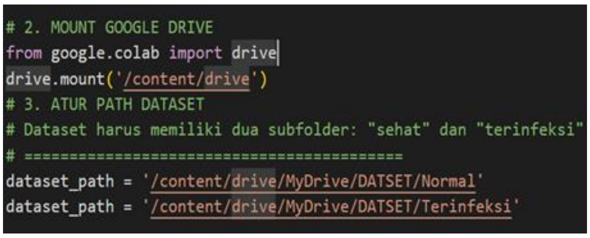
3.6. Model Evaluation

Once the model is trained and used to classify the test data, the next stage is the evaluation of the model. At this stage, we measure how well the model classifies images based on never-before-seen test data. Evaluations were conducted using several metrics, such as accuracy, which measures the percentage of correct predictions, as well as precision, recall, and F1-score, which provide a deeper picture of the model's performance, especially in dealing with unbalanced classes (e.g., more uninfected images than infected ones).

4. Results and Discussion

4.1. Data Collection

The data collection used in this study is a microscopic image of blood samples of horses that are normal and infected with Surra's disease. The image set data was obtained from the Animal Husbandry Service microscopic results on horse blood in the form of images. These images serve as the main data in the training and testing process of automated classification systems. After being obtained, the data is then entered into the Google Colaboratory (Colab) for further processing, including image preprocessing, feature extraction using the Gray Level Co-occurrence Matrix (GLCM) method, and classification using the K-Nearest Neighbor (KNN) method. The use of Colab allows data processing to be done efficiently, flexibly, and integrated with the relevant Python libraries. With this approach, the study aims to build a classification model that is able to automatically and accurately identify blood images of Surra-infected horses.



```
# 2. MOUNT GOOGLE DRIVE
from google.colab import drive
drive.mount('/content/drive')
# 3. ATUR PATH DATASET
# Dataset harus memiliki dua subfolder: "sehat" dan "terinfeksi"
# ==========================================
dataset_path = '/content/drive/MyDrive/DATSET/Normal'
dataset_path = '/content/drive/MyDrive/DATSET/Terinfeksi'
```

Figure 3: Upload Code

After the microscopic image set data is obtained from the Animal Husbandry Office, the next step is to import the data into the Google Colaboratory work environment. For this reason, the Google Drive mount process is carried out so that Colab can access the data set files stored in the Google Drive account. This process allows for direct integration between Colab and Drive, so that image files can be accessed and processed efficiently.

4.2. Preprocessing

After the data is successfully loaded from Google Drive, the next stage is to preprocess or pre-process the microscopic image of the horse's blood. Preprocessing aims to prepare the image to be more optimal and consistent before feature extraction and classification. This stage is very important because the quality of the input will greatly affect the performance of the classification model.

```
# Preprocessing
import cv2
import matplotlib.pyplot as plt
# Path ke gambar
path_gambar = '/content/drive/MyDrive/DATSET/Normal/1.jpg'

# Ukuran resize (misalnya 64x64)
ukuran_resize = (64, 64)

# Baca gambar dalam format BGR (OpenCV default)
image_bgr = cv2.imread(path_gambar)

# Cek apakah gambar berhasil dibaca
if image_bgr is None:
    print("❌ Gambar tidak ditemukan atau path salah.")
else:
    # Konversi BGR ke RGB (agar tampil benar di matplotlib)
    image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)
    # Konversi ke grayscale
    image_gray = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)

    # Resize grayscale (tidak ditampilkan)
    image_gray_resized = cv2.resize(image_gray, ukuran_resize, interpolation=cv2.INTER_AREA)
    # Tampilkan gambar RGB dan Grayscale original (tanpa resize)
    plt.figure(figsize=(10, 4))
    plt.subplot(1, 2, 1)
    plt.imshow(image_rgb)
    plt.title('Gambar RGB')
    plt.axis('off')

    plt.subplot(1, 2, 2)
    plt.imshow(image_gray, cmap='gray')
    plt.title('Gambar Grayscale')
    plt.axis('off')

    plt.tight_layout()
    plt.show()
```

Figure 4: Preprocessing Code

In the preprocessing stage, the first step is to change the image color format from RGB to grayscale. Images obtained from microscopes are generally color, but color information is not required in texture analysis using the Gray Level Co-occurrence Matrix (GLCM) method. Therefore, the image is first converted to grayscale format so that it only stores pixel intensity information. This conversion aims to simplify the data without compromising important information related to texture patterns. Once the image is converted to grayscale, the next step is to resize the resize of the image to 64x64 pixels. This process is done to equalize the dimensions of all images in the data set so that they can be processed efficiently and consistently. The size was chosen because it was small enough to speed up computation, but still able to maintain the relevant texture patterns in the blood image. With this preprocessing, the image is ready to enter the feature extraction and classification stage using the K-Nearest Neighbor (KNN) algorithm.

Based on figure 4. Preprocessing results are obtained as follows:

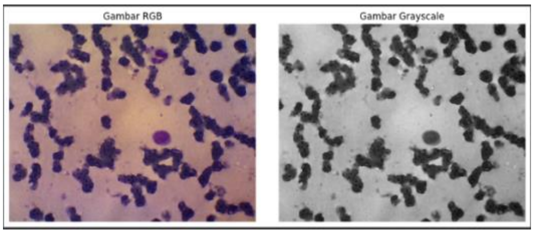


Figure 5: Preprocessing Results

4.2.1. Feature Extraction

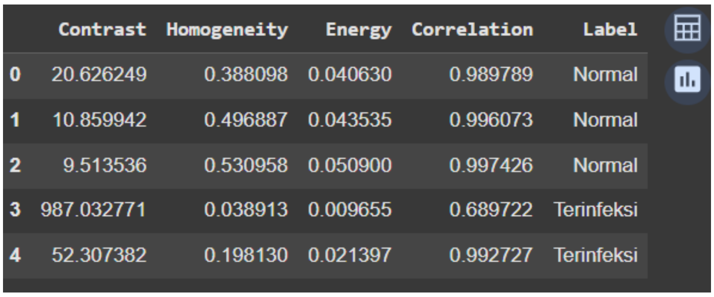
After the image goes through the preprocessing stage, the next step is to extract features to obtain important information that can be used in the classification process. In this study, the method used is the Gray Level Co-occurrence Matrix (GLCM). GLCM is a texture analysis technique that calculates how often pairs of certain pixel intensity values appear in a certain direction and distance in a grayscale image. This technique is effective for capturing the patterns and textured structures that often appear in microscopic images.

```
# Load data dan ekstraksi fitur GLCM
data = []
labels = []

# Path dataset
Normal_path = '/content/drive/MyDrive/DATSET/Normal'
Terinfeksi_path = '/content/drive/MyDrive/DATSET/Terinfeksi'
# Proses gambar Normal
for img_name in os.listdir(Normal_path):
    img_path = os.path.join(Normal_path, img_name)
    if os.path.isfile(img_path):
        img = cv2.imread(img_path)
        if img is not None:
            fitur = ekstraksi_fitur_lengkap(img)
            data.append(fitur)
            labels.append("Normal")

# Proses gambar Terinfeksi
for img_name in os.listdir(Terinfeksi_path):
    img_path = os.path.join(Terinfeksi_path, img_name)
    if os.path.isfile(img_path):
        img = cv2.imread(img_path)
        if img is not None:
            fitur = ekstraksi_fitur_lengkap(img)
            data.append(fitur)
            labels.append("Terinfeksi")

# Buat DataFrame hasil ekstraksi
df = pd.DataFrame(data, columns=['Contrast', 'Homogeneity', 'Energy', 'Correlation'])
df['Label'] = labels

# Acak data
df = df.sample(frac=1, random_state=42).reset_index(drop=True)

# Tampilkan data
display(df.head())
```

Figure 6: Feature Extraction Code

After the feature extraction function is completed, the next stage is to load the image data from two classes, namely Normal and Infected, then apply the extraction function to each image. This process begins by defining the path of the data set folder that contains the images of each class. The normal folder contains microscopic images of the blood of healthy horses, while the infected folder contains images of the blood of horses identified as containing the parasite Trypanosoma evansi, the cause of Surra's disease. Each image that is successfully read will be processed using the full feature extraction function, which has been previously defined. This function converts to grayscale, resizes the image, and calculates the four main texture parameters of the GLCM, namely contrast, homogeneity, energy, and correlation. The results of the calculation of this feature are then stored in a data list, and the labels according to their class are included in the list of labels. Once all the images have been successfully processed and the features extracted, the data and labels are combined into a DataFrame with four feature columns and one label column. To ensure that the data sequence does not affect the model's training process later, the data is randomized using the sample() function with a fixed random_state to keep the replication of the results consistent. This stage marks the end of the feature extraction process and all data is now ready to be used in the process of sharing training data and test data as well as the classification stage using the K-Nearest Neighbor (KNN) algorithm.

Based on figure 6. The results of the extraction of the features are obtained as follows:

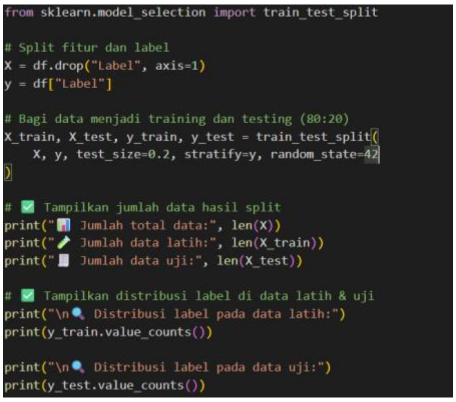


	Contrast	Homogeneity	Energy	Correlation	Label
0	20.626249	0.388098	0.040630	0.989789	Normal
1	10.859942	0.496887	0.043535	0.996073	Normal
2	9.513536	0.530958	0.050900	0.997426	Normal
3	987.032771	0.038913	0.009655	0.689722	Terinfeksi
4	52.307382	0.198130	0.021397	0.992727	Terinfeksi

Figure 7: Feature Extraction Results

4.2.2. Data Set Formation

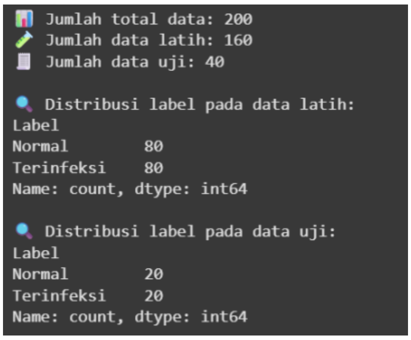
Once the entire image has been successfully processed and the texture features are extracted using the Gray Level Co-occurrence Matrix (GLCM) method, the next step is to form a structured dataset that is ready to be used for the training and testing process of the classification model. This dataset is created in the form of a DataFrame using the pandas library, with each row representing an image and each column containing the values of the features that have been calculated, namely contrast, homogeneity, energy, and correlation. In addition to features, each row is also labeled as a class in the form of normal or infected, according to the condition of the original image. The formation of this data set aims to organize the data systematically and make it easier in the next analysis process. After the data set is formed, a data shuffling process is carried out to ensure that the data sequence does not affect the model training process. Randomization is done using the sample() function with a specific random_state so that the results obtained remain consistent if retested. This data set that has been formed and randomized then becomes the basis for the process of sharing training data and test data.



```
from sklearn.model_selection import train_test_split

# Split fitur dan label
X = df.drop("Label", axis=1)
y = df["Label"]

# Bagi data menjadi training dan testing (80:20)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)

# ✅ Tampilkan jumlah data hasil split
print("📊 Jumlah total data:", len(X))
print("🧪 Jumlah data latih:", len(X_train))
print("📄 Jumlah data uji:", len(X_test))

# ✅ Tampilkan distribusi label di data latih & uji
print("\n🔍 Distribusi label pada data latih:")
print(y_train.value_counts())

print("\n🔍 Distribusi label pada data uji:")
print(y_test.value_counts())
```

Figure 8: Data Set Formation Code

After the data set is formed, the next stage is to divide the data into two parts, namely training data and test data. This division aims to allow models to be trained using some of the data, and then tested for their ability on data that has never been seen before. In this study, the data was divided into a proportion of 80% as training data and 20% as test data using the train_test_split function of the sklearn library. Before sharing, the data set is first separated into two components, namely X which contains features (Contrast, Homogeneity, Energy, and Correlation), and y which contains class labels (Normal or Infected). The train_test_split function is used with the stratify=y parameter to keep the class distribution on the drill data and test data balanced. In addition, random_state=42 is specified so that the data sharing is random but still replicable.



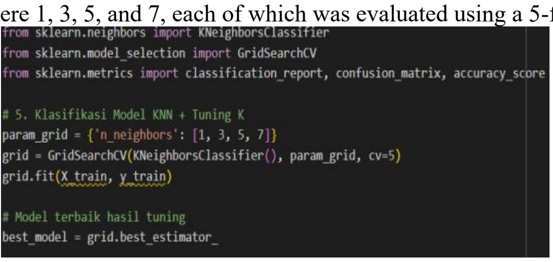
```
📊 Jumlah total data: 200
🧪 Jumlah data latih: 160
📄 Jumlah data uji: 40

🔍 Distribusi label pada data latih:
Label
Normal        80
Terinfeksi    80
Name: count, dtype: int64

🔍 Distribusi label pada data uji:
Label
Normal        20
Terinfeksi    20
Name: count, dtype: int64
```

Figure 9: Dataset Formation Results

After the partitioning process, the amount of data on each *subset* is displayed to ensure that the partition is successful and proportional. The distribution of labels in each *subset* is also examined, to ensure that there are no class inconsistencies that could affect the model's performance at the classification stage.

4.2.3. Model Classification

After this stage, a classification model training process was carried out using the K-Nearest Neighbor (KNN) algorithm. To obtain optimal model performance, parameter adjustments (hyperparameter tuning) are made to the value of K or the number of closest neighbors used by the algorithm. The K-values tested were 1, 3, 5, and 7, each of which was evaluated using a 5-fold cross-validation technique.



```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score

# 5. Klasifikasi Model KNN + Tuning K
param_grid = {'n_neighbors': [1, 3, 5, 7]}
grid = GridSearchCV(KNeighborsClassifier(), param_grid, cv=5)
grid.fit(X_train, y_train)

# Model terbaik hasil tuning
best_model = grid.best_estimator_
```

Figure 10: Model Classification

The tuning process is carried out using the *GridSearchCV* function of the *scikit-learn library*, which automatically tests various combinations of parameters and selects the best model based on the results of cross-validation evaluation. After the training process is completed, the best KNN model is obtained which is stored in *the best_model* variable. It is this best model that is then used to classify the test data. This process aims to evaluate the model's ability to distinguish between normal and surra-infected horses' blood images based on the four texture features of GLCM extraction, namely *contrast, homogeneity, energy,* and *correlation*.

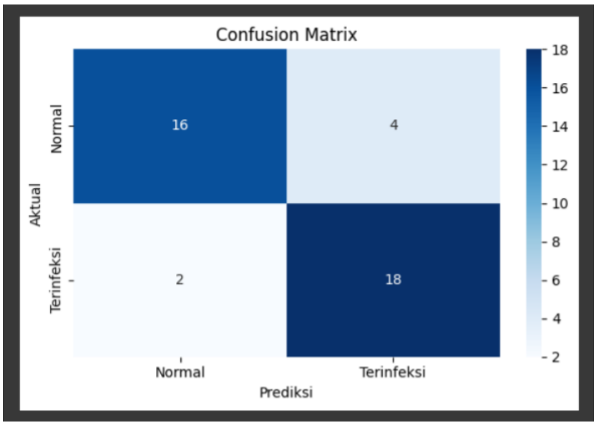
4.2.4. Model Evaluation

After the best K-Nearest Neighbor (KNN) model is obtained through the Grid Search process, the next step is to evaluate the model's performance using test data. This evaluation was carried out by comparing the prediction results of the model against the actual label on the test data (y_test). The goal was to find out how well the model classified the horse's blood images into two categories, namely normal and infected. Then the confusion matrix value is calculated as well as other evaluation metrics such as precision, recall, and F1-score. Here is a script to perform model evaluation:

```
# 6. Evaluasi Model
# ============================

# Make predictions
y_pred = best_model.predict(X_test)

# Define class names for the report and confusion matrix
kelas = sorted(y.unique())

akurasi = accuracy_score(y_test, y_pred)
print("=== HASIL EVALUASI ===")
print("Akurasi       :", round(akurasi * 100, 2), "%")
print("K terbaik     :", grid.best_params_['n_neighbors'])
print("Laporan Klasifikasi:\n", classification_report(y_test, y_pred, target_names=kelas))

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=kelas, yticklabels=kelas)
plt.xlabel("Prediksi")
plt.ylabel("Aktual")
plt.title("Confusion Matrix")
plt.tight_layout()
plt.show()
```

Figure 11: Model Evaluation

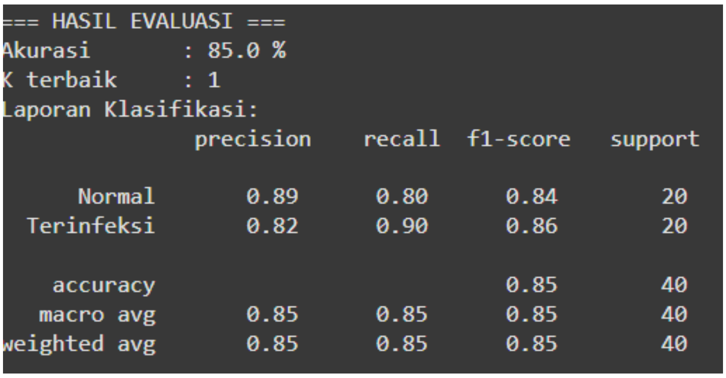
After the evaluation process is carried out, a confusion matrix graph is obtained that describes the model's prediction results on the test data.



Gambar 12: Confusion Matrix

Based on the confusion matrix in Figure 12, it can be seen that the classification model is able to identify blood images with fairly good performance. Out of a total of 40 images in the validation data, 16 normal blood images were correctly predicted, while the other 4 normal images were misclassified as infected. On the other hand, out of the infected class, there were 18 images that were correctly classified, and only 2 images were incorrectly predicted as normal. Although there were some misclassifications, especially in normal classes, the model generally showed a better ability to recognize infected images, which was characterized by a high sensitivity to the condition. These results indicate that the model is already quite effective in recognizing the typical visual patterns of surra disease, although the accuracy to normal classes can still be improved.

In addition to using a confusion matrix, the evaluation of model performance is also strengthened through a classification report that presents metrics such as precision, recall, and f1-score for each class. The following is a picture for the evaluation metrics:



```
=== HASIL EVALUASI ===
Akurasi       : 85.0 %
K terbaik     : 1
Laporan Klasifikasi:
               precision    recall  f1-score   support

      Normal       0.89      0.80      0.84        20
  Terinfeksi       0.82      0.90      0.86        20

    accuracy                           0.85        40
   macro avg       0.85      0.85      0.85        40
weighted avg       0.85      0.85      0.85        40
```

Figure 13: Evaluation Results

Based on Figure 13, the results of the evaluation of the K-Nearest Neighbor (KNN) model show that the model has an accuracy level of 85% on the test data, with the best k-value of 1. The evaluation was detailed using precision, recall, and f1-score metrics for each class, namely Normal and Infected. For the Normal class, the model obtains a precision value of 0.89, which means that 89% of all "Normal" predictions are correct. A recall value of 0.80 indicates that 80% of all completely normal images are correctly recognized. The f1-score value of 0.84 represents a balance between precision and recall in this class. Meanwhile, in the Infected class, the precision model is 0.82, with a recall of 0.90. This suggests that the model is very good at recognizing images that are actually infected, although there are still some normal images that are misclassified as infected. The F1-score for the Infected class of 0.86 indicates the model's excellent performance in detecting surra disease. The overall average of the model, both macro average and weighted average, was 0.85 on all three metrics (precision, recall, and f1-score), indicating the model's stable and consistent performance in both classes.

Overall, the KNN model with k = 1 is able to classify blood images with high effectiveness, especially in recognizing conditions infected with surra disease. However, the model still has room for performance improvement in normal classes, so that the classification distribution is more balanced and reduces misclassification.

5. Conclusion

Based on the research that has been conducted, it can be concluded that the feature extraction method using the Gray Level Co-occurrence Matrix (GLCM) is effective in representing the visual pattern of livestock blood images. The features used, namely contrast, homogeneity, energy, and correlation, are able to capture important characteristics of normal blood conditions as well as those infected with surra disease. The K-Nearest Neighbor (KNN) classification model used in this study gave the best results at the nearest neighbor (k) value of 1. The model managed to achieve an accuracy of 85% on the test data consisting of 40 images. In addition, the average precision and recall values achieved, each of 0.85, indicate that the model has a fairly good and balanced performance. The model was able to detect infected blood images very well, indicated by a recall of 0.90, while its performance in detecting normal blood images was slightly lower with a recall of 0.80. Thus, this KNN model shows high effectiveness in detecting surra disease infections, although there is still an opportunity to improve the model's ability to recognize normal blood conditions more accurately.

Confession

The author would like to thank Wira Wacana Christian University Sumba and the Informatics Engineering Study Program for the support and facilities provided during this research process. Thank you also to the supervisors and colleagues who have helped in collecting and processing data, so that this research can be completed properly.

Reference

[1] M. R. M. Ndiha, I. A. P. Apsari, and I. M. Dwinata, "Prevalence and Intensity of Trypanosoma Evansi Infection in Horses in Kabaru Village, Rindi District, East Sumba Regency," *Bul. Vet. Udayana*, vol. 10, no. 1, p. 70, 2020, doi: 10.24843/bulvet.2018.v10.i01.p11.

[2] M. U. Horung and I. P. Sirappa, "The Influence of Breeder Characteristics on Community Interest in Horse Breeding in Napu Village, Haharu District, East Sumba Regency.," *J. Peternak. Sabana*, vol. 1, no. 1, p. 48, 2022, doi: 10.58300/jps.v1i1.218.

[3] D. Hutagalung and Z. Paulanda, "Early Prevention of Cattle Disease with Expert Systems," *J. Tekensos*, vol. 2, no. 1, 2020.

[4] N. S. Dharmawan, I. G. Mahardika, U. Y. A. Praing, and L. D. Anggreni, "Biochemical Profile of Horse Blood Tracked with Surra Disease in East Sumba, East Nusa Tenggara, Indonesia," *J. Vet.*, vol. 25, no. 2, pp. 235–242, 2024, doi: 10.19087/jveteriner.2024.25.2.235.

[5] E. Safitri, R. Heppy Ria Sibarani, Y. SM Sidabutar, and D. Kiswanto, "Classification of Grape Leaf Disease Based on Image Using the K-Nearest Neighbors (Knn) Method," *JATI (Journal of Mhs. Tek. Inform.*, vol. 8, no. 6, pp. 12633–12642, 2024, doi: 10.36040/jati.v8i6.12004.

[6] A. A. D. Halim and S. Anraeni, "Analysis of Pneumonia Disease Image Dataset Classification Using the K-Nearest Neighbor (KNN) Method," *Indones. J. Data Sci.*, vol. 2, no. 1, pp. 01–12, 2021, doi: 10.33096/ijodas.v2i1.23.

[7] A. Astati, M. Mursidin, R. Rusny, H. I. Susanti, A. Hifizah, and A. Qurniawan, "Public Perception of the Benefits of Horse Breeding in Jeneponto Regency," *Anoa J. Anim. Husb.*, vol. 3, no. 1, pp. 1–9, 2024, doi: 10.24252/anoa.v3i1.44387.

[8] M. F. Akbarollah, W. Wiyanto, D. Ardiatma, and A. T. Zy, "Application of the K-Nearest Neighbor Algorithm in the Classification of Heart Disease," *J. Comput. Syst. Informatics*, vol. 4, no. 4, pp. 850–860, 2023, doi: 10.47065/josyc.v4i4.4071.

[9] U. Riyanto, "Comparative Analysis of Naive Bayes Algorithm and Support Vector Machine in Classifying the Number of Readers of Online Articles," *IF (Inform.*, vol. 2, no. 2, pp. 62–72, 2019, doi: 10.31000/.v2i2.1521.