Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9372

Detecting Deepfake Videos Using CNN and GRU Methods: Evaluating Performance on the Celeb-DF(v2) Dataset

Rusdi Afandi, Bedy Purnama*

Informatics Study Program, Faculty of Informatics, Telkom University, Bandung
Email: [1] rusdiafandi@student.telkomuniversity.ac.id , [2,*]bedypurnama@telkomuniversity.ac.id
Email Penulis Korespondensi: bedypurnama@telkomuniversity.ac.id *
Submitted: **24/11/2025**; Accepted: **07/12/2025**; Published: **31/12/2025**

Abstract– The development of deep learning technology has allowed the emergence of the phenomenon of deepfakes, which is the manipulation of digital videos that resemble real videos with a high level of realism. These technologies pose serious threats to privacy, digital security, and the spread of false information. As the quality of deepfake videos increases, the detection of this fake content becomes increasingly challenging. This study aims to design and evaluate a deepfake video detection model using a combination of Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU). CNN is used to extract spatial features from each video frame, while GRU is used to capture the temporal relationships between frames. The dataset used is Celeb-DF(v2), which is a benchmark dataset that contains real videos and high-quality deepfake videos. The CNN-GRU model was trained and tested on the dataset, and its performance was evaluated using accuracy, precision, recall, and F1-score metrics.

Keywords: Deepfake; CNN; GRU; Video detection; Deep learning.

1. INTRODUCTION

The development of artificial intelligence (AI) technology in the current digital era has produced various innovations that have a major impact on human life. AI-based technology is now able to perform complex tasks that previously could only be done by humans, such as facial recognition, voice identification, automated decision-making, and realistic manipulation of digital media [1]. The real implementation of AI's capabilities in media manipulation is the creation of various image and video processing techniques that are increasingly sophisticated and realistic. As this technology matures, various applications have emerged that utilize deep learning algorithms to produce digital media that closely resembles the original, both visually and audio. Among the various forms of digital media manipulation, one that is now increasingly popular and controversial is deepfake technology.

Deepfake is a fake content creation technique that uses deep learning algorithms to replace a person's face, voice, or expression in an image or video in a very realistic way [2]. The content generated by this technology is often difficult to distinguish from the original, even by careful visual observation. The sophistication of deepfake technology carries double implications. On the one hand, deepfakes can be used positively, for example in the film, education, simulation, and entertainment industries to produce creative and innovative content. For example, deepfake technology is used to engineer film scenes involving historical figures or replacing deceased actors, so that it can provide a unique and authentic visual experience for the audience [3]. But on the other hand, this technology also poses negative potential. Deepfakes are vulnerable to being misused for malicious purposes, such as the spread of hoax information or fake news, the falsification of individual identities in certain videos, the spread of fake pornographic content, to sabotage the public image of a person or certain figures for political or criminal purposes [4].

This threat is increasing because deepfake technology has developed very rapidly, both in terms of the quality of the results produced and the ease of access to the software used [5]. Today, a wide range of deepfake maker apps and tools are freely available on the internet, allowing individuals with even minimal technical knowledge to produce high-quality fake videos. As a result, this issue has become a serious concern in the realm of digital security, individual privacy, and the protection of the integrity of information circulating in the global community. Given the growing technology of deepfakes, the challenges in detecting the authenticity of videos or multimedia content are becoming increasingly complex. Deepfake content not only looks visually realistic, but it also continues to improve in its audio-visual quality, making it increasingly difficult for humans to distinguish between real and fake videos. Conventional methods that rely on manual inspections or simple digital forensic techniques are no longer sufficient to meet these challenges.

In response to this need, various AI-based methods and especially deep learning have begun to be widely applied in deepfake detection systems. The deep learning method was chosen because of its ability to process large volumes of data and recognize complex patterns that are difficult to identify with manual approaches or traditional algorithms [6]. One of the deep learning models that is often used in deepfake detection is the Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a model of artificial neural networks that has been shown to be effective in extracting spatial features from image data or video frames automatically [7]. CNN is able to recognize very subtle visual characteristics in an image or video, such as textures, light patterns, shadows, and changes in facial expressions that are often used as markers of deepfake manipulation.

Although CNN is effective in identifying the visual features of each video frame independently, deepfake

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9372

detection is not enough just by analyzing spatial information. Videos are a series of images arranged temporally (sequentially in time), so this temporal aspect is an important factor in identifying video manipulation [9]. Therefore, an approach that is able to capture changes or continuity between frames is needed. One of the techniques that is often used to handle temporal aspects is the recurrent neural network (RNN) method, specifically the Gated Recurrent Unit (GRU).

GRU is an artificial neural network model specifically designed to process sequential data with the ability to efficiently capture temporal dependencies or relationships between frames in a video [10]. The combination of CNN that manages visual (spatial) features and GRU that manages temporal features provides more advantages than using CNN or RNN separately, as it can provide a more comprehensive and accurate analysis in detecting deepfakes in real-time and semi-real-time.

The integration of CNN and GRU methods in the context of deepfake detection is a promising strategic solution because it combines the advantages of both models simultaneously. CNN has a superior ability to capture highly detailed spatial information from every video frame, such as skin texture patterns, facial expressions, color differences, and hidden signs of manipulation [12]. However, because video is time-based data made up of a series of frames, visual information alone is not enough to get a full picture of the authenticity of the content. Deepfakes often leave traces of manipulation in temporal patterns, such as the incompatibility of lip movements with sound or inconsistencies of expression changes over time that are difficult to observe statically significantly. In this case, GRU is a type of Recurrent Neural Network that has a special ability to manage sequential or temporal data. GRU is effectively able to identify and model the relationships between frames in a time sequence, so that it can recognize abnormal or inconsistent patterns in the manipulated video. The GRU works by remembering important information from the previous frames and using them to analyze the next frames, thus being able to recognize the temporal inconsistencies that are one of the hallmarks of deepfake videos.

Through the combination of CNN-GRU, this method is not only able to visually identify manipulations in each frame independently, but also detect temporal inconsistencies simultaneously. This is especially important because sophisticated deepfakes often succeed in tricking visual observations frame-by-frame, but still leave traces of manipulation of movement patterns and inter-frame continuity. This combination is able to provide a higher level of detection accuracy than methods that rely only on visual analysis (CNN only) or temporal analysis (GRU only). Therefore, in this study, the use of the CNN-GRU method will be applied to comprehensively detect deepfake videos by conducting a performance evaluation using the Celeb-DF(v2) dataset, which is a dataset that is notoriously challenging because of its very high quality and resemblance to the original video. This dataset allows for more realistic testing of the capabilities of the CNN-GRU model and provides a clearer picture of the effectiveness of this method under real-world conditions.

The novelty of this study lies in the development of a deepfake detection model that integrates Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) simultaneously to capture spatial and temporal features more comprehensively, while testing its performance on the Celeb-DF(v2) dataset which is known as one of the most challenging datasets in deepfake detection due to its very high level of similarity to the original video. This study not only improves the accuracy of the identification of visual manipulation in each frame through CNN, but also strengthens the detection of temporal inconsistencies through GRU, thus being able to overcome the weaknesses of previous methods that only focused on spatial or temporal aspects. By combining the two approaches and testing them on high-standard data, the study offers a new contribution in the form of a more robust, adaptive and realistic deepfake detection system to meet the challenges of modern video manipulation.

2. RESEARCH METHODOLOGY

This study is an experimental quantitative research that aims to design and evaluate a deep learning model in detecting deepfake videos. The model used is a combination of Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU), which is implemented to analyze spatial and temporal features of video simultaneously. This study uses the Celeb-DF(v2) benchmark dataset as a video data source containing original content and high-quality deepfake manipulation results.

The design of this study focuses on model architecture development, data processing, model training, and performance evaluation using binary classification evaluation metrics. The use of CNN aims to extract visual features from each frame in the video, while the GRU is used to capture sequence patterns between frames that indicate the temporal characteristics of the video. The steps in the design of this research are as follows:

1. Literature Review: Examines theories and previous research related to deepfake detection, CNN and GRU architecture, as well as studies using the Celeb-DF(v2) dataset.
2. Data Collection and Pre-Processing: Downloading the Celeb-DF(v2) dataset, extracting frames from videos, normalizing, and compiling data for model training.
3. Model Architecture Design: Develop a deep learning architecture with a combination of CNN and GRU that can manage spatial-temporal data.
4. Model Training and Testing: Training the model using training data and testing it using data testing to see how it performs.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9372

5. Model Evaluation: Measure model performance based on accuracy, precision, recall, and F1-score metrics to assess the model's effectiveness in detecting deepfake videos.
6. Analysis and Interpretation: Interpret the results of the evaluation and compare it with other relevant approaches to see the strengths and limitations of the model.

2.1 Dataset Collection

In this study, the data source used is Celeb-DF(v2), which is one of the benchmark datasets specifically designed to support the development and evaluation of deepfake detection systems obtained from www.kaggle.com sites. This dataset was chosen because it has high video quality, realism, and challenges that are relevant to real conditions, making it suitable for testing the robustness of deep learning-based detection models. Here's a glimpse of the dataset

Figure 3. Celeb-DF(v2) Dataset at a Glance

Celeb-DF(v2) consists of 6,229 videos divided into 590 original videos from public interviews of celebrities and famous figures on YouTube and 5,639 deepfake videos (manipulated videos), which are generated using the GAN (Generative Adversarial Network) autoencoder algorithm to swap the faces of the characters in the videos for the faces of others. All videos are available in .mp4 format with an average resolution of 480p–720p, and a duration of about 13 seconds per video. The dataset will later be used for CNN-GRU model testing by dividing into two parts, namely an 80% training set to train the model and a 20% test set to evaluate the model's performance

2.2 Preprocessing Data

Data pre-processing is a critical stage in the deep learning pipeline for video deepfake detection. In this study, the Celeb-DF(v2) dataset consisting of the original video and the results of deepfake manipulation had to go through a series of transformation processes to convert the raw video data into a format suitable for the CNN-GRU model input. This preprocessing process aims to extract numerical representations in the form of image frame sequences that can be analyzed spatially by CNN and temporally by GRU.

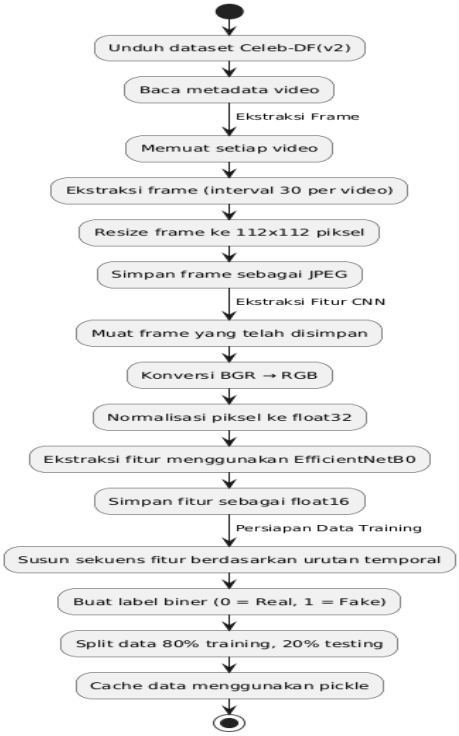

Figure 4. Flowchart Preprocessing Data

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9372

Data pre-processing in this study began by downloading the Celeb-DF(v2) dataset containing a collection of high-quality original and deepfake videos, then extracting the video files into the project directory and creating metadata containing the video path and associated labels. Each video is processed through the frame extraction stage by taking a maximum of 30 frames at uniform intervals, which are then resized to 112 × 112 pixels for uniformity of the EfficientNetB0 model input. Next, spatial feature extraction is carried out where the resized frames are converted from BGR to RGB and normalized by maintaining pixel values in the range of 0-255 using float data type32, different from conventional normalization practices. The feature extraction process using the pre-trained EfficientNetB0 results in a 1280-dimensional feature vector for each frame, which is stored in float16 format for memory optimization, with each feature sequence labeled binary (0 for the original video, 1 for deepfake video). The entire sequence of features that have been extracted and labeled is then compiled in an array format, stored using caching with pickles for computational efficiency, and divided by a ratio of 80% of the training data and 20% of the test data using stratification techniques to maintain a balance of the proportions of real and false data, thus ensuring the readiness of the input data for the training and evaluation of the CNN-GRU model efficiently and consistently.

2.3 Model Architecture

The model architecture used in this study was designed by combining two deep learning approaches, namely Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU). This combination aims to leverage CNN's advantages in extracting spatial features from each video frame, as well as the power of GRU in understanding the temporal dependencies between frames in a single video sequence. The following is the framework of CNN and GRU

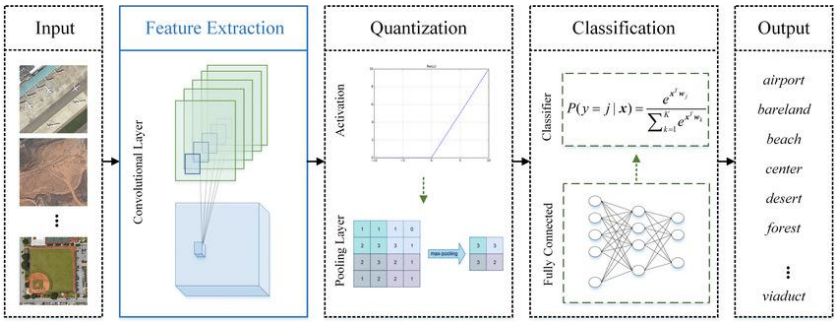

Figure 5. CNN Framework

The image illustrates the basic workflow of the Convolutional Neural Network (CNN) architecture in performing image classification. The process starts from input, in the form of digital images that enter the system, which in the context of this research can be in the form of frames extracted from deepfake videos. After that, the image is processed in the feature extraction stage through the convolutional layer, where the CNN network applies a number of filters to extract important spatial features such as edges, textures, and other visual shapes. The result of this process is feature maps that represent the visual information of the image in a more structured form. Furthermore, in the quantization stage, there are two main processes, namely activation and pooling. Activation functions such as ReLU are applied to improve the non-linearity of the network and determine whether neurons are activated. After that, pooling (usually max pooling) is carried out to reduce the dimensions of the feature map without eliminating the main features, so that the network becomes more efficient and not easily overfitting. The output from this stage is then flattened and goes to the fully connected layer at the classification stage. In the classification stage, the data from the previous layer is processed by a fully connected neural network layer (dense layer), which produces predictive values for each class. The softmax activation function is used to calculate the probability that an input belongs to one of the output classes.

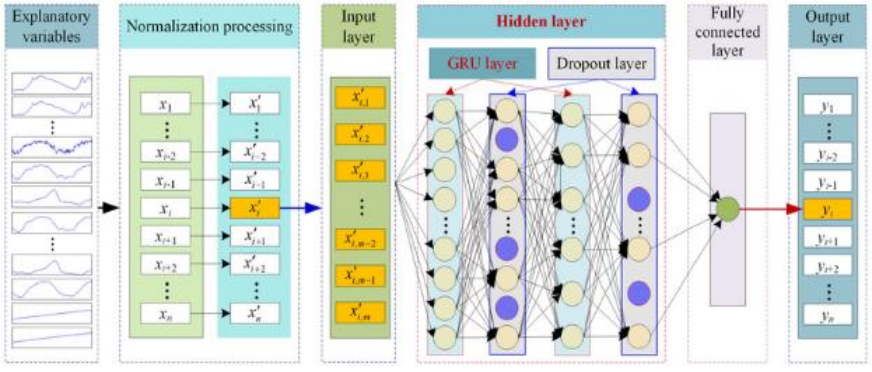

Figure 6. GRU Framework

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9372

The process starts from the far left, which is explanatory variables or input variables in the form of sequential data, such as time series data or feature sequences from video frames. This data can include various representational values of pre-processed video frames. Each variable undergoes a normalization process, aiming to ensure that all inputs have a uniform scale of values, so that the model can learn more stably and efficiently. After the normalization process, the data goes to the input layer, which represents the data in the form of time steps and features. This input is then sent to a hidden layer consisting of multiple GRU layers. At this layer, the GRU processes sequential data and captures temporal dependencies between input elements. The GRU works using two main mechanisms: the update gate and the reset gate which controls which information needs to be retained or forgotten from the previous step. This GRU layer allows the model to recognize patterns of unnatural changes in the data, such as inconsistencies between frames of deepfake videos. Inside the hidden layer there is also a dropout layer, which serves to prevent overfitting by randomly deactivating some neurons during the training process. Furthermore, the output of the GRU and dropout layer are brought together in a fully connected layer (dense layer), which is in charge of processing the final representation before classification. The output from the dense layer is sent to the output layer, which produces a final prediction of a probability value for classification, as in the case of a real or fake deepfake detection.

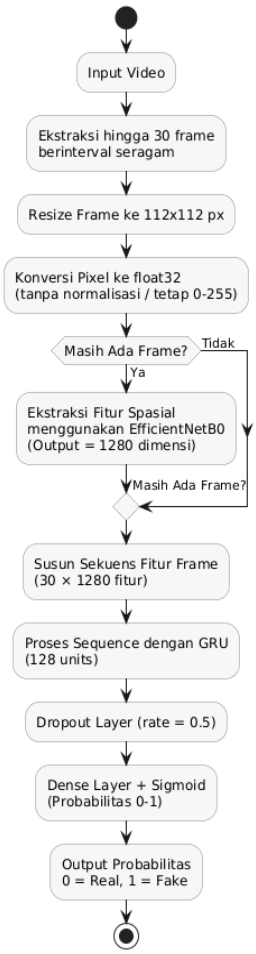

Figure 7. Architecture Flowchart

The CNN-GRU model architecture flowchart for deepfake video detection begins with video input processed through frame extraction to a maximum of 30 frames at uniform intervals. The frames were then resized to a fixed size of 112×112 pixels to match the EfficientNetB0 model input. In contrast to conventional normalization practices, pixel values are maintained in the range of 0-255 with the float data type 32 without division by 255. Each processed frame is then extracted its spatial features using a pre-trained *Convolutional Neural Network* (CNN) based on EfficientNetB0, resulting in a feature vector with dimensions of 1280 for each frame. The extraction process of this feature is done sequentially for all frames in the video. All feature extraction results from each frame are then compiled into a sequence of temporal features that represent the visual chronology of the video. This feature sequence is fed into the Gated Recurrent Unit (GRU) layer with 128units to study temporal patterns and inconsistencies between frames, such as anomalies in facial movements or expression inconsistencies that characterize deepfake videos. The output from the GRU is then processed through the dropout layer at a rate of 0.5 to prevent overfitting, and passed to the last dense layer with a sigmoid activation function that results in a probability value between 0 to 1. This output value indicates the probability that the video falls into the category of real (real) or fake (deepfake), where a value close to 1 indicates a tendency to be a deepfake video.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9372

3. RESULTS AND DISCUSSION

3.1 Experimental results

In this section, the results of training and testing of the CNN-GRU model that have been built are presented. Results include training accuracy graphs, confusion matrix, and model performance evaluations based on binary classification metrics at multiple threshold values. The analysis was conducted to understand the characteristics of the model in distinguishing real and fake videos in the Celeb-DF(v2) dataset.

3.1.1 Accuracy & Loss Graph

The following figure shows a graph of *training accuracy* and *validation accuracy* over the 20 epoch of the training process.

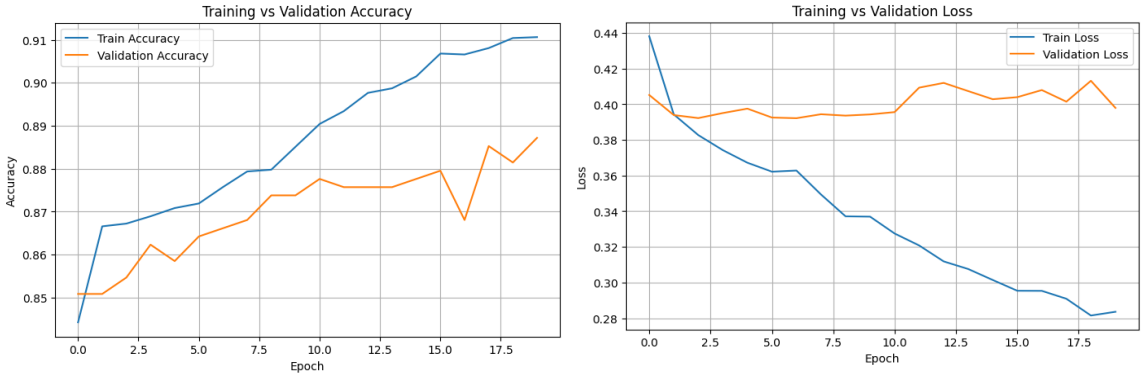

Figure 8. Accuracy & Loss Graph

Training and validation accuracy graphs show that model performance improved consistently over 20 epochs. The accuracy of the training rose from about 0.84 in the first epoch to 0.912 in the 20th epoch, while the validation accuracy increased from 0.852 to about 0.889. The difference in accuracy between the training and validation data was relatively small, suggesting that the model was not overfitting because both curves had a similar and stable increment pattern in the range of 0.87–0.89. In addition, the graph shows that the model began to reach a point of convergence in the 15th to 20th epoch, where the increase in accuracy was no longer significant after the 17th epoch, so the number of 20 epochs was appropriate for the training process.

3.1.2 Evaluation Testing Results

The evaluation of model performance was carried out using test data of 20% of the total dataset. The test is focused on several key metrics, namely Accuracy, Precision, Recall, F1-Score, and Confusion Matrix. In addition, the model was also tested on several variations of threshold values (0.25, 0.30, 0.35, 0.40, 0.45, and 0.50) to analyze performance stability.

3.1.2.1 Confusion Matrix

The confusion matrix shown in Figure *"coffesien matrix"* shows the result of the model classification at a threshold of 0.50. The model managed to classify fake videos very well, marked by a True Positive (TP) value of 1125 and a False Negative (FN) of only 3. However, the model seems quite aggressive in detecting videos as fake, as there are 141 real videos that are wrongly classified as fake (False Positive/FP). Meanwhile, True Negative (TN) was recorded as many as 37. This pattern is common in deepfake datasets because the number of fake videos is much more dominant, so the model tends to focus on detecting the majority class.

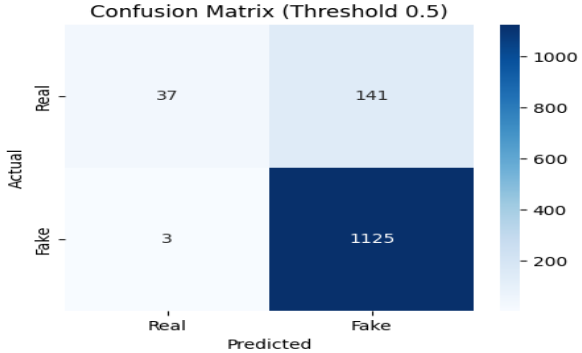

Figure 9. Confusion Matrix

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9372

3.1.2.2 Evaluation Metrics at Various Thresholds

Model testing at multiple thresholds showed a stable performance pattern across a range of values. The *recall*, *accurcy*, *pressiion*, and *trasehold* graphs depict the metric frequency based on threshold variations.

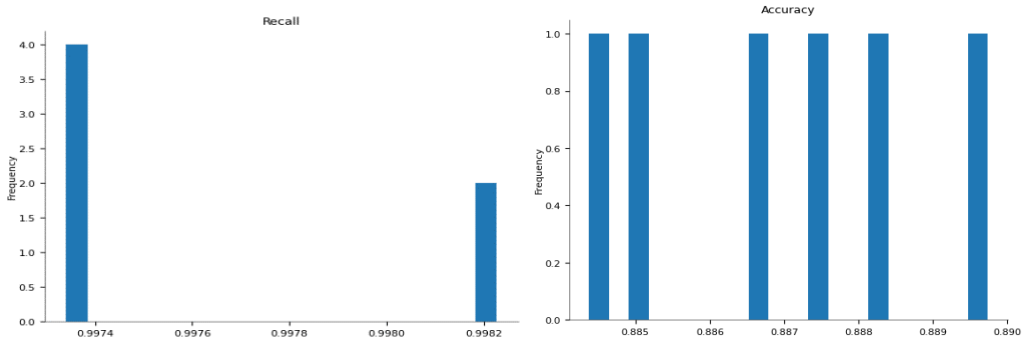

Figure 10. Recall and Accurcy

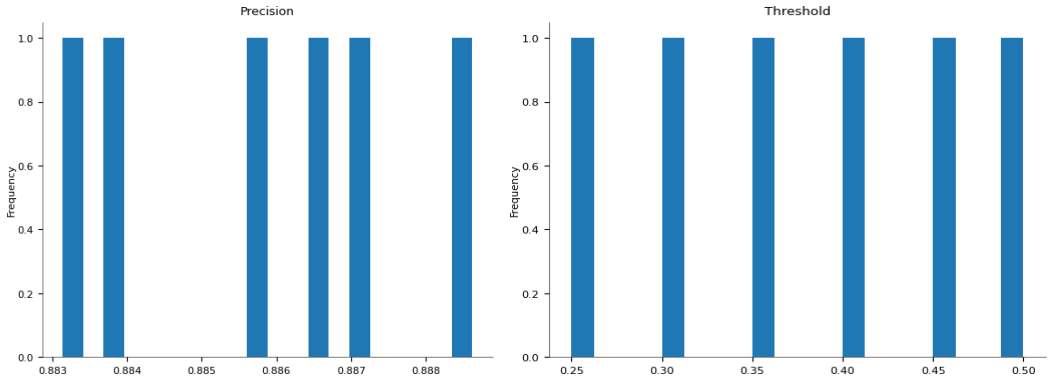

Figure 12. Precision and Trasehold

From the recall chart, it can be seen that the recall value is very high, ranging from 0.9974 to 0.9982, indicating that the model is able to detect almost all fake videos consistently and minimize False Negatives. The accuracy value is in the range of 0.885 to 0.889, relatively stable at all thresholds, while the precision is in the range of 0.883 to 0.888. This slightly lower precision is due to a fairly high number of False Positives. Based on the threshold graph, the model performs best when the threshold is in the range of 0.35–0.45, because a threshold that is too high such as 0.50 tends to increase the False Positive value.

3.2 Evaluation Metrics Summary Table

The following table summarizes the value of the model evaluation metrics in the threshold range of 0.25 to 0.50, based on the interpreted graph pattern.

Table 1. Model Evaluation Results on Data Testing

Metric	Score (Average)
Accuracy	0.885 – 0.889
Accuracy	0.883 – 0.888
Recall	0.9974 – 0.9982
F1 Score	0.936 – 0.942 (estimated)

The F1-Score is calculated based on the Precision and Recall values, and shows that overall the model performs very well in detecting deepfake videos, with high sensitivity and stable consistency at various threshold values.

3.3 Analysis of Experimental Results

The results of the experiments showed that the CNN–GRU model had excellent deepfake detection capabilities, characterized by near-perfect recall values. However, a fairly high number of False Positives indicates that the model is still sensitive to real videos that have low quality, unstable lighting, or motion blur. These conditions often make real videos look like deepfakes so the model misclassifies them. Meanwhile, False Negatives are very few and only occur in high-quality deepfakes, where face blending and motion synchronization are so subtle that they are difficult to distinguish between CNN and GRU. Video quality is proven to have a great influence on detection results. Low-quality videos, both real and fake, tend to have strong visual artifacts that

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9372

affect the model's confidence. In contrast, high-quality deepfakes are more difficult to detect because of their lack of spatial and temporal distortion. Architecturally, CNN plays an effective role in capturing spatial features such as skin texture, facial contours, and traces of subtle manipulation on individual frames. GRU complements this process by studying temporal changes between frames, including lip movement patterns, blinking, and expression consistency. The collaboration between the two allows the model to recognize deepfakes with high accuracy while still showing weaknesses in real videos that degrade in quality.

3.4 Implementation of a Web-Based Deepfake Detection System

3.4.1 CNN–GRU Model Integration on the Flask Framework

The integration of the CNN–GRU model into the Flask framework is done by loading the model that has been stored in .h5 format at the time the server is running. This process involves invoking EfficientNetB0 as a spatial feature extractor and GRU as a temporal sequence modeler in a single inference flow. Before the prediction is made, the video is processed through an internal preprocessing stage, which is taking a maximum of 30 frames, resizing to a size of 112×112 pixels, converting the color format from BGR to RGB, and maintaining the pixel value in the float32 type in the range of 0–255. The processed frames were then extracted using frozen CNNs, resulting in a dimensional feature vector (30, 1280) which was then fed into the GRU to study temporal patterns. The final result is given through a sigmoid layer that results in the probability of whether the video is a deepfake or real. With this integration, the web system is able to infer directly without the need for a retraining process, so the model can be used efficiently and responsively on Flask-based applications.

3.4.2 System Inference Flow

3.4.2.1 Video Appeal

Users upload videos in MP4, AVI, MOV, or MKV formats with a maximum size of 100 MB, The uploading interface is shown as in the following image:

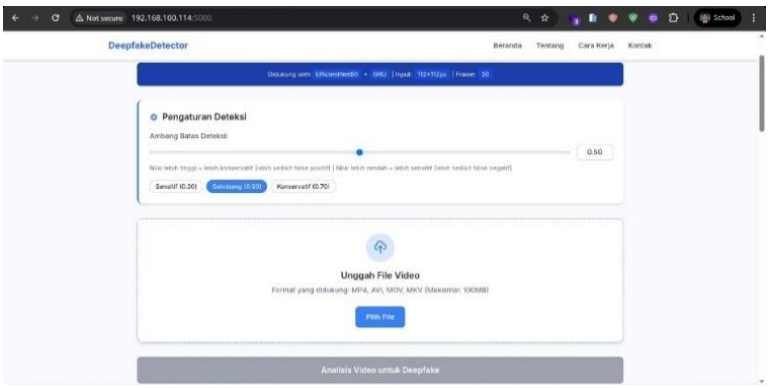

Figure 14. Video upload view

3.4.2.2 Detection Threshold Settings

The system provides sliders and several preset buttons to set the model's sensitivity level, namely *Sensitive* (threshold 0.30) which makes it easier for the model to detect video as fake, *Balanced* (threshold 0.50) as the default setting, and *Conservative* (threshold 0.70) which aims to reduce False Positives. This threshold setting allows the results of classification to adjust user preferences and needs to be displayed as shown in the following image:

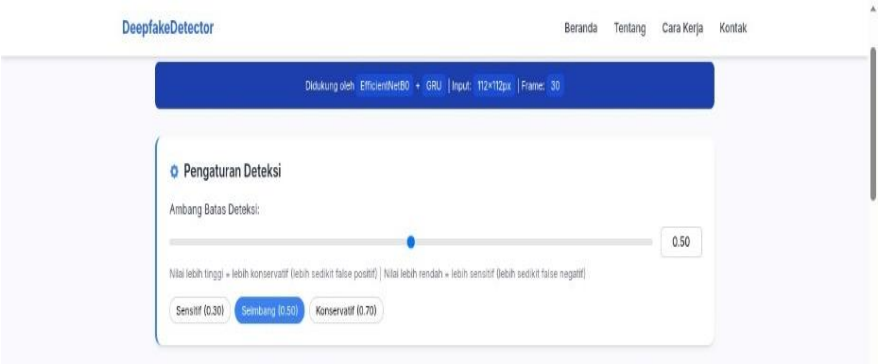

Figure 15. Threshold setting view

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9372

3.4.2.3 Extract frame

Flask receives video input from the user, then extracts a maximum of 30 frames at uniform intervals to keep the temporal representation consistent. Each frame taken is resized to 112×112 pixels and converted to *float32 format* without the 1/255 normalization process, so that the pixel value remains in its original range. This process ensures that the incoming data matches the format used during the model training.

3.4.2.4 CNN Feature Extraction

Each frame is processed by EfficientNetB0 to generate a 1280-dimensional feature vector, All features are then constructed as a dimensional sequence (30, 1280).

3.4.2.5 Temporal Modeling and Prediction

At the temporal modeling and prediction stages, the CNN extracted feature sequences are fed into the GRU model for processing. GRU analyzes the pattern of change between frames so that it is able to recognize the inconsistencies of movement that are characteristic of deepfakes. Once the temporal information is learned, the GRU output is passed to the Dense layer with Sigmoid activation that generates a probability score, determining whether the video is classified as a deepfake or real.

3.5 Class Determination (Real/Fake)

The probability result is then compared to the user-selected threshold.

3.5.2 System Detection Output Display

The system displays the results of the analysis in two categories: deepfake or authentic, according to the probability generated by the model.

3.5.2.1 Deepfake Results Detected

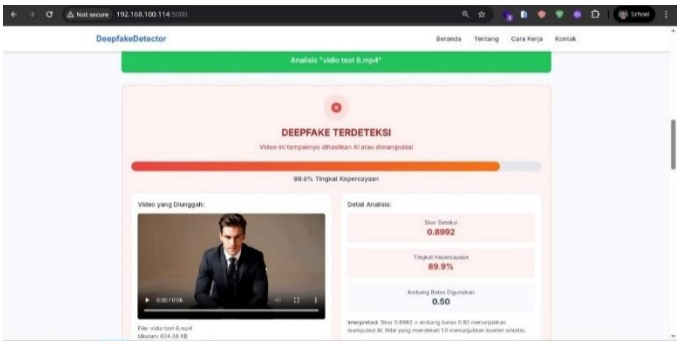

Figure 16. Deepfake results display

The system displays the prediction results in the form of *a Detection Score*, for example 0.6810, which is then converted to *Confidence* of 68.1%. Threshold *Used* information, such as 0.50, is also displayed to indicate the decision limit used during classification. In addition, users can view a *preview* of the analyzed video. The interpretation of the results is given clearly, for example "Score 0.6810 > threshold 0.50 indicates AI manipulation. Values closer to 1.0 suggest synthetic content." The interface uses a red color theme to emphasize the indication of danger in content detected as deepfakes.

3.5.2.2 Authentic Video Results

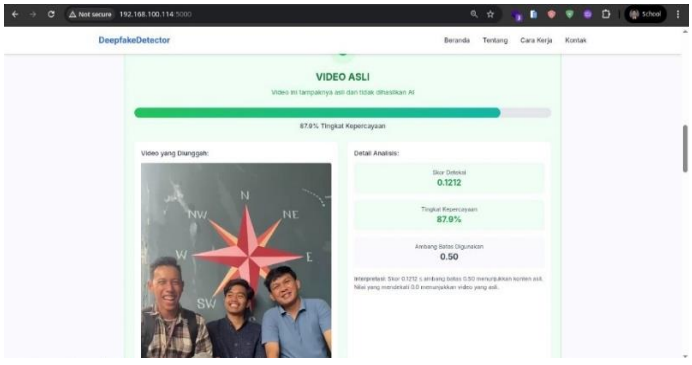

Figure 17. Display of original video results

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9372

The system displays the results of the analysis in the form of *a Detection Score*, such as 0.1561, followed by *a Confidence* of 84.4% and *Threshold Used* information, such as 0.30, to indicate the limits of the decision used. Users can also preview processed videos. Interpretation of the results is provided directly, for example "Score 0.1561 ≤ threshold 0.30 indicates authentic content. Values closer to 0.0 suggest genuine video." The interface display uses shades of green as a marker that the video is not detected as a deepfake.

3.5.2.3 System Inference Performance

Inference performance testing was performed on laptops with local CPU or GPU support, and the results showed fairly efficient processing times. The inference process consists of several stages, namely frame extraction which takes about 0.5–1.5 seconds, CNN feature extraction for 1–2 seconds, and GRU prediction which takes very fast at less than 0.1 seconds. Overall, the average time it takes to process a single video ranges from 2–4 seconds. The system can run well using CPUs alone, although performance improves when CUDA-based GPUs are used, especially at the most computationally intensive CNN feature extraction stages. In terms of responsiveness, the Flask app is able to provide fast feedback because the model has been fully loaded into memory, so the main wait time only comes from the video upload process, not from the model's inference.

3.5.2.4 System Implementation Conclusion

The web system is capable of real-time deepfake detection with a processing time of just a few seconds.
The system provides threshold settings so that the sensitivity of the model can be adjusted based on the user's needs. The output results are presented in an informative and easy-to-understand manner, including:

1. *Preview* video,
2. *detection score*,
3. *Confidence level*.
4. interpretation of classification.

The system successfully integrates the CNN–GRU pipeline end-to-end, from frame extraction, feature extraction, to prediction stage.

4. CONCLUSION

Conclusions drawn from the entire research process and describe the level of achievement of the objectives of the Final Project as stated in Chapter 1. This research successfully designed and built a deepfake detection system based on the CNN-GRU architecture that is able to process spatial and temporal information in an integrated manner. EfficientNetB0 plays a role in extracting the visual features of each frame, while GRU models the relationships between frames to detect inconsistencies that characterize video manipulation. The preprocessing pipeline used from frame extraction, resizing, to feature sequence formation runs optimally and according to model needs. The CNN–GRU model built showed good performance on the Celeb-DF(v2) dataset, with stable evaluation metrics at various thresholds and a low misclassification rate. The implementation of the Flask-based web application was also successful, allowing the system to make inferences quickly and display the detection results informatively. Thus, the research goal of building a deepfake detection system based on a spatio-temporal approach can be said to have been achieved.

REFERENCES

[1] M. Farwati, I. Talitha Salsabila, K. Raihanun Navira, T. Sutabri, en U. Bina Darma Palembang, "Analysis of the influence of artificial intelligence (AI) technology in daily life", *J. Sist. Inform. and Management*, vol 11, no 1, bll 41–42, 2023,doi: 10.47024/js.v11i1.563.

[2] Y. Patel *et al.*, "Deepfake Generation and Detection: Case Study and Challenges", *IEEE Access*, vol 11, December 2023, bll 143296–143323, 2023, doi: 10.1109/ACCESS.2023.3342107.

[3] A. A. Owie, E. Wibowo, and A. F. Yogyakarta, "THE USE OF ARTIFICIAL INTELLIGENCE IN THE MAKING OF FILM PROPERTY DESIGN", bll 130–138.

[4] M. Yudha, "Legal protection for victims of deepfake use in pornographic crimes", vol 2, no 1, 2025.

[5] T. Zhafira, "The role of international law in addressing the threat of deepfake technology to global security", November no, 2024, doi: 10.13140/RG.2.2.31626.68806.

[6] H. F. Fadhilah, "Advantages and Challenges in the Use of Computer Vision for the Diagnosis of Pediatric Pneumonia: A Systematic Review", vol 5, no 1, 2024, doi: 10.7454/bikfokes.v5i1.1077.

[7] Y. D. Zhang, S. C. Satapathy, S. Liu, and G. R. Li, "A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis", *Mach. Vis. Appl.*, vol 32, no 1, bll 1–13, 2021, doi: 10.1007/s00138-020-01128-8.

[8] W. Nagy, H. Alsalamah, M. Hassan, and E. Moahamed, "AUTO-HAR: An adaptive human activity recognition framework using an automated CNN architecture design", *Hell*, vol 9, bl e13636, Feb 2023, doi: 10.1016/j.heliyon.2023.e13636.

[9] Y. Xu, J. Liang, G. Jia, Z. Yang, Y. Zhang, and R. He, "TALL: Thumbnail Layout for Deepfake Video Detection", *Proc. IEEE Int. Conf. Comput. Vis.*, BLL 22601–22611, 2023, doi: 10.1109/ICCV51070.2023.02071.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9372

[10] S. AYDIN, Ö. ÇAYLI, V. KILIÇ, and A. ONAN, "Sequence-to-Sequence Video Captioning with Residual Connected Gated Recurrent Units", *Eur. J. Sci. Technol.*, no. 35, bll 380–386, 2022, doi: 10.31590/ejosat.1071835.

[11] H. Namdari, A. Haghighi, and S. M. Ashrafi, "Short-term urban water demand forecasting; application of 1D convolutional neural network (1D CNN) in comparison with different deep learning schemes", *Stoch. Approximately. Res. Risk Assess.*, bll 1–16, Sep 2023, doi: 10.1007/s00477-023-02565-3.

[12] D. Aldiani, G. Dwilestari, H. Susana, R. Hamonangan, en D. Pratama, "Implementation of CNN Algorithm in Facial Recognition-Based Attendance System", *J. Inform. Polynesian*, vol 10, no 2, bll 197–202, 2024, doi: 10.33795/jip.v10i2.4852.

[13] N. R. Muntiari and K. H. Hanif, "Classification of Breast Cancer Disease Using Comparative Machine Learning Algorithms," *J. Computing Science. and Technology.*, vol. 3, no. 1, pp. 1–6, 2022, doi: 10.35960/ikomti.v3i1.766.

[14] N. L. P. C. Savitri, R. A. Rahman, R. Venyutzky, and N. A. Rakhmawati, "Analysis of Sentiment Classification of Online Schools on Twitter Using Supervised Machine Learning," *J. Tek. Inform. and Sist. Inf.*, vol. 7, no. 1, pp. 47–58, 2021, doi: 10.28932/jutisi.v7i1.3216.

[15] A. Ouadah, L. Zemmouchi-Ghomari, and N. Salhi, "Selecting an appropriate supervised machine learning algorithm for predictive maintenance," *Int. J. Adv. Manuf. Technol.*, vol. 119, Apr. 2022, doi: 10.1007/s00170-021-08551-9.

[16] Muttaqin, D. Fernando, and S. Sulastriani, "Implementation of Unsupervised Learning on the Physical Value of Uniformity of State Police Schools with the Clustering Analysis Method," *J. PROSISKO*, vol. 10, no. 1, 2023.

[17] J. Ghahremani-Nahr, H. Nozari, and M. E. Sadeghi, "Artificial intelligence and Machine Learning for Real-world problems (A survey)," *Int. J. Innov. Eng.*, vol. 1, pp. 38–47, Oct. 2021, doi: 10.59615/ijie.1.3.38.

[18] A. Baradja and T. I. Tjendrowasono, "Application of Deep Reinforcement Q-Learning for Automated Forex Trading Predictions," *J. Rekayasa Sist. Inf. and Technology.*, vol. 1, no. 3, pp. 190–198, 2024, doi: 10.59407/jrsit.v1i3.519.

[19] E. Baccour *et al.*, *Pervasive AI for IoT Applications: Resource-efficient Distributed Artificial Intelligence*. 2021. doi: 10.48550/arXiv.2105.01798.

[20] M. R. S. Alfarizi, M. Z. Al-farish, M. Taufiqurrahman, G. Ardiansah, and M. Elgar, "The Use of Python as a Programming Language for Machine Learning and Deep Learning," *Works of Ilm. Mhs. Bertauhid (KARIMAH TAUHID)*, vol. 2, no. 1, pp. 1–6, 2023.

[21] M. F. Naufal and S. F. Kusuma, "Comparison Analysis of Machine Learning and Deep Learning Algorithms for Image Classification of Indonesian Language Signing Systems (Sibi)," *Jtiik*, vol. 10, no. 4, pp. 873–882, 2023, doi: 10.25126/jtiik.2023106828.

[22] A. M. Alhassan and W. M. N. W. Zainon, "Brain tumor classification in magnetic resonance image using hard swish-based RELU activation function-convolutional neural network," *Neural Computing. Appl.*, vol. 33, no. 15, pp. 9075–9087, 2021, doi: 10.1007/s00521-020-05671-3.

[23] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, "A Comprehensive Survey of Loss Functions in Machine Learning," *Ann. Data Sci.*, vol. 9, no. 2, pp. 187–212, 2022, doi: 10.1007/s40745-020-00253-5.