JOURNAL OF APPLIED INFORMATION SYSTEMS AND INFROMATICS

Global Sentiment Analysis of Video Surveillance Technology Using BERT

Syifaa Puspita Rahayu[a], Ade Bastian[a*], Dadan Zaliluddin[a], Ii Sopiandi[a], Ardi Mardiana[a]

[a] *Department of Informatics, Universitas Majalengka, KH Abdul Halim Street No. 108, Majalengka, 45418, Indonesia*

Corresponding author: adebastian@unma.ac.id

***Abstract*— The significant imbalance between population growth and the availability of video surveillance systems in several major cities across different countries has raised concerns regarding the effectiveness of public space monitoring. As urban areas become increasingly dense and complex, CCTV is expected to function not only as a security support tool but also as part of a broader strategy for crime prevention, public safety, and urban management. However, the implementation of video surveillance also generates diverse public opinions, especially on social media platforms such as Twitter (X), where users actively express their views, support, concerns, and criticism. This study aims to identify public sentiment trends toward video surveillance and evaluate the performance of the BERT (Bidirectional Encoder Representations from Transformers) model in classifying these sentiments. The research uses social media data collected from various countries and applies BERT as a contextual natural language processing model. The findings show that most users expressed positive sentiments toward video surveillance, indicating that CCTV is generally perceived as beneficial for improving security and monitoring public spaces. In terms of model performance, BERT achieved its highest accuracy of 0.85 during the third trial at epoch 15. However, a slight decrease in accuracy occurred at epoch 20, indicating the possibility of overfitting when the model was trained for too long. These findings suggest that BERT is effective in capturing public opinion contextually and can be used as a valuable analytical tool to support evidence-based decision-making related to surveillance technology implementation in urban environments.**

***Keywords*— Video Surveillance, Sentiment Analysis, BERT, Twitter (X)**

Manuscript received 17 Nov. 2025; revised 03 Mar 2026; accepted 12 May 2026. Date of publication 25 May 2026.

I. INTRODUCTION

Intelligent video surveillance has become an essential resource for maintaining safety in numerous scenarios. This system utilizes cutting-edge technologies like computer vision, machine learning, and artificial intelligence, enabling immediate recognition, assessment, and management of potential threats[1]. As technology evolves, monitoring systems are becoming more sophisticated and capable of identifying behavior patterns and security threats. Nevertheless, this advancement has not been consistently smooth. Several major cities in the world still have a very low CCTV ratio compared to their population. For example, Nairobi in Kenya has only 42 cameras for more than 4.5 million residents. This situation indicates a gap in the implementation of surveillance systems, which can impact the effectiveness of public protection and the accuracy of incident reporting. On the other hand, the existence of video surveillance has also elicited responses from the public, especially on social media platforms such as X (Twitter). This platform is used by millions of users around the world to voice their opinions on various public policies, including video surveillance. With over 25 million users in Indonesia alone, X holds a wealth of public opinion data that can be leveraged to gain a broader understanding of public perception[2]. Sentiment analysis is part of text mining research that performs text document classification procedures[3]. There are many algorithms used to perform sentiment analysis, including naïve bayes, support vector machine, CNN, decision tree, BERT, and many more. Of these various models, BERT stands out due to its ability to understand context more deeply.

BERT (Bidirectional Encoder Representations from Transformers) is an AI model from Google that uses a transformer architecture to optimize language understanding. This model has the ability to read text in two directions, namely from front to back and vice versa[4].

Accidents in the workplace are a serious problem that contributes to high mortality rates. Video surveillance technology has proven effective in identifying risky behaviors through in-depth analysis and long-term monitoring. Without this system, safety reports tend to be inaccurate and prone to bias, thereby reducing the quality of risk management[5].

Furthermore, in the educational environment, the installation of CCTV cameras has proven to be effective in reducing the

incidence of violence, theft, and bullying, while also supporting the creation of a safer environment for students and teaching staff. The implementation of a Closed Circuit Television (CCTV) system in school areas is an appropriate solution because it can help in monitoring student activities, controlling negative behavior, and improving the overall security surveillance system[6]. Meanwhile, in public spaces such as the city of Recife, the lack of surveillance technology hinders the creation of smart services that improve public safety. The placement of cameras at strategic points has been proven to reduce crime rates, demonstrating that video surveillance systems play a vital preventive role in maintaining safety in various aspects of life[7].

Based on the description of the background of the existing problems, it is necessary to conduct research on the application of BERT for sentiment analysis in order to understand public perceptions in various countries about video surveillance and to find out positive or negative feedback on the implementation of Video Surveillance.

II. MATERIALS AND METHOD

This research begins with collecting data from Twitter, followed by text preprocessing to prepare the data for analysis. The processed dataset is then split into training, validation, and testing sets. The text is tokenized using the BERT tokenizer and used for training a pre-trained BERT model. Finally, the models performance is evaluated using standard classification metrics.

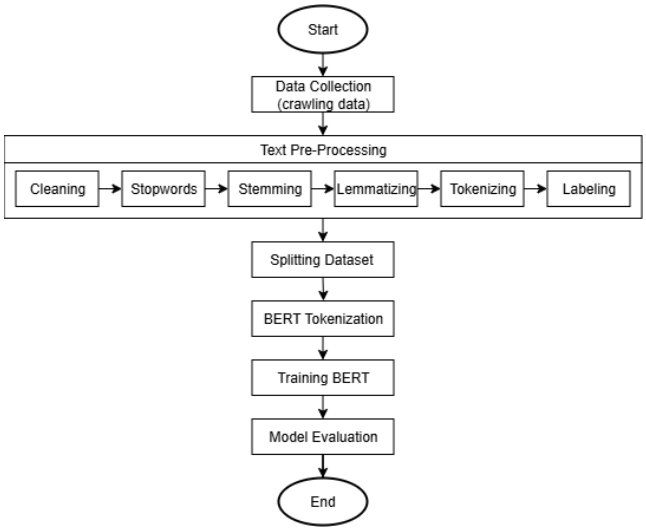

Figure 1. Method

A. Data Collection

Data collection or data crawling refers to a software application that autonomously collects information from the internet. This technique allows for organized and effective gathering of data from multiple origins[8].

Table 1. Keyword used

No	Keyword Crawling	Total number of datasets
1	Video Surveillance	January 2024 - March 2025 1071 datasets

B. Text Pre-Processing

A vital part of the sentiment analysis is the preprocessing stage, where unprocessed data is transformed into a structure that computers and machine learning algorithms can interpret. This key phase significantly influences the accuracy and effectiveness of machine learning models[9].

The following are the stages of preprocessing that are very important in this study.

1. Cleaning

   The initial stage is text cleaning, which aims to remove unnecessary elements or characters in the text, such as punctuation marks, numbers, uncommon symbols, and irrelevant characters. For example, in data from social media, cleaning is done by eliminating HTML tags, URLs, or emojis so that the text looks neater and more organized.

2. Stopwords

   Stopwords are words that are frequently found in text but are not considered important in text analysis or natural language processing, such as “that,” “from,” “to,” “is,” and so on. In text analysis, stopwords are generally removed because they do not contribute significantly to the understanding of the context or meaning of a document.

3. Stemming

   Stemming is a method for converting words that have affixes (whether prefixes, suffixes, or infixes) into their base form. The main purpose of stemming is to simplify various word variations so that they can be analyzed consistently, especially in the context of information retrieval, text classification, and sentiment analysis.

4. Lemmatization

   Lemmatization is the process of transforming words into their basic form or lemma. Unlike stemming, which tends to cut words in a crude manner, lemmatization takes linguistic context into account so that the resulting words retain their correct grammatical meaning.

5. Tokenizing

   Tokenizing is the process of dividing text into small elements known as tokens. These tokens can be words, phrases, or characters, depending on the requirements. The tokenizing process facilitates machine understanding of text structure and supports subsequent processing steps.

6. Labeling

   The final stage in preprocessing is labeling, which is the step of assigning sentiment labels to the data, either manually or automatically. The labels assigned usually include sentiment categories such as positive, negative, or neutral.

C. Splitting Dataset

Data splitting is the process of dividing a dataset into three main parts: training set, validation set, and testing set. This is done to ensure that the machine learning model is well learned on the training data, tested with the validation data to adjust the model parameters, and measured for performance with the test data[10].

D. BERT Tokenization

Tokenization with BERT: This tokenization is used to break text into smaller tokens according to the rules of the BERT model. The text is parsed into tokens, then padded if it is too short or truncated if it is too long. This ensures that the text reaches the specified maximum length[11].

E. *Training BERT*

Train the prediction model using the training data and compare it with the actual labels to calculate the loss value. The training process lasts for several epochs[12].

F. *Model Evaluation*

Model evaluation is the stage after processing data mining using a classification model, with the Confusion Matrix used as a performance measure. Basically, this measurement serves as a tool to evaluate the suitability between the classification results produced by the algorithm and the actual labels.

The confusion matrix consists of four main elements, namely True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN)[13].

III. RESULT AND DISCUSSION

A. *Data Collection*

This research began by collecting data from Twitter, consisting of comments and opinions in English with the keyword "Video Surveillance." The tool used to collect opinions from Twitter social media was Tweet Harvest. It is a tool that utilizes Application Programming Interface (API) to extract text data and obtain relevant information.

Figure 2. Crawling data

Figure 2 shows the process of collecting data from Twitter using the keyword "video surveillance" through tweet-harvest version 2.6.1. The data collection process was carried out in stages by scrolling until the required number of tweets was obtained. The results show that 1,071 tweets were successfully collected through crawling, limited to the period between January 1, 2024, and March 23, 2025, and filtered to include only tweets in English.

Table 2 Crawling Results

No	Created at	Full Text	Location	User Id
1	Sat Mar 22 18:00: 41 +0000 2025	Maybe also add some window stickers saying the car has 24 hour video surveillance :-) I don't want an all electric car because I've heard they emit a lot of EMFs and I've read reports of people complaining about headaches and such if they've been driving a long time. Someone		rainlvr737
...	...	…	...	...
1071	Sat Mar 08 06:10: 28 + 0000 2025	@beatlejonatika NYC Police Search For Suspect Who Shot 2 Sleeping Homeless Men One Fatally. Surveillance video of the violence is chilling to see		Jonathan

B. *Text Pre-Processing*

The data pre-processing stage plays an important role before implementing the BERT model, as this stage is responsible for processing raw data obtained through the crawling process.

1. Cleaning

   Data cleansing is the process of removing or correcting data that is inaccurate, duplicated, empty, inconsistent, or contains errors from a collection of obtained data. This process aims to improve data quality so that it is better suited for use in analysis, modeling, or visualization. At this stage, unused columns were deleted and duplicate data was removed, reducing the dataset from 1071 to 1044.

Table 3 Cleaning Results

No	Before Cleaning	After Cleaning
1	@kristenmag Maybe also add some window stickers saying the car has 24 hour video surveillance :-) I don't want an all electric car because I've heard they emit a lot of EMFs and I've read reports of people complaining about headaches and such if they've been driving a long time. Someone	kristenmag maybe also add some window stickers saying the car has 24 hour video surveillance i dont want an all electric car because i ve heard they emit a lot of emfs and i ve read reports of people complaining about headaches and such if they ve been driving a long time someone
2	@J_Jammer@manny fidel The topic is about Tesla's video surveillance of its customers and legitimate doubts about their data security. Nothing to do with violence. You opened the topic with	J jammer mannyfidel the topic is about tesla s video surveillance of its customers and legitimate doubts about their data security nothing to do with violence you opened the topic with an unjustified accusation

No	Before Cleaning	After Cleaning
	an unjustified accusation against me. Reading comprehensively is appropriate.	against me reading comprehensively is appropriate
…	…	…
1044	@beatlejonatika NYC Police Search For Suspect Who Shot 2 Sleeping Homeless Men One Fatally. Surveillance video of the violence is chilling to see	beatlejonatika nyc police search for suspect who shot 2 sleeping homeless men one fatally. surveillance video of the violence is chilling to see

2. Stopwords

The purpose of stopwords is to streamline data and focus analysis on relevant keywords.

Table 4. Stopwords Results

No	Before Stopwords	After Stopwords
1	kristenmag maybe also add some window stickers saying the car has 24 hour video surveillance i dont want an all electric car because i ve heard they emit a lot of emfs and i ve read reports of people complaining about headaches and such if they ve been driving a long time someone	kristenmag maybe also add window stickers saying car 24 hour video surveillance want electric car heard emit lot emfs read reports people complaining headaches driving long time someone
2	J jammer mannyfidel the topic is about tesla s video surveillance of its customers and legitimate doubts about their data security nothing to do with violence you opened the topic with an unjustified accusation against me reading comprehensively is appropriate	j jammer mannyfidel topic tesla s video surveillance customers legitimate doubts data security nothing violence opened topic unjustified accusation me reading comprehensively appropriate
…	…	…
1044	beatlejonatika nyc police search for suspect who shot 2 sleeping homeless men one fatally. surveillance video of the violence is chilling to see	beatlejonatika nyc police search suspect shot 2 sleeping homeless men one fatally surveillance video violence chilling see

3. Stemming

The purpose of this step is to ensure that the text data has consistency in word structure.

Table 5. Stemming Results

No	Before Stopwords	After Stopwords
1	kristenmag maybe also add window stickers saying car 24 hour video surveillance want electric car heard emit lot emfs read reports people complaining headaches driving long time someone	kristenmag mayb also add window sticker say car 24 hour video surveil want electr car heard emit lot emf read report peopl complain headach drive long time someon
2	j jammer mannyfidel topic tesla s video surveillance customers legitimate doubts data security nothing violence opened topic unjustified accusation me reading comprehensively appropriate	j jammer mannyfidel topic tesla s video surveil custom legitim doubt data secur noth violenc open topic unjustifi accus me read comprehens appropri
…	…	…
1044	beatlejonatika nyc police search suspect shot 2 sleeping homeless men one fatally surveillance video violence chilling see	beatlejonatika nyc polic search suspect shot 2 sleep homeless men one fatal surveil video violenc chill see

4. Lemmatization

The lemmatization stage is applied to ensure that words are converted to their basic form in accordance with grammatical structure.

Table 6. Lemmatization Results

No	Before Stopwords	After Stopwords
1	kristenmag mayb also add window sticker say car 24 hour video surveil want electr car heard emit lot emf read report peopl complain headach drive long time someon	kristenmag maybe also add window sticker say car 24 hour video surveillance want electric car heard emit lot emfs read report people complain headache drive long time someone
2	j jammer mannyfidel topic tesla s video surveil custom legitim doubt data secur noth violenc open topic unjustifi accus me read comprehens appropri	j jammer mannyfidel topic tesla s video surveillance customer legitimate doubt data security nothing violence open topic unjustified accusation me read

No	Before Stopwords	After Stopwords
		comprehensively appropriate
…	…	…
1044	beatlejonatika nyc polic search suspect shot 2 sleep homeless men one fatal surveil video violenc chill see	beatlejonatika nyc police search suspect shot 2 sleep homeless men one fatally surveillance video violence chill see

5. Tokenization

After the lemmatization process, the next step is tokenization, which is breaking down the text into basic word units called tokens.

Table 7. Tokenization Results

No	Before Stopwords	After Stopwords
1	kristenmag maybe also add window sticker say car 24 hour video surveillance want electric car heard emit lot emfs read report people complain headache drive long time someone	[‘kristenmag’, ‘maybe’, ‘also’, ‘add’, ‘window’, ‘sticker’, ‘say’, ‘car’, ‘24’, ‘hour’, ‘video’, ‘surveillance’, ‘want’, ‘electric’, ‘car’, ‘heard’, ‘emit’, ‘lot’, ‘emfs’, ‘read’, ‘report’, ‘people’, ‘complain’, ‘headache’, ‘drive’, ‘long’, ‘time’, ‘someone’]
2	j jammer mannyfidel topic tesla s video surveillance customer legitimate doubt data security nothing violence open topic unjustified accusation me read comprehensively appropriate	[‘j’, ‘jammer’ ‘mannyfidel’, ‘topic’, ‘tesla’, ‘s’, ‘video’, ‘surveillance’, ‘customer’, ‘legitimate’, ‘doubt’, ‘data’, ‘security’, ‘nothing’, ‘violence’, ‘open’, ‘topic’, ‘unjustified’, ‘accusation’, ‘me’, ‘read’, ‘comprehensively’, ‘appropriate’]
…	…	…
1044	beatlejonatika nyc police search suspect shot 2 sleep homeless men one fatally surveillance video violence chill see	[‘beatlejonatika’, ‘nyc’, ‘police’, ‘search’, ‘suspect’, ‘shot’, ‘2’, ‘sleep’, ‘homeless’, ‘men’, ‘one’, ‘fatally’, ‘surveillance’, ‘video’, ‘violence’, ‘chill’, ‘see’]

6. Labeling

After the preprocessing stage is complete, the next step is to label the text data to identify the sentiment category of each tweet. The labels applied in this study consist of two categories, namely positive and negative.

Table 8. Labeling Results

No	Tweet	Sentiment
1	kristenmag maybe also add window sticker say car 24 hour video surveillance want electric car heard emit lot emfs read report people complain headache drive long time someone	Negative
2	j jammer mannyfidel topic tesla s video surveillance customer legitimate doubt data security nothing violence open topic unjustified accusation me read comprehensively appropriate	Positive
3	elonmusk large sign warn post tesla dealership immediately especially time democrat plan 3 29 protest include warn dealership 24 hour hi def video surveillance act violence refer fbi	Positive

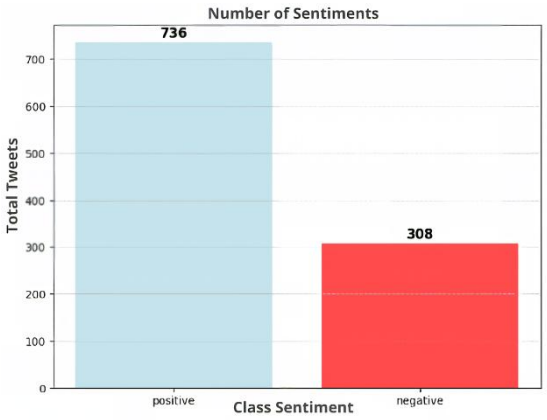

Figure 3. Number of Sentiments

Figure 3 above shows the number of sentiment analyses on the issue of video surveillance. Of the total tweets analyzed, there were 736 tweets with positive sentiment and 308 tweets with negative sentiment.

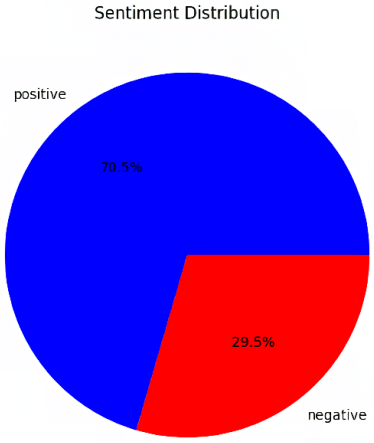

Figure 4. Sentiment Distribution

Figure 4 above shows the distribution of sentiment towards the issue of video surveillance. The analysis results show that the majority of sentiment is positive, at 70.5%, while negative sentiment is at 29.5%. This indicates that most users are supportive of or have no objection to the implementation of video surveillance.

C. Splitting Dataset

In this study, researchers used the BERT-base-uncased model, which is a variant of the BERT (Bidirectional Encoder

Representations from Transformers) model that has been pre-trained by Google. The dataset was divided into three parts, namely training data (70%), validation data (15%), and test data (15%) proportionally based on sentiment labels, and displayed the class distribution in each part of the data.

```
from sklearn.model_selection import train_test_split

# Pisahkan data menjadi Train (70%) dan Temp (30%)
df_train, df_temp = train_test_split(df, test_size=0.3, random_state=42, stratify=df['sentiment'])

# Pisahkan Temp menjadi Validation (15%) dan Test (15%)
df_val, df_test = train_test_split(df_temp, test_size=0.5, random_state=42, stratify=df_temp['sentiment'])

print("Train Data Size: ", len(df_train))
print("Validation Data Size: ", len(df_val))
print("Test Data Size: ", len(df_test))

# Cek distribusi kelas setelah pembagian
print("\nDistribusi Kelas di Train Set:\n", df_train['sentiment'].value_counts(normalize=True))
print("\nDistribusi Kelas di Validation Set:\n", df_val['sentiment'].value_counts(normalize=True))
print("\nDistribusi Kelas di Test Set:\n", df_test['sentiment'].value_counts(normalize=True))
```

Figure 5. Splitting Dataset Code

D. BERT Tokenization

The next step is to use the code in Figure 6 to load the tokenizer from the pre-trained BERT-base uncased model taken from Hugging Face.

```
[ ] PRETRAINED_MODEL = 'bert-base-uncased'  # https://huggingface.co/google-bert/bert-base-uncased

    from transformers import BertTokenizer

    tokenizer = BertTokenizer.from_pretrained(PRETRAINED_MODEL)
    vocab = tokenizer.get_vocab()
```

Figure 6. BERT Tokenization Code

E. Training BERT

The code in Figure 7 is used to build a model with the Adam optimizer, the SparseCategoricalCrossentropy loss function, and the accuracy metric. Next, the model is trained using the training data and validated with the validation data for five epochs to monitor the model's performance.

```
[ ] EPOCHS = 5
    BATCH_SIZE = 32
    LEARNING_RATE = 2e-5
```

```
[ ] train_data = encode_dataset(df_train).shuffle(42).batch(BATCH_SIZE)
    val_data = encode_dataset(df_val).batch(BATCH_SIZE)
    test_data = encode_dataset(df_test).batch(BATCH_SIZE)
```

```
[ ] from transformers import TFBertForSequenceClassification

    model = TFBertForSequenceClassification.from_pretrained(PRETRAINED_MODEL, num_labels=2)

    model.summary()
```

Figure 7. Train a Model Code

Then, the code in Figure 8 is used to load the uncased BERT-base model, evaluate its performance against the test data, and generate predictions and actual labels as the basis for further evaluation metric calculations.

```
optimizer = tf.keras.optimizers.Adam(learning_rate=LEARNING_RATE)
loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
metrics = tf.keras.metrics.SparseCategoricalAccuracy('accuracy')  # Perhatikan penamaan

model.compile(optimizer=optimizer, loss=loss, metrics=[metrics])  # Perbaiki: gunakan 'metrics'

history = model.fit(
    train_data,
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    validation_data=val_data
)
```

```
Epoch 1/5
23/23 [==============================] - 571s 23s/step - loss: 0.6263 - accuracy: 0.6808 - val_loss: 0.5916 - val_accuracy: 0.7070
Epoch 2/5
23/23 [==============================] - 535s 23s/step - loss: 0.5664 - accuracy: 0.7068 - val_loss: 0.5837 - val_accuracy: 0.7070
Epoch 3/5
23/23 [==============================] - 538s 23s/step - loss: 0.4996 - accuracy: 0.7192 - val_loss: 0.5665 - val_accuracy: 0.6688
Epoch 4/5
23/23 [==============================] - 536s 23s/step - loss: 0.3753 - accuracy: 0.8384 - val_loss: 0.5744 - val_accuracy: 0.7452
Epoch 5/5
23/23 [==============================] - 522s 23s/step - loss: 0.1902 - accuracy: 0.9452 - val_loss: 0.8126 - val_accuracy: 0.7134
```

Figure 8. Model Training with 5 Epochs

```
optimizer = tf.keras.optimizers.Adam(learning_rate=LEARNING_RATE)
loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
metrics = tf.keras.metrics.SparseCategoricalAccuracy('accuracy')  # Perhatikan penamaan

model.compile(optimizer=optimizer, loss=loss, metrics=[metrics])  # Perbaiki: gunakan 'metrics'

history = model.fit(
    train_data,
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    validation_data=val_data
)
```

```
Epoch 1/10
23/23 [==============================] - 568s 23s/step - loss: 0.6166 - accuracy: 0.6877 - val_loss: 0.5994 - val_accuracy: 0.7070
Epoch 2/10
23/23 [==============================] - 555s 24s/step - loss: 0.5768 - accuracy: 0.7055 - val_loss: 0.5847 - val_accuracy: 0.7070
Epoch 3/10
23/23 [==============================] - 531s 23s/step - loss: 0.4970 - accuracy: 0.7411 - val_loss: 0.5656 - val_accuracy: 0.6879
Epoch 4/10
23/23 [==============================] - 532s 23s/step - loss: 0.3262 - accuracy: 0.8767 - val_loss: 0.6648 - val_accuracy: 0.7197
Epoch 5/10
23/23 [==============================] - 529s 23s/step - loss: 0.2097 - accuracy: 0.9247 - val_loss: 0.6621 - val_accuracy: 0.7389
Epoch 6/10
23/23 [==============================] - 535s 23s/step - loss: 0.0948 - accuracy: 0.9726 - val_loss: 1.0093 - val_accuracy: 0.6624
Epoch 7/10
23/23 [==============================] - 539s 24s/step - loss: 0.1445 - accuracy: 0.9466 - val_loss: 0.9280 - val_accuracy: 0.7516
Epoch 8/10
23/23 [==============================] - 538s 24s/step - loss: 0.0931 - accuracy: 0.9630 - val_loss: 0.7334 - val_accuracy: 0.7707
Epoch 9/10
23/23 [==============================] - 530s 23s/step - loss: 0.0281 - accuracy: 0.9918 - val_loss: 0.9243 - val_accuracy: 0.7771
Epoch 10/10
23/23 [==============================] - 516s 23s/step - loss: 0.0391 - accuracy: 0.9863 - val_loss: 1.0398 - val_accuracy: 0.7325
```

Figure 9. Model Training with 10 Epochs

```
optimizer = tf.keras.optimizers.Adam(learning_rate=LEARNING_RATE)
loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
metrics = tf.keras.metrics.SparseCategoricalAccuracy('accuracy')  # Perhatikan penamaan

model.compile(optimizer=optimizer, loss=loss, metrics=[metrics])  # Perbaiki: gunakan 'metrics'

history = model.fit(
    train_data,
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    validation_data=val_data
)
```

```
Epoch 1/15
23/23 [==============================] - 650s 26s/step - loss: 0.6024 - accuracy: 0.7068 - val_loss: 0.5793 - val_accuracy: 0.7070
Epoch 2/15
23/23 [==============================] - 597s 26s/step - loss: 0.5206 - accuracy: 0.7247 - val_loss: 0.5447 - val_accuracy: 0.6815
Epoch 3/15
23/23 [==============================] - 595s 26s/step - loss: 0.3527 - accuracy: 0.8671 - val_loss: 0.6003 - val_accuracy: 0.7325
Epoch 4/15
23/23 [==============================] - 592s 26s/step - loss: 0.1604 - accuracy: 0.9493 - val_loss: 0.7767 - val_accuracy: 0.7452
Epoch 5/15
23/23 [==============================] - 593s 26s/step - loss: 0.0812 - accuracy: 0.9795 - val_loss: 0.8088 - val_accuracy: 0.7580
Epoch 6/15
23/23 [==============================] - 588s 26s/step - loss: 0.0555 - accuracy: 0.9849 - val_loss: 0.9890 - val_accuracy: 0.7261
Epoch 7/15
23/23 [==============================] - 583s 25s/step - loss: 0.0700 - accuracy: 0.9767 - val_loss: 0.8461 - val_accuracy: 0.7643
Epoch 8/15
23/23 [==============================] - 583s 25s/step - loss: 0.0397 - accuracy: 0.9890 - val_loss: 0.8982 - val_accuracy: 0.7516
Epoch 9/15
23/23 [==============================] - 578s 25s/step - loss: 0.0244 - accuracy: 0.9890 - val_loss: 1.0731 - val_accuracy: 0.7452
Epoch 10/15
23/23 [==============================] - 580s 25s/step - loss: 0.0095 - accuracy: 0.9973 - val_loss: 1.1481 - val_accuracy: 0.7325
Epoch 11/15
23/23 [==============================] - 572s 25s/step - loss: 0.0295 - accuracy: 0.9890 - val_loss: 1.1991 - val_accuracy: 0.7197
Epoch 12/15
23/23 [==============================] - 580s 25s/step - loss: 0.0250 - accuracy: 0.9918 - val_loss: 1.0640 - val_accuracy: 0.7580
Epoch 13/15
23/23 [==============================] - 583s 25s/step - loss: 0.0119 - accuracy: 0.9959 - val_loss: 1.2428 - val_accuracy: 0.7516
Epoch 14/15
23/23 [==============================] - 587s 26s/step - loss: 0.0080 - accuracy: 0.9973 - val_loss: 1.4801 - val_accuracy: 0.7389
Epoch 15/15
23/23 [==============================] - 581s 25s/step - loss: 0.0123 - accuracy: 0.9973 - val_loss: 1.4646 - val_accuracy: 0.7325
```

Figure 10. Model Training with 15 Epochs

```
optimizer = tf.keras.optimizers.Adam(learning_rate=LEARNING_RATE)
loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
metrics = tf.keras.metrics.SparseCategoricalAccuracy('accuracy')  # Perhatikan penamaan

model.compile(optimizer=optimizer, loss=loss, metrics=[metrics])  # Perbaiki: gunakan 'metrics'

history = model.fit(
    train_data,
    epochs=EPOCHS,
    batch_size=BATCH_SIZE,
    validation_data=val_data
)
```

```
Epoch 1/20
23/23 [==============================] - 610s 25s/step - loss: 0.6109 - accuracy: 0.7041 - val_loss: 0.6025 - val_accuracy: 0.7070
Epoch 2/20
23/23 [==============================] - 556s 24s/step - loss: 0.5584 - accuracy: 0.7205 - val_loss: 0.5970 - val_accuracy: 0.6688
Epoch 3/20
23/23 [==============================] - 548s 24s/step - loss: 0.4640 - accuracy: 0.7781 - val_loss: 0.5988 - val_accuracy: 0.6688
Epoch 4/20
23/23 [==============================] - 543s 24s/step - loss: 0.3309 - accuracy: 0.8863 - val_loss: 0.6073 - val_accuracy: 0.7006
Epoch 5/20
23/23 [==============================] - 560s 24s/step - loss: 0.2077 - accuracy: 0.9329 - val_loss: 0.6583 - val_accuracy: 0.7197
Epoch 6/20
23/23 [==============================] - 541s 24s/step - loss: 0.1260 - accuracy: 0.9699 - val_loss: 0.8582 - val_accuracy: 0.7134
Epoch 7/20
23/23 [==============================] - 553s 24s/step - loss: 0.0882 - accuracy: 0.9781 - val_loss: 0.8335 - val_accuracy: 0.7134
Epoch 8/20
23/23 [==============================] - 564s 25s/step - loss: 0.0584 - accuracy: 0.9904 - val_loss: 0.8400 - val_accuracy: 0.7134
Epoch 9/20
23/23 [==============================] - 551s 24s/step - loss: 0.0395 - accuracy: 0.9932 - val_loss: 0.9120 - val_accuracy: 0.7261
Epoch 10/20
23/23 [==============================] - 552s 24s/step - loss: 0.0365 - accuracy: 0.9904 - val_loss: 1.0749 - val_accuracy: 0.7070
Epoch 11/20
23/23 [==============================] - 542s 24s/step - loss: 0.0352 - accuracy: 0.9904 - val_loss: 1.0877 - val_accuracy: 0.7261
Epoch 12/20
23/23 [==============================] - 541s 24s/step - loss: 0.0341 - accuracy: 0.9890 - val_loss: 1.2162 - val_accuracy: 0.6879
Epoch 13/20
23/23 [==============================] - 553s 24s/step - loss: 0.0800 - accuracy: 0.9740 - val_loss: 1.0782 - val_accuracy: 0.7389
Epoch 14/20
23/23 [==============================] - 546s 24s/step - loss: 0.0333 - accuracy: 0.9918 - val_loss: 1.1309 - val_accuracy: 0.7261
Epoch 15/20
23/23 [==============================] - 552s 24s/step - loss: 0.0303 - accuracy: 0.9890 - val_loss: 1.0162 - val_accuracy: 0.7452
Epoch 16/20
23/23 [==============================] - 546s 24s/step - loss: 0.0237 - accuracy: 0.9959 - val_loss: 1.0364 - val_accuracy: 0.7389
Epoch 17/20
23/23 [==============================] - 541s 24s/step - loss: 0.0149 - accuracy: 0.9973 - val_loss: 1.0683 - val_accuracy: 0.7580
Epoch 18/20
23/23 [==============================] - 544s 24s/step - loss: 0.0139 - accuracy: 0.9945 - val_loss: 1.1909 - val_accuracy: 0.7325
Epoch 19/20
23/23 [==============================] - 550s 24s/step - loss: 0.0113 - accuracy: 0.9973 - val_loss: 1.2155 - val_accuracy: 0.7261
Epoch 20/20
23/23 [==============================] - 537s 23s/step - loss: 0.0072 - accuracy: 0.9973 - val_loss: 1.3055 - val_accuracy: 0.7516
```

Figure 11. Model Training with 20 Epochs

F. Model Evaluation

1. Confusion Matrix with 5 Epochs

Based on Table 9, of the total 157 data tested, approximately 0.79 can be classified correctly. The average F1-score of 0.80 indicates that this model shows a good balance between precision and sensitivity.

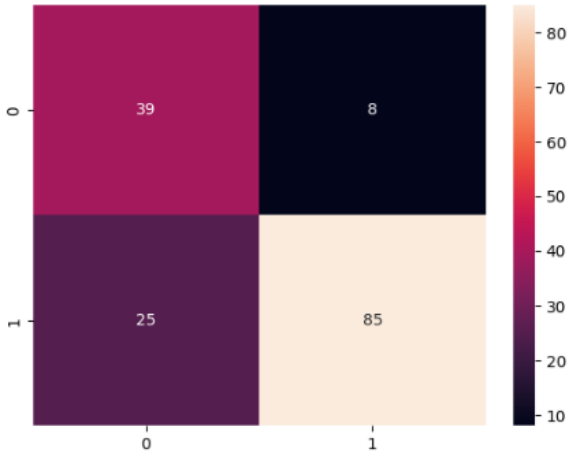

Figure 12. Confusion Matrix for the First Experiment

Table 9. Classification Report for the First Experiment

	Precision	Recall	f1-Score	Support
0	0.61	0.83	0.70	47
1	0.91	0.77	0.84	110
accuracy			0.79	157
macro avg	0.76	0.80	0.77	157
weighted avg	0.82	0.79	0.80	157

2. Confusion Matrix with 10 Epochs

The model shows an overall accuracy rate of 0.80, with a fairly satisfactory average f1-score of 0.80.

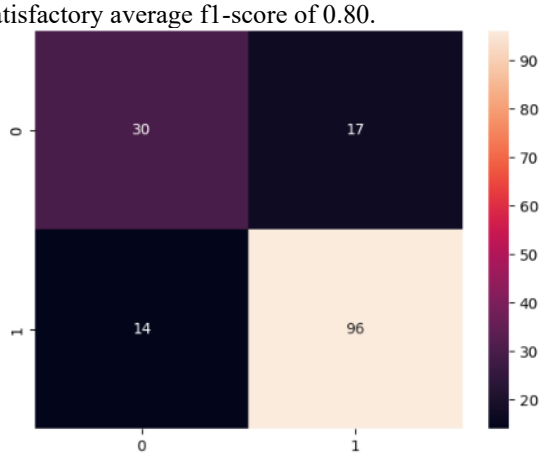

Figure 13. Confusion Matrix for the Second Experiment

Table 10. Classification Report for the Second Experiment

	Precision	Recall	f1-Score	Support
0	0.68	0.64	0.66	47
1	0.85	0.87	0.86	110
accuracy			0.80	157
macro avg	0.77	0.76	0.76	157
weighted avg	0.80	0.80	0.80	157

3. Confusion Matrix with 15 Epochs

The model performed well in identifying the main class (class 1), with a recall value of 0.95 and an F1 score of 0.90. The total accuracy reached 0.85.

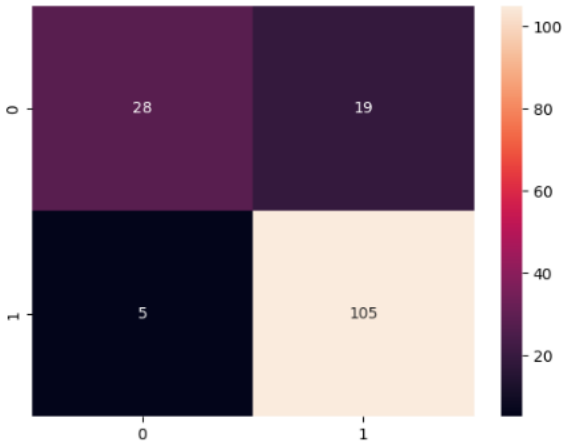

Figure 14. Confusion Matrix for the Third Experiment

Table 11. Classification Report for the Third Experiment

	Precision	Recall	f1-Score	Support
0	0.85	0.60	0.70	47
1	0.85	0.95	0.90	110
accuracy			0.85	157
macro avg	0.85	0.78	0.80	157
weighted avg	0.85	0.85	0.84	157

4. Confusion Matrix with 20 Epochs

Looking at Table 12, the accuracy in the fourth experiment decreased to 0.84, while in the third experiment the accuracy was 0.85.

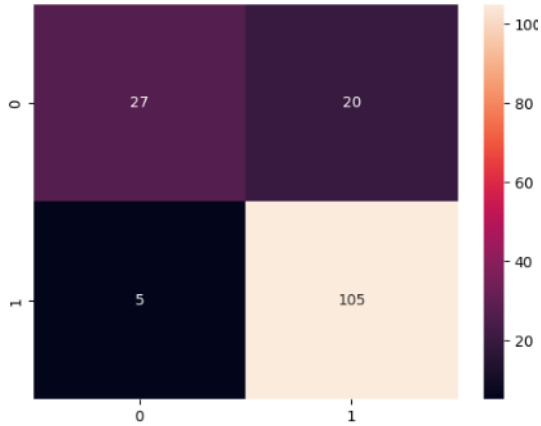

Figure 15. Confusion Matrix for the Fourth Experiment

Table 12. Classification Report for the Fourth Experiment

	Precision	Recall	f1-Score	Support
0	0.84	0.57	0.68	47
1	0.84	0.95	0.89	110
accuracy			0.84	157
macro avg	0.84	0.76	0.79	157
weighted avg	0.84	0.84	0.83	157

Positive Class

Precision

1. Precision calculation with 5 epochs

   $Precision = \frac{TP}{TP+FP} = \frac{85}{85+8} = \frac{85}{93} = 0{,}91$

2. Precision calculation with 10 epochs

   $Precision = \frac{TP}{TP+FP} = \frac{96}{96+17} = \frac{96}{113} = 0{,}85$

3. Precision calculation with 10 epochs

   $Precision = \frac{TP}{TP+FP} = \frac{105}{105+19} = \frac{105}{124} = 0{,}85$

4. Precision calculation with 15 epoch

   $Precision = \frac{TP}{TP+FP} = \frac{105}{105+20} = \frac{105}{125} = 0{,}84$

Recall

1. Recall calculation with 5 epochs

   $Recall = \frac{TP}{TP+FN} = \frac{85}{85+25} = \frac{85}{110} = 0{,}77$

2. Recall calculation with 10 epochs

   $Recall = \frac{TP}{TP+FN} = \frac{96}{96+14} = \frac{96}{110} = 0{,}87$

3. Recall calculation with 15 epochs

   $Recall = \frac{TP}{TP+FN} = \frac{105}{105+5} = \frac{105}{110} = 0{,}95$

4. Recall calculation with 15 epochs

   $Recall = \frac{TP}{TP+FN} = \frac{105}{105+5} = \frac{105}{110} = 0{,}95$

F1-Score

1. F1-score calculation with 5 epochs

   $F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0{,}91 \times 0{,}77}{0{,}91 + 0{,}77}$

   $= \frac{2 \times 0{,}7007}{1{,}68} = \frac{1{,}4014}{1{,}68} = 0{,}84$

2. F1-score calculation with 10 epochs

   $F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0{,}85 \times 0{,}87}{0{,}85 + 0{,}87}$

   $= \frac{2 \times 0{,}7395}{1{,}72} = \frac{1{,}479}{1{,}72} = 0{,}85$

3. F1-score calculation with 15 epochs

   $F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0{,}85 \times 0{,}95}{0{,}85 + 0{,}95}$

   $= \frac{2 \times 0{,}8075}{1{,}80} = \frac{1{,}615}{1{,}80} = 0{,}90$

4. F1-score calculation with 20 epochs

   $F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0{,}84 \times 0{,}95}{0{,}84 + 0{,}95}$

   $= \frac{2 \times 0{,}798}{1{,}79} = \frac{1{,}596}{1{,}79} = 0{,}89$

Negative Class

Precision

1. Precision calculation with 5 epochs

   $Precision = \frac{TN}{TN+FN} = \frac{39}{39+25} = \frac{39}{64} = 0{,}61$

2. Precision calculation with 10 epochs

   $Precision = \frac{TP}{TP+FP} = \frac{30}{30+14} = \frac{30}{44} = 0{,}68$

3. Precision calculation with 10 epochs

   $Precision = \frac{TP}{TP+FP} = \frac{28}{28+5} = \frac{28}{33} = 0{,}85$

4. Precision calculation with 10 epochs

   $Precision = \frac{TP}{TP+FP} = \frac{27}{27+5} = \frac{27}{32} = 0{,}84$

Recall

1. Recall calculation with 5 epochs

   $Recall = \frac{TN}{TN+FP} = \frac{39}{39+8} = \frac{39}{47} = 0{,}83$

2. Recall calculation with 10 epochs

   $Recall = \frac{TN}{TN+FP} = \frac{30}{30+17} = \frac{30}{47} = 0{,}64$

3. Recall calculation with 15 epochs

   $Recall = \frac{TN}{TN+FP} = \frac{28}{28+19} = \frac{28}{47} = 0{,}60$

4. Recall calculation with 15 epochs

   $Recall = \frac{TN}{TN+FP} = \frac{27}{27+20} = \frac{27}{47} = 0{,}57$

F1-Score

1. F1-score calculation with 5 epochs

   $F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0{,}61 \times 0{,}83}{0{,}61 + 0{,}83}$

   $= \frac{2 \times 0{,}5063}{1{,}44} = \frac{1{,}0126}{1{,}44} = 0{,}70$

2. F1-score calculation with 10 epochs

   $F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0{,}68 \times 0{,}64}{0{,}68 + 0{,}64}$

   $= \frac{2 \times 0{,}4352}{1{,}32} = \frac{0{,}8704}{1{,}32} = 0{,}66$

3. F1-score calculation with 15 epochs

   $F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0{,}85 \times 0{,}60}{0{,}85 + 0{,}60}$

   $= \frac{2 \times 0{,}51}{1{,}45} = \frac{1{,}02}{1{,}45} = 0{,}70$

4. F1-score calculation with 20 epochs

   $F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0{,}84 \times 0{,}57}{0{,}84 + 0{,}57}$

   $= \frac{2 \times 0{,}4788}{1{,}41} = \frac{0{,}9576}{1{,}41} = 0{,}68$

Accuracy

1. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{85+39}{85+39+8+25} = \frac{124}{157} = 0{,}79$
2. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{96+30}{96+30+17+14} = \frac{126}{157} = 0{,}80$
3. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{105+28}{105+28+19+5} = \frac{132}{157} = 0{,}85$
4. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{105+27}{105+27+20+5} = \frac{132}{157} = 0{,}84$

G. Visualization

The following is the visualisation result using the WordCloud library, which can be seen in the image below.

Figure 16. Wordcloud

Based on Figure 16, the United States (USA) dominated public discussions or opinions related to video surveillance issues, with 302 opinions out of a total of 428, or more than 70% of the available data. In fact, the total number of tweets collected reached approximately 1,044. However, because many users did not include their location, only 428 users could be identified based on location and analysed further.

Figure 17. 30 Countries That Expressed Their Opinion

IV. Conclusion

This study investigated global public sentiment toward video surveillance technology by analyzing 1,044 English tweets collected from Twitter (X) using the keyword "video surveillance" over the period January 2024–March 2025. After a comprehensive preprocessing pipeline—including cleaning, stopword removal, stemming, lemmatization, tokenization, and manual sentiment labeling—the data were classified into positive and negative categories and modeled using the BERT-base-uncased architecture. The results show that public perception is predominantly supportive: 70.5% (736 tweets) expressed positive sentiment, while 29.5% (308 tweets) conveyed negative views, indicating that the majority of users tend to accept or at least not strongly oppose the implementation of video surveillance in public and private contexts.

From the modeling perspective, the BERT-based classifier demonstrated consistently strong performance across four experimental settings with different numbers of epochs. Model accuracy increased from 0.79 at 5 epochs to 0.80 at 10 epochs and peaked at 0.85 at 15 epochs, before slightly decreasing to 0.84 at 20 epochs due to indications of overfitting. The third experiment (15 epochs) provided the best trade-off between accuracy and generalization, with an F1-score that reflects robust performance particularly on the positive class, while still maintaining reasonable capability in detecting negative sentiment. These results confirm that BERT is effective for capturing nuanced opinion patterns in short social media texts related to surveillance technologies.

The location-based analysis further reveals that discussions on video surveillance are not evenly distributed across countries; a substantial proportion of identifiable tweets originated from the United States, with the remaining tweets spread across several other countries. This suggests that global discourse on video surveillance is strongly influenced by regions with higher social media activity and more mature surveillance infrastructures. These findings can provide valuable input for policymakers, system designers, and stakeholders to better understand how surveillance initiatives are perceived by the public, and to anticipate potential concerns related to security, privacy, and data protection when planning or expanding video surveillance deployments.

Nevertheless, this study has several limitations. The analysis was restricted to English-language tweets, relied on a single keyword ("video surveillance"), and focused on a specific time window, which may not capture all relevant discussions or perspectives from non-English-speaking communities. In addition, sentiment was modeled in a binary manner (positive–negative) without considering neutral or more fine-grained categories such as privacy concern, trust, or perceived usefulness. Future research could address these limitations by incorporating multilingual or cross-lingual models (such as multilingual BERT), expanding the set of surveillance-related keywords, adding neutral or aspect-based sentiment labels, and comparing BERT with other advanced transformer-based architectures. Such extensions would provide a more comprehensive picture of global sentiment and yield deeper insights into how different populations negotiate the trade-offs between safety, surveillance, and civil liberties in the era of intelligent video monitoring.

References

[1] E. D. Carter *et al.*, "Surveillance system integration: reporting the results of a global multicountry survey," *Public Health*, vol. 231, pp. 31–38, 2024, doi: 10.1016/j.puhe.2024.03.004.

[2] E. Aulia, "Analisis User Experience Aplikasi Twitter Menggunakan Metode User Experience Questionnaire (Ueq)," *J. Tek. dan Sci.*, vol. 3, no. 1, pp. 31–39, 2024, doi: 10.56127/jts.v3i1.907.

[3] I. S. K. Idris, Y. A. Mustofa, and I. A. Salihi, "Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Mengunakan Algoritma Support Vector Machine (SVM)," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 1, pp. 32–35, 2023, doi: 10.37905/jjeee.v5i1.16830.

[4] M. D. Deepa and A. Tamilarasi, "Bidirectional Encoder Representations from Transformers (BERT) Language Model for Sentiment Analysis task: Review," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 7, pp. 1708–1721, 2021, [Online]. Available: https://turcomat.org/index.php/turkbilmat/article/view/

3055
[5] S. Adhikesaven, “An Industrial Workplace Alerting and Monitoring Platform to Prevent Workplace Injury and Accidents,” pp. 0–4, 2022.
[6] E. E. UDOFIA, “School Safety Issues Exhibited Among Secondary School Students In Federal Capital Territory (Fct), Abuja, Nigeria: Implications For,” *Int. J. Educ. Humanit. Soc. Sci.*, vol. 7, no. 02, pp. 10–23, 2024, [Online]. Available: http://ijehss.com/
[7] D. L. de S. Ferreira, S. M. de Novaes, and F. G. L. Macedo, “Smart cities and innovation: video surveillance in the public security of Recife, Brazil,” *Cad. Metrópole*, vol. 25, no. 58, pp. 1095–1122, 2023, doi: 10.1590/2236-9996.2023-5814.e.
[8] S. A. Rismawan, Y. Syahidin, P. Piksi, G. Bandung, K. Bandung, and S. Informasi, “Implementasi Website Berita Online Menggunakan Metode Crawling Data Dengan Bahasa Pemrograman Python,” vol. 10, no. 3, 2023.
[9] A. Brijith, “Data Preprocessing for Machine Learning,” no. October, 2023.
[10] L. Hakim, A. Sobri, L. Sunardi, and D. Nurdiansyah, “Prediksi penyakit jantung berbasis mesin learning dengan menggunakan metode k-nn,” vol. 07, no. 02, pp. 14–20, 2025.
[11] A. R. Hanum *et al.*, “Mendeteksi Berita Hoaks Performance Analysis Of The Bert Text Classification Algorithm,” vol. 11, no. 3, pp. 537–546, 2024, doi: 10.25126/jtiik938093.
[12] P. Romadloni, B. Adhi Kusuma, and W. Maulana Baihaqi, “Komparasi Metode Pembelajaran Mesin Untuk Implementasi Pengambilan Keputusan Dalam Menentukan Promosi Jabatan Karyawan,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 2, pp. 622–628, 2022, doi: 10.36040/jati.v6i2.5238.
[13] M. Ferian, R. Akbari, B. Rahayudi, and L. Muflikhah, “Implementasi Deep Learning menggunakan Algoritma EfficientDet untuk Sistem Deteksi Kelayakan Penerima Bantuan Langsung Tunai berdasarkan Citra Rumah di Wilayah Kabupaten Kediri,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, pp. 1817–1825, 2023.