International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

LeakCare: An AI-Driven Platform for Detecting Deepfake and Illegally Filmed Content with Automated Removal Request Generation

Seohyun Lee[1]; Sujin Kim[2]; Jimin Nam[3]; Minseo Park[4]; Seungjae Lee[5]*

Department of Computer Science, Sunmoon University, Korea
[1] seo7316@sunmoon.ac.kr; [2] tnwls78001@sunmoon.ac.kr; [3] nam10244@sunmoon.ac.kr;
[4] alstjrdk2813@sunmoon.ac.kr; [5]* leeko@sunmoon.ac.kr (Corresponding Author)

DOI: https://doi.org/10.47760/ijcsmc.2026.v15i07.003

Abstract: This paper presents LeakCare, an AI-based detection and automated removal request platform designed to support victims of deepfake synthesis and illegally filmed content leakage. Once a victim registers their facial photographs, the system automatically crawls suspect URLs, performs ArcFace-based facial similarity comparison and ConvNeXt-based deepfake detection, generates a PDF evidence report, and produces a legally-oriented removal request letter using the Claude API. The system adopts a four-layer architecture consisting of a React-based frontend (FE), a FastAPI backend (BE), a Playwright-based web evidence collection engine (SYS), and an AI module built on insightface and PyTorch. During face registration, a BiSeNet-based face parsing model verifies whether the eyes, nose, and mouth are unobstructed. During detection, parallel image analysis is performed using asynchronous semaphores to enhance processing throughput. LeakCare overcomes the limitations of conventional passive reporting mechanisms and provides victims with a proactive, automated response tool.

Keywords: Deepfake Detection, Face Recognition, ArcFace, Illegal Filming, LLM, Removal Request Automation

I. INTRODUCTION

The proliferation of online platforms and advances in deepfake synthesis technology have fundamentally transformed the landscape of digital sexual crimes. According to the 2024 Digital Sexual Crime Victim Support Report, the total number of reported support cases reached 16,833 in 2024, a 15.6% increase from 14,565 in 2023.

Notably, deepfake-related synthesis and editing cases surged by 227.2% year-over-year. Once distributed, illegal footage and deepfake-synthesized content spread rapidly through continuous replication. By the time victims become aware of the situation, the content has often reached a state where complete retrieval is nearly impossible. Furthermore, 95.4% of sites hosting such content operate on overseas servers, creating significant legal and linguistic barriers for individual victims attempting to request removal. Consequently, victims are often forced into secondary victimization by directly searching for and viewing their own leaked content.

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

Prior studies have largely focused on improving individual technical components, such as enhancing deepfake detection performance or developing methodologies for digital forensics. However, an integrated, victim-centered response framework that consolidates detection results into legally admissible evidence and automates the subsequent removal request process has yet to be proposed. Under current support systems, the burden of evidence collection, drafting removal requests, and filing reports falls entirely on the victim, while administrative response capacities struggle to keep pace with the rapidly growing volume of cases.

To address these limitations, this paper proposes LeakCare, an integrated detection and response platform for deepfake and illegal footage leakage. By simply submitting a target URL, LeakCare automatically processes the entire response pipeline: detecting leaked content through InsightFace ArcFace-based facial similarity analysis and ConvNeXt-based deepfake classification, collecting forensic evidence and server metadata via Playwright, generating PDF evidence reports using ReportLab, and producing localized removal request notices through Claude API integration.

The key contributions of this work are as follows:

- An end-to-end victim-centered automated response pipeline spanning from detection to removal request submission
- An adaptive dual-threshold detection approach combining a face similarity threshold and a deepfake classification threshold to enable differentiated detection of standard illegal footage and deepfake-synthesized content
- A six-stage face registration quality verification system for precise and reliable victim identity representation

Through these contributions, LeakCare provides an environment in which victims can initiate the response procedure without the trauma of directly searching for or viewing the leaked material.

II. RELATED WORK

A. Deepfake Detection Technology Trends

Deepfake refers to technology that uses generative AI such as GAN (Generative Adversarial Network) to produce synthetic images or videos that appear realistic. Early deepfake detection research primarily relied on binary classification approaches based on single frames; however, as generation technology advanced and visual artifacts became less noticeable, the difficulty of detection increased significantly. Park et al. analyzed trends in CNN-based deepfake video detection, demonstrating that pipelines involving facial region extraction followed by classification represented the mainstream approach in early research. Subsequently, Kim et al. surveyed the latest trends in data-driven deepfake detection techniques, summarizing the characteristics and limitations of various detection algorithms [1], while Hwang et al. proposed the use of time-series learning across video frames to demonstrate the feasibility of video-level deepfake detection. Furthermore, Jeong et al. proposed a Vision Transformer (ViT)-based detection model leveraging multi-scale features, showing that simultaneously learning facial characteristics at various resolutions could improve accuracy [5], and Kim et al. highlighted performance bias arising from inconsistencies in validation methodologies for detection models, emphasizing the need for a standardized evaluation framework [6]. However, these existing studies focus primarily on improving the performance of detection models themselves, and lack an integrated perspective that connects detection results to actual victim response.

B. Digital Sexual Crime Response Systems

Research on victim response systems has also been actively conducted alongside the rapid increase in digital sexual crimes. According to the 2024 Digital Sexual Crime Victim Support Report, the number of deepfake synthesis and editing cases surged by 227.2% compared to the previous year, with the total number of support cases reaching 332,341, indicating that administrative response capacity has reached saturation. Furthermore,

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

since 95.4% of illegal sites operate on overseas servers, domestic administrative blocking measures alone are clearly insufficient. Shin analyzed the reality of digital sexual crimes utilizing deepfake technology from a legal and institutional perspective, pointing out the structural problem that victims must manually handle every step from detection to removal requests [3]. Jo examined the reality of deepfake sexual crimes through analysis of court rulings and emphasized the need for victim-centered responses [7], while Ko et al. comprehensively reviewed technical and legal countermeasures against deepfake sexual crimes [8]. Song et al. studied AI-based detection and punishment measures for digital sexual crimes [4], but integrated response platforms implemented as actual services remain difficult to find. Since existing studies address detection, evidence collection, and removal requests as separate and independent procedures, the need for an integrated platform where victims can handle the entire process through a single interface has become increasingly apparent. LeakCare, proposed in this study, aims to fill this gap by integrating everything from detection to evidence report generation and multilingual removal requests into a single platform.

III. PROPOSED SYSTEM

A. System Architecture

The LeakCare system proposed in this study adopts a three-tier architecture consisting of a frontend, a backend, and an AI and system module. Fig. 1 illustrates the overall structure of the LeakCare system.

Users register facial images and input target URLs for detection through a React 19-based SPA (Single Page Application) frontend. Requests are forwarded to a Python FastAPI-based RESTful API backend server, which handles security processing including user authentication via JWT and bcrypt, Pydantic schema validation, and file forgery prevention. During the face registration phase, an Average Embedding is generated from up to five images, and data quality is ensured through BiSeNet-based occlusion verification that measures pixel-level occlusion of the eyes, nose, and mouth. Upon receiving an analysis request from the backend, the AI and system module collects images and metadata from the target webpage through Playwright-based dynamic crawling, extracts 512-dimensional facial embedding vectors using the InsightFace (buffalo_l) model, and computes cosine similarity scores. Completed analysis results are stored in MongoDB, while the frontend monitors task status in real time through HTTP polling at five-second intervals. Through this series of processes, LeakCare is designed to handle the entire workflow from detection requests to final report generation within a single dashboard.

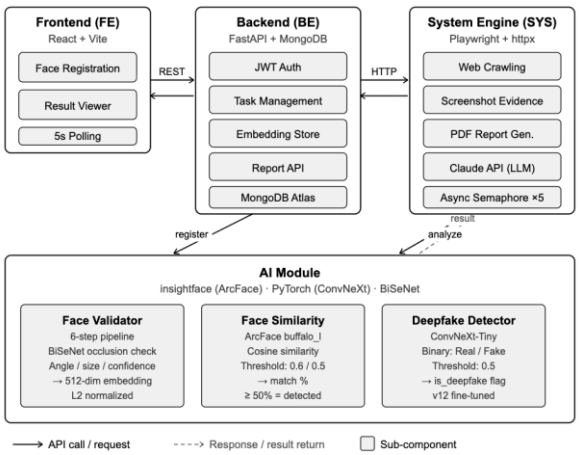


Fig. 1 LeakCare System Architecture Diagram

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

TABLE I
TECHNOLOGY STACK OF LEAKCARE SYSTEM

Layer	Technology
FE	React 19, Vite 7, React Router
BE	FastAPI, MongoDB Atlas, JWT, Motor
AI	insightface (ArcFace), PyTorch (ConvNeXt, BiSeNet)
SYS	Playwright, httpx, ReportLab, Anthropic API (Claude)

Fig. 2 End-to-End Analysis Sequence Diagram

B. Face Registration and Validation

The face registration and validation stage converts a user-uploaded facial image into a single normalized identity embedding. Because the registered embedding is later used as the reference against which faces collected from web content are compared, the quality of the registration image directly affects the overall detection performance of the system. Accordingly, before an image is registered, the system is designed to sequentially verify face detectability, the number of faces, face size, face orientation, facial occlusion, and detection confidence, and an embedding is generated only for images that pass all stages. Table 2 summarizes the six-stage validation procedure.

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

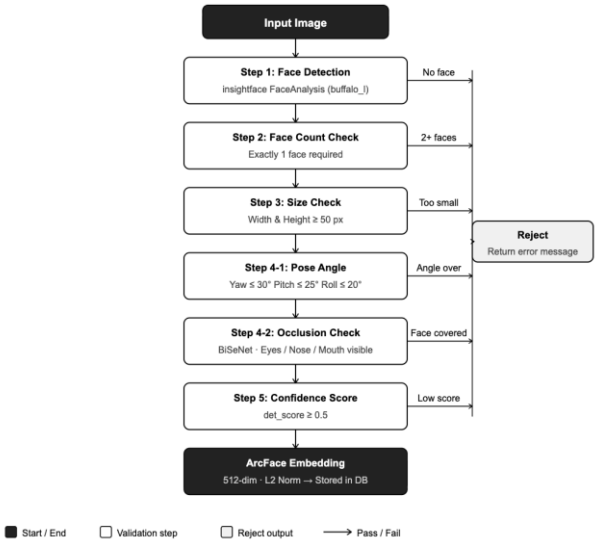


Fig. 3 Face Registration Validation Pipeline

TABLE 2
SIX-STAGE FACE VALIDATION PROCEDURE

Stage	Check	Verified Property / Method	Rejection Condition
1	Face detection	At least one face present (InsightFace buffalo_l)	No face detected
2	Face count	Exactly one face in the image	More than one face detected
3	Face size	Bounding-box width and height ≥ minimum size	Face smaller than minimum size
4	Head pose	Frontal orientation (|yaw|, |pitch|, |roll|)	Any angle exceeds its limit
5	Occlusion	Eyes, nose, mouth visible (BiSeNet parsing)	Any key part below visibility threshold
6	Confidence	Detection score ≥ detection threshold	Detection score below threshold

1) Face Image Input and Preprocessing

The user uploads an image of their own face to the system. Face detection on the uploaded image is performed using the face analysis module of InsightFace [9]. The system uses the InsightFace buffalo_l model, which includes a face detection model and an ArcFace-based embedding extraction model [10]. Through the detection process, the input image is converted into a set of detected faces, where each face provides attributes such as the location of the face region (bounding box), the detection confidence, the head-pose information (yaw, pitch, roll), and the embedding vector. The subsequent validation stages then examine the number of detected faces and the attributes of each face in turn.

2) Face Detection and Single-Face Validation

In the first validation stage, the system checks whether a face is properly detected in the input image. An image in which no face is detected cannot produce the embedding vector required for user identification and is therefore treated as a registration failure. In the second stage, the system checks the number of faces in the image. Because the registration image must be based solely on the facial features of the user, an image

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

containing two or more faces does not allow the reference face to be unambiguously determined and is excluded from registration.

3) *Face Size Validation*

In the third stage, the system checks the size of the detected face region. If the face region is too small, key facial features such as the eyes, nose, and mouth may not be sufficiently captured, which can lower the quality of the embedding. The system therefore sets a minimum face-size criterion and does not register face images smaller than this threshold.

4) *Face Orientation Validation*

In the fourth stage, the system verifies the face orientation based on the yaw, pitch, and roll values of the face. When a face is strongly rotated or captured in profile, the accuracy of feature extraction may degrade compared with a frontal face. Accordingly, registration is rejected when the horizontal rotation, vertical tilt, or in-plane rotation of the face each exceeds its allowable range.

5) *Facial Occlusion Validation*

In the fifth stage, the system checks whether the key facial parts are occluded. For this purpose, a BiSeNet-based face parsing model [11], [12] trained on the CelebAMask-HQ dataset [13] is used to segment the face region into 19 parts, and the visibility of key parts such as the eyes, nose, and mouth is examined. When these key parts are not sufficiently visible due to a mask, a hand, hair, or shadow, registration is restricted.

6) *Detection Confidence Check and Embedding Storage*

In the final stage, the system checks the confidence of the face detection result. If the detection confidence is lower than the configured threshold, an incorrect face region may have been detected, and the registration is therefore treated as a failure. An image that passes all validation stages is converted into a 512-dimensional facial embedding vector via ArcFace, which is then L2-normalized and stored as a unit vector. The normalized embedding vector is subsequently used as the reference for computing the cosine similarity with face embeddings extracted from collected images, serving as the reference data for the identity-matching process in LeakCare's similarity-based comparison.

C. AI Analysis Stage (Similarity and Deepfake Detection)

The AI analysis stage performs face similarity analysis and deepfake detection on images collected through web crawling, and constitutes the core inference stage of the system. This stage compares the face embedding registered by the user with face embeddings extracted from collected images to determine whether they belong to the same person, and additionally analyzes the possibility of manipulation through a deepfake detection model.

1) *Face Embedding-based Similarity Analysis*

After detecting a face in an image collected through web crawling, the system extracts a face embedding vector using the ArcFace model of InsightFace [9], [10]. ArcFace is a face recognition model trained so that faces of the same person are mapped close together in a high-dimensional embedding space, while faces of different people are mapped far apart. The system computes the cosine similarity between the user face embedding stored during registration and the face embedding extracted from the collected image. Both embedding vectors are L2-normalized before comparison, and the pair is judged to be the same person when the cosine similarity is at or above the configured similarity threshold.

2) *Face Similarity Threshold Setting*

The similarity threshold for the same-person decision directly affects the false acceptance and false rejection of the system. If it is too low, the likelihood of judging different people as the same person increases; if it is too high, the likelihood of failing to detect an actual same-person match increases. Therefore, this study established the face-matching criterion through a dedicated threshold experiment.

The experiment used 200 curated facial images, consisting of 10 images each for 20 public figures collected from publicly available web sources. From these 200 images, all possible pairs were constructed, yielding

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

19,900 pairs in total, of which 900 were same-person pairs and 19,000 were different-person pairs. For each pair, the cosine similarity between ArcFace embeddings was computed to compare the two distributions.

The same-person pairs showed a mean cosine similarity of 0.6038, while the different-person pairs showed a mean of 0.0941. Moreover, the minimum similarity among same-person pairs was 0.4517 and the maximum among different-person pairs was 0.4259, confirming that the two distributions are relatively clearly separated. Based on this separation, the system set the similarity threshold to 0.44, considering the boundary region between the two distributions: when the cosine similarity between a registered face and a collected face is 0.44 or higher, the pair is judged to be the same person, and otherwise a different person.

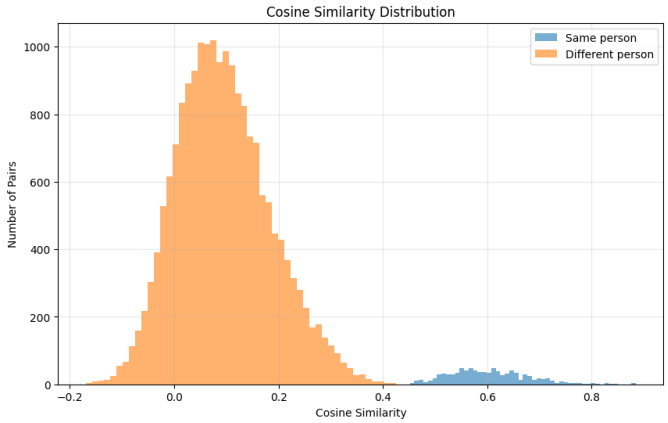


Fig. 4 Cosine similarity distribution of same-person and different-person pairs

3) Deepfake Detection Module

Model and Dataset Configuration

Face similarity analysis determines whether a collected image contains the user's face, but whether the image is an actually captured face or one generated or manipulated by deepfake techniques requires separate analysis. The system therefore combines a deepfake detection module with the similarity analysis to additionally judge the manipulation possibility of collected images. For this purpose, a ConvNeXt-Tiny-based binary classification model [14] is used. ConvNeXt-Tiny is a lightweight vision backbone that extracts visual features from an image; its final classification layer was modified for real/fake binary classification and the model was fine-tuned on the DF40 dataset [15].

The DF40 dataset contains real facial images and manipulated facial images generated using various deepfake manipulation techniques. In this study, nine manipulation techniques were used for training: e4s, uniface, facedancer, facevid2vid, tpsm, pirender, StyleGAN2, ddim, and VQGAN. These techniques cover multiple manipulation categories, including face swapping, face reenactment, and entire-face synthesis. During testing, four additional unseen techniques that were not included in the training set, namely e4e, stargan, starganv2, and styleclip, were added to evaluate the model's generalization performance on new manipulation methods.

The DF40 dataset has a structure in which each video or ID folder contains multiple frame images. Since frames extracted from the same video are likely to share similar visual characteristics, using all frames without filtering may cause the model to overfit to specific videos or identities. To mitigate this issue, folder-level sampling was applied. Specifically, at most 10 images were selected from each subfolder, reducing data redundancy and preventing overfitting to particular videos or manipulation methods.

In addition, to prevent the model from being biased toward specific manipulation methods due to method-level data imbalance, the fake images were reconstructed to have an approximately equal number of samples for each

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

manipulation method. Specifically, about 1,675±1 images per method were used for the training set, about 208±1 images per method for the validation set, and about 147±1 images per method for the test set. This method-level balancing was intended to reduce the dominance of any particular manipulation method during training and to encourage the model to learn common forgery-related features across different types of deepfake generation.

The real images were collected from FaceForensics++ (FF++) [16] and Celeb-DF (CDF) [17], and the real class was sampled to match the number of fake images in each split. The reconstructed dataset was divided into training, validation, and test sets. In this experiment, the training set consisted of 15,080 fake images and 15,080 real images, the validation set consisted of 1,870 fake images and 1,870 real images, and the test set consisted of 1,910 fake images and 1,910 real images. The real-to-fake ratio was maintained at 1:1 in each split to prevent the model from being biased toward a particular class.

TABLE 3
DATASET DISTRIBUTION BY SPLIT (REAL / FAKE BALANCED 1:1)

Split	Real	Fake	Total
Train	15,080	15,080	30,160
Validation	1,870	1,870	3,740
Test	1,910	1,910	3,820

Training and Classification Threshold

Input images were resized to the model input size and normalized before being fed to ConvNeXt-Tiny, and the model was trained to predict whether an input face is real or manipulated. The validation set was used to monitor performance during training, while the test set was held out for evaluation. The deepfake classification threshold is applied to the fake-class probability produced by the model. Several candidate thresholds were compared using the validation set, and the threshold that achieved the highest F1-score, 0.65, was selected. Accordingly, a face is judged to be a deepfake when its fake probability is 0.65 or higher. This deepfake classification threshold is conceptually distinct from the face similarity threshold (0.44) described in Section III-C-2: the former operates on the manipulation probability of a single face, whereas the latter operates on the identity similarity between two faces.

Deepfake Detection Results

On the balanced test set of 3,820 images, the fine-tuned ConvNeXt-Tiny model achieved an overall accuracy of 0.936. Table 4 reports the per-class precision, recall, and F1-score. Of 1,910 real images, 1,867 were correctly classified, while 43 were misclassified as fake. For the fake class, 1,708 out of 1,910 images were correctly classified, while 202 were misclassified as real.

TABLE 4
CLASSIFICATION PERFORMANCE ON THE TEST SET (ACCURACY = 0.936)

Class	Precision	Recall	F1-score	Support
Real	0.902	0.978	0.938	1,910
Fake	0.975	0.894	0.933	1,910
Macro avg	0.939	0.936	0.936	3,820

Since the test set includes both seen and unseen manipulation techniques, the method-wise fake detection rate provides additional insight beyond overall accuracy. The detector performs almost perfectly on ddim (1.00), e4s (0.99), starganv2 (0.99), and facevid2vid (0.99), and remains strong on stargan (0.96), e4e (0.97), VQGAN (0.91), and tpsm (0.88). The lowest rates occur on uniface (0.58) and StyleGAN2 (0.76). Notably, the weakness is not aligned with the seen/unseen split.

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

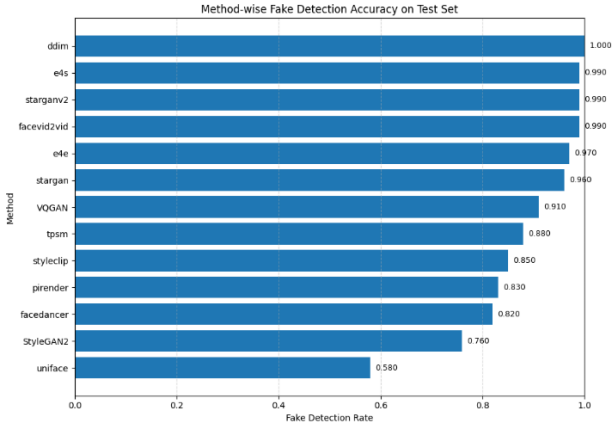


Fig. 5 Method-wise fake detection rate on the test set (13 techniques)

The trained model is saved as deepfake_detector_v12.pth and integrated into the LeakCare AI analysis pipeline. When a face similar to the user's registered face is detected in a collected image, that image is passed to the deepfake detection module, which predicts whether it is real or fake and returns a manipulation-possibility score.

4) Final Decision Procedure

The AI analysis stage uses the face similarity result and the deepfake detection result together. After detecting a face in a collected image, it computes the cosine similarity between the registered and collected face embeddings to determine identity. When the similarity is at or above the threshold (0.44), the image is classified as leak-suspected content related to the user. The deepfake detection model is then applied only to the images judged to be the same person, to analyze whether each is real or fake. Restricting deepfake detection to images that have already passed the similarity check reflects the intended pipeline—manipulation analysis is meaningful only for content that actually contains the user—and avoids running the detector on unrelated images. Finally, the system stores the face similarity score, the same-person decision, and the deepfake detection result; this information is subsequently used in the automatic evidence-report generation and takedown-request drafting stages.

D. Web Evidence Collection Engine

The SYS module automatically collects evidence from target URLs using a Playwright-based headless browser.

Bot Detection Evasion: To bypass automated browser detection, the system removes the webdriver property, configures Korean locale and Seoul timezone settings, and applies a User-Agent string resembling a real user. Requests to tracking domains such as Google Analytics and DoubleClick are blocked, and font and media file loading is also blocked to improve processing speed.

Analysis Modes: The system supports two modes: single-page analysis (single) and full board traversal (board). In board mode, the system automatically detects pagination by analyzing URL query parameters (page= or po= format) and traverses multiple pages.

Image Collection and Evidence Capture: After accessing a page, the system performs scrolling to force lazy-loaded images to render, captures a full-page screenshot, and collects image URLs. These are then downloaded asynchronously via httpx. Server IP address, country, and city information are retrieved via ip-api.com and included in the evidence data.

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

Parallel Processing: Collected images are analyzed in parallel using asyncio.Semaphore(5), processing up to five images simultaneously to maximize throughput.

E. Automated Removal Request Generation

In this study, a Claude API-based multilingual automated removal request system was implemented to support victims in taking initial action against illegal content distributed on overseas servers. While the urgency of responding to overseas platforms is high given that 70.4% of illegal sites are located in the United States and 95.4% operate on overseas servers overall, it is practically very difficult for victims to personally draft removal requests in local languages due to language barriers and a lack of legal knowledge.

The data flow of this system is as follows. First, the AI analysis module collects the country and city information of the server hosting the detected image by retrieving the server IP and integrating with ip-api.com. The SYS engine then constructs a prompt containing the detection results, server country, URL, and other relevant information, which is transmitted to Claude API (Haiku 4.5). Claude API automatically generates a draft removal request email in accordance with the language and legal framework of the country where the target server is located, including DMCA (Digital Millennium Copyright Act) guidelines. The generated text is returned to the frontend via the server and made immediately available to the user through a clipboard copy feature.

In addition, a PDF evidence report is automatically generated using the ReportLab library, incorporating detection timestamps, server IP and country information, on-site capture screenshots, similarity scores, and deepfake classification results. SHA-256 hashing and KST precision timestamps are applied to ensure the report can serve as legally valid digital evidence [2]. In this way, LeakCare goes beyond a simple detection tool to implement an integrated automated system that enables victims to independently carry out the entire initial response process without specialized legal or technical knowledge.

IV. RESULTS

A. Platform Interface

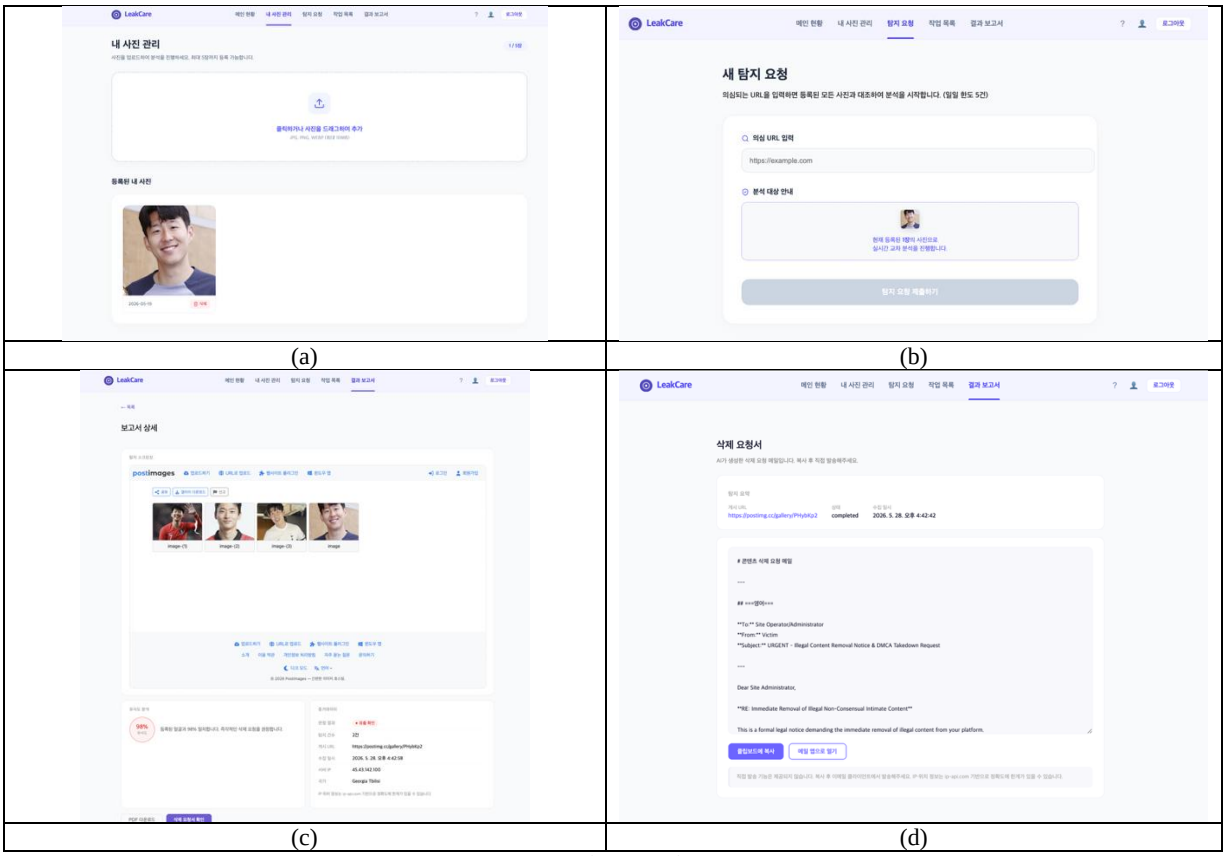
The LeakCare platform provides an intuitive web-based interface designed to guide victims through the entire detection and response workflow without requiring specialized technical expertise. The frontend was developed using React 19 and Vite, ensuring a responsive and efficient user experience. Fig. 6 illustrates the key screens of the implemented frontend.

As shown in Fig. 6(a), the main dashboard presents a summary of the user's detection activity, including the total number of detections, pending removal requests, and ongoing tasks. Detection results are categorized by risk level, and recent detection logs are displayed with their corresponding risk status, allowing users to monitor the current state of all submitted tasks at a glance.

Fig. 6(b) shows the report detail page, which displays the forensic evidence screenshot captured from the target URL alongside the AI analysis results. The page presents the facial similarity score, detection count, target URL, collection timestamp, server IP address, and geographic location of the hosting server. Users can download the PDF evidence report or proceed to the removal request generation directly from this page.

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056



(a) (b)

(c) (d)

Fig. 6 LeakCare Platform Interface Screenshots

B. Removal Request Letter Generation

Removal request letters generated via the Claude API within the SYS engine are dynamically produced based on the detected server's country. For example, when the server is located in the United States, English and Korean versions are generated. When the server is located in Japan, English and Japanese versions are generated. The generated emails include the detected URL, server IP address, collection timestamp, number of detections, and similarity score. Table 5 summarizes a sample output.

TABLE 5
SAMPLE REMOVAL REQUEST LETTER GENERATION OUTPUT

Item	**Value**
Target URL	https://postimg.cc/gallery/v5T718Z
Server Country	Georgia Tbilisi
Detection Count	3
Top Similarity Score	84%
Content Type	Deepfake
Languages Generated	English + Georgian language

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

(a) (b)

Fig. 7 Sample PDF Evidence Report

V. CONCLUSION

This paper proposed and implemented LeakCare, an AI-based detection and automated response platform for victims of deepfake and illegally filmed content. The key contributions of this system are as follows.

First, high-quality face embedding registration is ensured through a six-stage quality validation pipeline incorporating BiSeNet-based face parsing, enabling precise and reliable victim identity representation.

Second, efficient image collection and analysis across large-scale web pages is achieved through a Playwright-based automatic evidence collection engine combined with asyncio-based asynchronous parallel processing, significantly reducing the time required for forensic evidence gathering.

Third, two types of harmful content are detected in an integrated manner by combining ArcFace-based facial similarity comparison with ConvNeXt-based deepfake detection, with an adaptive dual-threshold strategy applied to maximize recall for deepfake-synthesized material.

Fourth, by leveraging the Claude API within the SYS engine to automatically generate legally-oriented removal request letters tailored to the server's host country, LeakCare enables victims without legal expertise to initiate formal removal procedures immediately and without the trauma of directly engaging with leaked content.

Future work includes adding real-time monitoring capabilities, improving deepfake detection accuracy through additional model training, and integrating platform-specific automated reporting APIs to further reduce the burden on victims.

ACKNOWLEDGEMENT

This research was supported by the MSIT(Ministry of Science ICT), Korea, under the National Program for Excellence in SW, supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation) in 2026 (No. 2024-0-00023).

International Journal of Computer Science and Mobile Computing
A Monthly Journal of Computer Science and Information Technology

Seohyun Lee *et al*, Volume 15, Issue 7, July 2026, pg. 33-45
(An Open Accessible, ISI Indexed, Fully Refereed and Peer Reviewed Journal)
ISSN: 2320-088X
Impact Factor: 7.056

REFERENCES

[1]. Kim, J., An, J., Yang, B., Jung, J., & Woo, S. S. (2020). 데이터 기반 딥페이크 탐지기법에 관한 최신 기술 동향 조사. REVIEW OF KIISC, 30(5), 79-92.

[2]. Kim, D., & Lee, S. (2022). Detection of Copyright Infringement in Deepfake Video using Face Detector. Journal of Digital Forensics , 16(1), 1-12.

[3]. Sin, S. (2023). A study on the digital sex crimes using deepfake technology. 한국치안행정논집, 20(4), 147-168.

[4]. Song, H. J., & Yoon, C. H. (2023-11-23). Study on Detection and Punishment Strategies for Pornography Produced by AI. Proceedings of KIIT Conference, Jeju.

[5]. Jung, H., Cho, H., Kim, W., & Lee, K. (2024). Vision Transformer-based Deepfake Detection Using Multiscale Features. Journal of Intelligence and Information Systems, 30(2), 275-285. 10.13088/jiis.2024.30.2.275

[6]. Kim, H., Ahn, H. E., Park, L. H., & Kwon, T. (2024). On the Performance Biases Arising from Inconsistencies in Evaluation Methodologies of Deepfake Detection Models. Journal of the Korea Institute of Information Security & Cryptology, 34(5), 885-893.

[7]. Hyemin, J. (2025). The Reality of 'Deepfake Sexual Crimes' based on Court Decision Analysis. The Women's Studies, 124(1), 5-28. 10.33949/tws.2025.124.1.001

[8]. Ko, M., & Kim, J. (2025). A Study on the Risk and Countermeasures of Deepfake Sexual Crimes. Legal Theory & Practice Review, 13(2), 201-219. 10.30833/LTPR.2025.05.13.2.201

[9]. InsightFace, "*InsightFace: 2D and 3D Face Analysis Project*," GitHub repository. [Online]. Available: https://github.com/deepinsight/insightface

[10]. J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "*ArcFace: Additive Angular Margin Loss for Deep Face Recognition*," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4690–4699.

[11]. C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "*BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation*," in Proc. European Conf. Computer Vision (ECCV), 2018, pp. 325–341.

[12]. zllrunning, "*face-parsing.PyTorch*," GitHub repository, 2019. [Online]. Available: https://github.com/zllrunning/face-parsing.PyTorch

[13]. C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "*MaskGAN: Towards Diverse and Interactive Facial Image Manipulation*," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5549–5558.

[14]. Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2022, pp. 11976–11986.

[15]. Z. Yan, T. Yao, S. Chen, Y. Zhao, X. Fu, J. Zhu, D. Luo, C. Wang, S. Ding, Y. Wu, and L. Yuan, "DF40: Toward Next-Generation Deepfake Detection," in Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track, 2024.

[16]. A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), 2019, pp. 1-11.

[17]. Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2020.

