CLIO: Climate-aware and Legacy-based Intelligent Outfit Recommendation System

Rio Kawasaki and Keiichi Takahashi *

Department of Information and Computer Science, Faculty of Humanity-Oriented Science and Engineering, Kindai University, Iizuka, Japan
Email: 2211140043v@ed.fuk.kindai.ac.jp (R.K.); ktakahas@fuk.kindai.ac.jp (K.T.)
* Corresponding author

Manuscript received July 5, 2026; revised August 12, 2026; accepted September 3, 2026; published September 22, 2026

***Abstract*—Choosing an appropriate outfit for the day is a common daily challenge. While many fashion applications exist, they typically recommend items from external catalogs without considering the user's own wardrobe or the current weather conditions. Furthermore, outfit logging applications record past outfits but offer no recommendations based on that history. This paper presents CLIO, a Climate-aware and Legacy-based Intelligent Outfit Recommendation System that addresses these gaps. CLIO recommends outfits exclusively from the user's personal wardrobe by scoring each registered outfit according to the similarity between the current weather and the weather on the day the outfit was worn. CLIO also prevents unintentional repetitive wearing by excluding recently worn or visually similar outfits from recommendations for a fixed period. Clothing regions are extracted using an instance segmentation model trained on a custom-annotated clothing dataset, and outfit similarity is computed via cosine similarity of deep convolutional feature vectors. A five-day deployment with a single user confirmed that both algorithmic components run end-to-end under real daily-use conditions. This paper is a systems contribution: we report the design, implementation, and proof-of-concept deployment of the pipeline, and we neither claim nor evaluate the effectiveness of the recommendations it produces. We specify the controlled study that establishing effectiveness would require, together with the quantitative evaluation the perception components require and the directions in which the weather model and preference handling should be extended.**

***Keywords*—outfit recommendation, weather-aware recommendation, instance segmentation, personal wardrobe**

I. Introduction

Deciding what to wear each day involves balancing multiple factors. A person must consider the weather forecast, ensure that recently worn items are not repeated too soon, and select from the clothes they actually own. Despite the widespread availability of fashion-oriented applications, no single system addresses all three concerns in an integrated manner.

A. Motivation

Two distinct sub-problems make daily outfit selection difficult.

One challenge is selecting clothing that is appropriate for the weather. A daily temperature forecast alone does not immediately translate into a specific outfit choice. Mobile applications and social media platforms offer fashion inspiration, but the outfits they display are drawn from external photo collections rather than the user's own wardrobe [1]. The presented images therefore serve only as abstract reference points; they cannot be directly adopted. Location-specific weather data is rarely integrated with the outfit selection process in a personal, wardrobe-bound context.

Another challenge is avoiding repetitive outfit selection. Most people own far more clothes than they regularly wear, yet without a deliberate tracking mechanism they tend to repeat a small subset of outfits. Outfit logging applications such as digital wardrobe managers let users photograph and catalog their garments, but they do not use the recorded history to suggest what to wear next.

B. Research Gap

Prior work on outfit recommendation has concentrated on learning compatibility among clothing items drawn from large public datasets such as DeepFashion [1] or Polyvore [2]. These systems excel at answering "which top matches these trousers?" but they treat the item pool as an open catalog rather than the user's personal, finite wardrobe. A smaller body of work considers contextual signals such as occasion or season [3], yet weather-level granularity, namely matching today's temperature range to the temperature range experienced when a past outfit was worn, remains largely unexplored. Similarly, while visual similarity measures have been applied to fashion retrieval [4], their use as a wear-history filter to prevent outfit repetition in a personal wardrobe setting has not been systematically studied. To address this gap, CLIO unifies personal wardrobe management, weather-score matching at the granularity of the daily temperature range, and similarity-based outfit exclusion into a single working system. The term legacy in the system's name refers to the user's own accumulated record of past outfits and the weather under which they were worn: CLIO draws its recommendations from that record rather than from an external catalogue or from prevailing fashion trends.

C. Contributions

The main contributions of this work are:

(1) A weather score metric that quantifies the suitability of each registered outfit for the current day by comparing the temperature range recorded on the day the outfit was worn with today's forecast, enabling ranking rather than binary filtering.

(2) A consecutive-wear prevention mechanism that uses an instance segmentation model trained on a custom clothing dataset and ResNet50-based feature extraction to detect visually similar garments and exclude the corresponding outfits from recommendations for a configurable exclusion window.

(3) A complete end-to-end prototype implemented on commodity hardware (Raspberry Pi4 and Camera

Module 3) combined with a Flask web application, showing that the pipeline is implementable and operable on hardware suited to everyday personal use.

(4) A proof-of-concept deployment with a single user, confirming that both algorithmic components run end-to-end under real daily-use conditions. This deployment is a functional check of the implementation; it is not an evaluation of recommendation effectiveness, and we make no such claim.

Scope and claims. This paper is a systems contribution. We report the design, implementation, and proof-of-concept deployment of an end-to-end pipeline that runs on commodity hardware. We deliberately do not claim, and do not evaluate, the effectiveness of the recommendations that CLIO produces. Establishing effectiveness would require a controlled study with independent participants and appropriate baselines; we specify such a study as a protocol in Section V rather than attempt it here. The deployment reported in Section IV is a functional check that the implemented components operate under real conditions, and the observations recorded during it form a narrative usage log, not a measurement of recommendation quality. We report the work at this stage because the design is what we wish to place on record, and the protocol in Section V is set out in enough detail that the evaluation it describes can be carried out either by us or by other groups.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the system architecture and details the two core algorithmic methods. Section IV reports the proof-of-concept deployment. Section V states the limitations of the present work and specifies the evaluation protocol that future work should follow. Section VI concludes the paper.

II. Literature Review

A. *Outfit Recommendation Systems*

Outfit recommendation has attracted considerable research attention over the past decade, driven by the availability of large-scale annotated fashion datasets and advances in deep learning. A central theme in this field is outfit compatibility—predicting whether a set of clothing items "go well together"—which was formalized as a learning problem by Han *et al.* [2]. Their bidirectional Long Short-Term Memory (LSTM) model encodes a sequence of garments and learns to score outfit compatibility from the Polyvore dataset, which contains over 400,000 community-curated outfits. Subsequent work has modeled outfit compatibility using graph neural networks [5], multi-task networks that perform both compatibility prediction and missing item retrieval [6], and contrastive learning frameworks [7].

A key limitation shared by these approaches is that they operate over a fixed, large item catalog. The user is implicitly assumed to have access to any item in the catalog, and the recommendation is "this top would complement those trousers," not "of the clothes you own, wear this outfit today." Personalization in this line of work typically means adapting to aggregated purchase or click history, not to an individual's physical wardrobe. This distinction is important in practice because a user who does not own the recommended item cannot act on the suggestion.

Several systems address personalization more directly. Kang *et al.* [8] demonstrated that incorporating user interaction history into visual embedding spaces markedly improves perceived relevance. Fashion recommendation from personal social media posts—where the user's own worn outfits appear as training signals—was explored by Zheng *et al.* [9], but the approach requires a substantial volume of labeled social images per user. In contrast, CLIO records outfits at the point of wearing through a dedicated hardware capture device, making the system usable from the first day.

B. *Wardrobe Management and Outfit Logging*

A separate category of applications focuses on recording what users wear rather than on recommendation. Digital wardrobe managers such as Stylebook and Smart Closet allow users to photograph and catalog their garments and create outfit combinations manually. These systems generate an outfit history as a side effect of daily use, but they do not analyze that history to drive future suggestions. Kosugi [10] proposed a "clothing playlist" system in which a Raspberry Pi camera and a passive infrared sensor automatically photograph the user's outfit each morning; the images are cataloged together with temperature and humidity readings from an environmental sensor. Although this system captures weather data alongside outfit images, it does not use that data to score or rank outfits for recommendation. CLIO extends this concept by introducing a weather-score metric that uses the temperature records to produce a ranked recommendation list.

Everyday accounts of outfit planning point to recurring difficulties: uncertainty about whether a garment is appropriate for the day's weather, difficulty recalling which items have been worn recently, and the cognitive load of combining garments into a coherent look. These findings directly motivate the two algorithmic features of CLIO: weather-score ranking addresses weather uncertainty, while consecutive-wear prevention addresses recent-use memory.

C. *Context-Aware and Weather-Aware Recommendation*

Context-aware recommender systems incorporate situational factors such as time, location, occasion, or weather into the recommendation process [11]. In the fashion domain, Chen *et al.* [3] proposed a model that conditions outfit generation on an occasion label (e.g., "formal," "casual," "sport"), demonstrating that occasion context substantially improves recommendation quality. Season has been used as a coarse weather proxy in several systems: items are tagged with a season attribute and filtered accordingly [12, 13].

Finer-grained weather signals have received less attention. Reviews of apparel recommendation systems show that weather-based fashion recommendation has typically operated at coarse category levels, mapping season or weather conditions (e.g., rainy or sunny) to broad garment types rather than ranking specific outfits from a personal wardrobe [14]. Generative models for full-body outfit synthesis and virtual try-on have also been explored [15]; however, such approaches produce new outfit visualizations rather than selecting from garments the user already owns.

CLIO takes a retrieval-based approach that is simpler and better suited to everyday use: the weather score is computed as the sum of absolute differences between today's temperature range and the temperature range recorded for each past outfit, and outfits are sorted by this score in ascending order. The proposed method does not require a pre-trained weather-garment mapping model; it instead uses the user's own record of past wearing days, accumulated across seasons, as the basis for weather-based retrieval.

D. *Clothing Detection and Visual Similarity*

Accurate identification of individual garments within a full-body outfit image is a prerequisite for computing meaningful outfit similarity. Liu *et al.* [1] released the DeepFashion benchmark, which contains 289,222 annotated images spanning 50 clothing categories and 1,000 attributes, enabling the development of robust clothing recognition models. Subsequent datasets, including DeepFashion-MultiModal [16], added pixel-level human parsing labels and rich textual descriptions, making them suitable for training segmentation-based detectors.

Instance segmentation—which assigns a per-pixel mask to each detected object—is particularly well suited to clothing detection because it can separate overlapping garments and preserve shape information useful for subsequent analysis. He *et al.* [17] introduced Mask R-CNN, which extends Faster R-CNN [18] with a parallel mask prediction branch, achieving state-of-the-art instance segmentation accuracy on the Common Objects in Context (COCO) benchmark. Detectron2 [19], developed by Facebook AI Research, provides a production-quality implementation of Mask R-CNN along with flexible configuration for fine-tuning on custom datasets. In CLIO, Detectron2 is used to train a clothing detector on a custom COCO-format dataset derived from DeepFashion-MultiModal, enabling extraction of five garment types from user-captured outfit photographs.

For computing visual similarity between extracted clothing regions, deep convolutional feature representations have consistently outperformed hand-crafted descriptors [20]. He *et al.* [21] showed that residual networks with skip connections (ResNets) achieve superior image classification accuracy and produce rich intermediate representations that transfer effectively to downstream tasks. ResNet50, with its 50-layer residual architecture pre-trained on ImageNet [22], has become a standard backbone for fashion retrieval tasks [2, 4]. In CLIO, the final pooling layer of ImageNet-pre-trained ResNet50 serves as a feature extractor for each cropped garment image, and cosine similarity between the resulting 2048-dimensional vectors is used to identify garments that are the same or visually similar.

The threshold choice for similarity-based retrieval is application-dependent. While tuning the prototype we observed that a threshold of 0.70 merged visually distinct garments, whereas 0.90 separated images of the same garment taken under different lighting, and we adopted 0.85 on the basis of those observations. These were qualitative observations on a small number of garment pairs, not a systematic threshold study; Section V specifies the sensitivity analysis that would be needed to establish an appropriate value. Similarity was computed using deep feature representations, following the general approach adopted in clothing retrieval studies [4].

E. *Summary and Research Gap*

Table 1 positions CLIO relative to the systems reviewed above across four key dimensions: recommendation from the personal wardrobe, fine-grained weather scoring—by which we mean matching against the daily temperature range rather than a coarse seasonal or occasion label—consecutive-wear prevention, and automated image capture. To the best of our knowledge, none of the systems reviewed above satisfies all four criteria. Outfit recommendation systems operate over external catalogs; wardrobe managers record history but do not recommend; weather-aware systems work at category or season granularity rather than with personal outfits; and visual-similarity systems focus on retrieval from large databases, not on personal repetition control. CLIO addresses all four dimensions in a unified, hardware-grounded prototype designed for everyday use.

Table 1. Comparison of Related Systems (✓ = yes, ✗ = no)

System	Personal wardrobe	Fine-grained weather	Repeat-wear prevention	Auto capture
Han *et al.* [2]	✗	✗	✗	✗
Cui *et al.* [5]	✗	✗	✗	✗
Chen *et al.* [3]	✗	partial	✗	✗
Guan *et al.* [14]	✗	partial	✗	✗
Kosugi [10]	✓	records	✗	✓
Digital wardrobe apps	√	✗	✗	✗
CLIO (proposed)	✓	✓	✓	✓

III. Materials and Methods

A. *Overview*

CLIO is designed around two principal use cases: registering a new outfit by taking a photograph at the time of wearing, and selecting and adopting a previously registered outfit from the recommendation list. In both cases, the system records the outfit together with the weather conditions of the day and updates the user's wearing history. When the user requests a recommendation, CLIO scores all registered outfits with the weather score, applies the consecutive-wear filter to remove recently or similarly worn outfits, and presents a ranked list through a web interface.

B. *Hardware*

The capture subsystem consists of a Raspberry Pi4 Model B, a Raspberry Pi Camera Module 3, and a tactile push button. The user positions themselves in front of the camera and presses the button to trigger a full-body outfit photograph. The captured image is immediately transmitted to the application server over the local network. The use of dedicated hardware avoids the need for the user to manually photograph themselves with a smartphone, reducing friction and encouraging consistent daily use.

C. *Software Architecture*

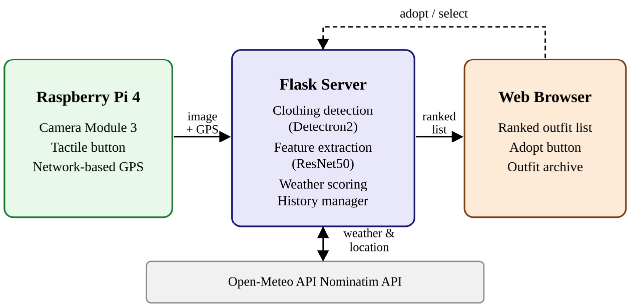

Fig. 1. Architecture of the CLIO system.

The server-side application is implemented as a Flask web application. Fig. 1 illustrates the overall architecture.

Upon receiving a new outfit image, the server performs the following steps automatically.

(1) **Clothing region extraction**. The outfit image is processed by the trained Detectron2 model, which produces per-garment segmentation masks and category labels. Individual garment crops are stored alongside the full outfit image.

(2) **Feature extraction**. For each garment crop, a 2048-dimensional feature vector is extracted using pre-trained ResNet50.

(3) **Weather data retrieval**. The Global Positioning System (GPS) coordinates transmitted by the Raspberry Pi are reverse-geocoded via the OpenStreetMap Nominatim Application Programming Interface (API) to obtain a place name, which is then used to retrieve the day's minimum and maximum temperature from the Open-Meteo free weather API.

(4) **Data persistence**. The outfit image, garment crops, feature vectors, adoption date, and weather record are saved in a JavaScript Object Notation (JSON) data file.

(5) **Consecutive-wear update**. Cosine similarities between the new outfit's garment features and all previously registered garments are computed. Outfits containing garments with similarity ≥ 0.85 are flagged for exclusion.

When the user opens the recommendation page, the server loads all registered outfits, removes those currently flagged for exclusion, computes the weather score for each remaining outfit using the current day's forecast, sorts the list by weather score in ascending order, and returns the ranked list to the web browser. Fig. 2 shows the resulting recommendation view.

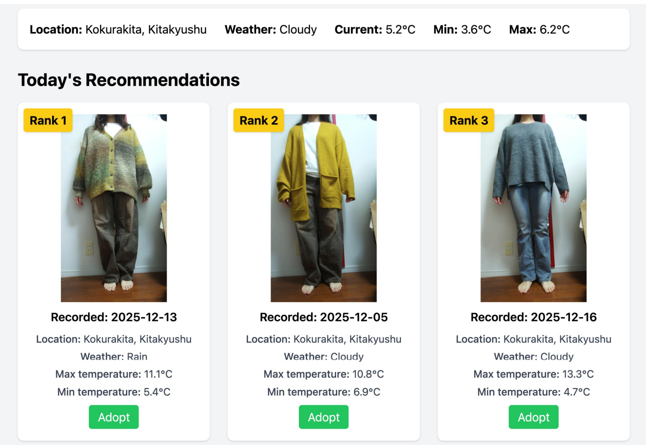

Fig. 2. Recommendation view of the CLIO web interface.

D. Outfit Adoption Workflow

If the user selects a previously registered outfit from the recommendation list and presses "Adopt," the server records today as the adoption date for that outfit and runs the consecutive-wear update procedure (step 5 above) against the adopted outfit's garment features. This ensures that the exclusion mechanism is applied whether the outfit was newly photographed or selected from the archive.

If the user decides to wear an outfit that is not in the archive, they photograph it via the Raspberry Pi interface (use-case 1), and the new outfit enters the archive immediately.

Weather Score

CLIO recommends outfits that were worn in weather conditions similar to today's, based on the intuition that if a garment combination was suitable on a day with a given temperature range, it is likely suitable on a day with a similar range. The suitability is quantified by the weather score:

$$S_{\text{weather}} = \left|T_{\text{today,min}} - T_{\text{past,min}}\right| + \left|T_{\text{today,max}} - T_{\text{past,max}}\right| \tag{1}$$

where $T_{\text{today,min}}$ and $T_{\text{today,max}}$ are today's minimum and maximum forecast temperatures (°C), and $T_{\text{past,min}}$ and $T_{\text{past,max}}$ are the corresponding values recorded on the day the outfit was worn (or registered). A lower score indicates a closer temperature match.

Outfits are sorted by S_{weather} in ascending order before being presented to the user. This simple non-parametric metric requires no training data beyond the user's own outfit records, makes no assumptions about garment type or material, and is directly interpretable. The metric uses both minimum and maximum temperature so that outfits recorded on days with a wide diurnal range are not over-matched to days with a narrow range at the same midpoint.

Weather data is fetched from the Open-Meteo API [23], which provides hourly and daily temperature forecasts at any geographic coordinate free of charge. Location is obtained from the coordinate reported by the Raspberry Pi or, as a fallback, from a pre-configured default location. Reverse geocoding is handled by the OpenStreetMap Nominatim API [24]. The current implementation operationalizes climate awareness through the daily temperature range alone. Precipitation, humidity, and wind speed are available from the same API but do not enter the weather score; Section V sets out how the score generalizes to include them.

E. Consecutive-Wear Prevention

1) Clothing region extraction

Identifying individual garments within an outfit photograph requires separating overlapping clothing items with pixel-level precision. We use instance segmentation rather than bounding-box detection because garments frequently occlude each other (e.g., a jacket partially covering a shirt), and bounding boxes from different items would substantially overlap. Mask R-CNN [17], implemented via Detectron2 [19], is fine-tuned on a custom dataset to detect five garment categories: outer layer, tops, trousers, skirts, and one-piece garments.

Dataset preparation. Training images are selected from DeepFashion-MultiModal [16], a large-scale dataset containing 44,096 high-resolution human images with 24-class human parsing annotations. We selected a subset of images in which the garments are clearly visible and representative of everyday wear. Because the existing parsing annotations in DeepFashion-MultiModal define fine-grained body part regions that do not correspond to our five garment-level categories, we created new instance-level annotations using COCO-Annotator [25]. Each garment region was manually outlined as a polygon mask and assigned one of the five category labels. The resulting annotations were exported as COCO-format JSON files for use with Detectron2.

Model training. The base model is a Mask R-CNN with a ResNet-50 + Feature Pyramid Network (FPN) backbone pre-trained on the COCO dataset, loaded from the Detectron2 model zoo. The custom clothing dataset is registered with Detectron2 and the model is fine-tuned by adjusting the number of output classes to five and training for a fixed number of iterations using stochastic gradient descent (SGD) with the default learning-rate schedule. After training, the model is used to create a predictor that is integrated into the CLIO server pipeline.

An example detection result is shown in Fig. 3, where the tops and trousers of a full-body outfit image are segmented as distinct instances.

(a) Input outfit image (b) Extracted garment regions (left: tops, right: pants)

Fig. 3. Clothing region extraction by Detectron2. (a) Input full-body outfit image. (b) Individual garment crops for ResNet50 feature extraction.

2) *Feature extraction and similarity computation*

Each extracted garment crop is resized and normalized, then passed through ImageNet-pre-trained ResNet50 [21] with the final fully connected layer removed. The output of the global average pooling layer is used as a 2048-dimensional feature vector that encodes color, texture, and shape information of the garment. Using an ImageNet-pre-trained network as a fixed feature extractor, rather than learning a representation from scratch, allows the system to operate from a small number of outfit photographs.

Cosine similarity between feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ is computed as:

$$\mathrm{sim}(\mathbf{f}_1, \mathbf{f}_2) = \frac{\mathbf{f}_1 \cdot \mathbf{f}_2}{\|\mathbf{f}_1\|\,\|\mathbf{f}_2\|} \tag{2}$$

Cosine similarity is chosen because it is insensitive to vector magnitude, which varies with garment size in the image, and because it captures directional agreement in the feature space, which correlates with perceptual visual similarity [20].

3) *Threshold selection and exclusion logic*

If the cosine similarity between a garment worn today and any garment in a registered outfit reaches or exceeds the threshold θ, the outfit is flagged for exclusion and removed from the recommendation list for the next Δt days. On the basis of qualitative observations made during prototype development, $\theta = 0.85$ was selected:

- At $\theta = 0.70$, visually distinct garments (different tops of a similar neutral color) were incorrectly matched, leading to over-exclusion of valid outfits.
- At $\theta = 0.90$, the same garment photographed under different lighting conditions or from a slightly different angle was sometimes classified as a different item, leading to missed exclusions.
- At $\theta = 0.85$, neither failure mode was observed in the small set of garment pairs examined during development.

The exclusion window is set to $\Delta t = 4$ days, reflecting a conventional guideline that wearing the same garment within four consecutive days is considered repetitive in typical everyday contexts. Both θ and Δt are configurable parameters in the system.

The three threshold values above were compared informally on a small number of garment pairs during development; they are qualitative observations rather than the result of a systematic sweep, and we report them as such. The value should be read as a provisional operating point for the prototype rather than as an empirically optimal threshold. The four-day exclusion window was likewise chosen by convention rather than by measurement. The window interacts with the size of the archive: because every excluded outfit is removed from the candidate list, a long window applied to a small archive can leave few or no candidates, whereas a short window permits the repetition that the mechanism exists to prevent. Section V specifies the sensitivity analysis needed to choose both parameters on an empirical basis.

Fig. 4 illustrates the complete consecutive-wear prevention pipeline from outfit capture to exclusion.

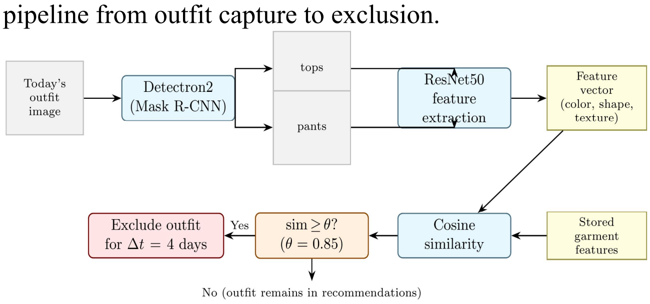

Fig. 4. Consecutive-wear prevention pipeline.

IV. Proof-of-Concept Deployment

This section reports a proof-of-concept deployment carried out to confirm that the two core features of CLIO—weather-score ranking and consecutive-wear prevention—run as intended in real daily use. It is a functional check of the implementation. It is not an evaluation of recommendation quality: the observations recorded below form a narrative usage log, and no quantity reported here should be read as a performance figure.

A. Setup

The deployment was conducted by the first author alone over fifteen consecutive days in December 2025. During the first ten days (December 3–12), the author photographed one outfit per day using the Raspberry Pi hardware, building an initial archive of ten registered outfits. During the subsequent five days (December 13–17), the author used CLIO each morning, received the top-ranked outfit suggestion, and recorded (i) whether the suggestion was adopted, (ii) the primary reason for adoption or rejection, and (iii) any alternative action taken.

The deployment period falls in winter in the Kyushu region of Japan. Weather conditions over the five trial days included one rainy day, three overcast days, and one clear day. Daily maximum temperatures ranged from 11.1 °C to 15.8 °C and daily minimum temperatures from 4.7 °C to 9.3 °C, a spread sufficient to produce meaningful variation in weather scores across the registered outfits.

B. Record of the Deployment

Table 2 summarizes the outcome of each day.

Table 2. Day-by-day record of the deployment (single user, five consecutive days)

Day	Date	Outcome	Primary Reason
1	Dec. 13	Adopted	Weather match; suited to preference
2	Dec. 14	Adopted	Weather match; suited to preference
3	Dec. 15	Not Adopted	Preference mismatch; used list as reference
4	Dec. 16	Not Adopted	Preference mismatch; new outfit registered
5	Dec. 17	Adopted	Weather match; appropriate for casual outing

On three of the five days the top-ranked suggestion was worn as proposed. This is a narrative record of five days of use by one person; it is not an adoption rate and should not be read as one. On the three adopted days, the author noted that the top-ranked outfit felt appropriate for the day's temperature, which is consistent with the weather score having surfaced outfits registered under similar conditions, although one user's impression cannot establish that it did so reliably. On the two non-adopted days, the stated reason was a subjective preference mismatch: the proposed outfit was judged weather-appropriate but did not match the author's mood or intended social context. On December 15, the author did not adopt any single outfit but used the ranked list as a visual reference while assembling a new combination, which was then registered in the archive.

Regarding consecutive-wear prevention, no garment that had been worn during the registration phase reappeared in the recommendation list during the verification phase, and the author did not observe any outfit being incorrectly excluded due to false-positive similarity matches. These observations suggest that the $\theta = 0.85$ threshold produced no visible failure in an archive of this size. With so few garments, and with no ground-truth labelling of which garments are in fact the same, this is an absence of observed errors rather than evidence of correctness.

C. Discussion

These observations carry no evidential weight beyond the fact that the system ran. Five days of use by a single evaluator who is also the system's developer cannot establish effectiveness or generalizability, and we make no such claim. What the deployment does establish is narrower and, for a proof of concept, sufficient: the pipeline executed end-to-end under real conditions, the weather score ordered the archive by temperature match as specified, and the consecutive-wear filter removed recently worn outfits without producing an exclusion that the user judged incorrect.

The preference mismatch observed on two days points to the most significant gap in the current system: the weather score captures temperature suitability but not personal taste, social context, or daily mood. Addressing this gap through preference modeling based on adoption history is the most direct path to improving recommendation quality in future work.

D. Threats to Validity

Several factors limit what can be inferred from the deployment, and we state them plainly. The evaluator was the system's developer, so confirmation bias cannot be excluded from the recorded judgments. The deployment covers five days, one person, a single winter month, and one geographic location, so nothing about other seasons, climates, or users follows from it. The candidate archive was small, which makes the ranking task easier than it would be for a full wardrobe. No baseline was run, so the record cannot be compared against random selection, most-recently-worn selection, or an existing wardrobe application. The judgments of adoption and rejection were self-reported and unblinded. These are not incidental weaknesses that a larger version of the same trial would remove; they are the reason the present work is reported as a proof of concept, and the reason Section V specifies a different study design.

V. Limitations and Future Evaluation

The present work includes no user evaluation. The deployment involved one user over five days and was carried out by the system's developer, which cannot support any inference about effectiveness or generalizability. The verification is qualitative in nature, and the reported outcomes reflect the judgment of a single evaluator who is also the system developer. Multi-user studies over longer periods are needed to assess generalizability.

The current study includes no baseline comparison. The verification does not compare CLIO against any baseline strategy, such as random outfit selection, most-recently-worn selection, or an existing wardrobe application. The day-by-day record is therefore descriptive rather than comparative, and no claim of relative improvement over any alternative approach can be made from it.

The current system does not include a preference model. CLIO ranks outfits purely by weather suitability. Aesthetic preferences, social context, and daily mood are not captured, which is consistent with the reasons recorded on the two days when the suggestion was not worn.

Segmentation accuracy has not been measured. The custom Detectron2 model was trained on a relatively small annotated subset of DeepFashion-MultiModal, and we report no detection or mask accuracy for it: no held-out test split was evaluated, so no average-precision figure is available. In cases where the outfit image is poorly lit, the garments partially overlap, or an unseen garment type is present, the detector may produce inaccurate segmentation masks. This can degrade the quality of extracted feature vectors and therefore the reliability of the similarity-based exclusion.

The threshold and exclusion window are fixed and were never swept. The cosine similarity threshold (0.85) and the exclusion window (4 days) are global constants, chosen from qualitative observation and from convention respectively rather than from a sensitivity analysis. Different garment types (e.g., solid-color T-shirts versus patterned jackets) and different users may require different threshold values for optimal performance.

The weather model considers only temperature. The weather score uses only the daily minimum and maximum temperature. Other factors relevant to outfit choice—precipitation, wind speed, and humidity—are retrieved from the API but are not used in the score. The score generalizes to a weighted sum in which the two temperature differences are joined by terms for precipitation, humidity, and wind, each carrying its own weight; the current implementation is the special case in which the two temperature weights are one and the remaining weights are zero. We did not activate the additional terms, because there is no basis on which to calibrate their weights: doing so requires labelled preference data that the present work does not have, and assigning weights arbitrarily would produce a ranking whose behaviour we could not account for.

Establishing effectiveness and generalizability requires a controlled study that the present work does not attempt, and we state its design here so that the claim boundary of this paper is explicit. Such a study should recruit participants independent of the development team—on the order of twenty, each using their own wardrobe—and should run for at least four weeks, so that the archive reaches a realistic size and the exclusion window is exercised repeatedly. Each participant should be exposed, within subjects, to CLIO's ranking and to three baselines: a random ordering of the archive, a most-recently-worn ordering, and unassisted selection, with an existing wardrobe-management application included where its interface permits a matched task. Appropriate outcome measures are the rate at which the top-ranked outfit is worn, the rate at which the worn outfit appears among the top three, the normalized discounted cumulative gain of the ranking against what the participant actually wore, the mean interval between repeat wearings of a garment, the proportion of the wardrobe worn at least once over the study period, and a standard usability instrument for the decision burden the system is intended to reduce. The analysis plan should be fixed before data collection, and outcomes should be recorded by someone other than the system's developers.

The perception components require measurement of their own, which the present work also does not provide. The clothing detector should be evaluated on a held-out split of the custom annotations, reporting mask average precision overall and for each garment category, with results broken out by the conditions under which the system is actually used: low light, garments that occlude one another, and patterned as opposed to plain fabrics. The similarity component should be evaluated separately from the detector, on a set of garment crops labelled as depicting the same garment or different garments, reporting the distribution of cosine similarities for the two classes and the precision and recall that follow as the threshold is swept from 0.60 to 0.95; this determines whether 0.85 is an appropriate operating point and whether a single global threshold is adequate across garment types. The exclusion window should be swept jointly, from one day to roughly two weeks, reporting both the repetition it prevents and the proportion of the archive it renders unavailable, since these move in opposite directions and the appropriate setting depends on wardrobe size.

Extending the model beyond temperature depends on that study rather than preceding it. Once adoption and rejection have been recorded across a population of users, the weights of the generalized weather score can be fitted rather than assumed, and the same feedback supports a preference model: adoption and rejection can be treated as implicit pairwise judgments and used to learn a re-ranking over features combining the weather score, the interval since a garment was last worn, garment category, and colour, falling back to the weather score alone while a new user's history remains short. Contextual variables can be introduced along the same path—occasion inferred from a calendar, expected time spent outdoors, and the social setting of the day—each straightforward to retrieve, but none of which can be weighted responsibly without the preference data such a study would produce.

VI. Conclusion

This paper presented CLIO, a Climate-aware and Legacy-based Intelligent Outfit Recommendation System that recommends outfits from the user's personal wardrobe by matching today's temperature range to the temperature range of past wearing days and by preventing repeated wearing of visually similar garments. The system integrates Raspberry Pi-based automatic outfit capture, Detectron2 instance segmentation for clothing extraction, and ResNet50-based cosine similarity for repetition detection. A five-day deployment with a single user confirmed that the pipeline runs end-to-end under real daily-use conditions. We report this as a proof of concept and make no claim about the effectiveness of the recommendations: establishing that would require a controlled study with independent participants and baselines, together with quantitative evaluation of the detection and similarity components, and Section V sets out the protocol for both. Preference modelling from adoption feedback, and the incorporation of precipitation, humidity, and wind into the score, would both require the data that such a study would produce.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

R. Kawasaki designed and implemented the system, prepared the dataset annotations, conducted the proof-of-concept deployment, and drafted the manuscript. K. Takahashi supervised the research, advised on the methodology and evaluation design, and revised the manuscript. Both authors have read and approved the final version of the manuscript.

Acknowledgment

The authors thank the members of the laboratory at Kindai University for their helpful discussions during the development of this work. During the preparation of this manuscript, the authors used a generative AI tool (Claude, Anthropic) to assist with the English-language translation and editing of the text. All research design, implementation, experiments, and results are the authors' own work; the authors reviewed and verified the final text and take full responsibility for its content.

References

[1] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1096–1104. doi: 10.1109/CVPR.2016.124

[2] X. Han, Z. Wu, Y.-G. Jiang, and L. S. Davis, "Learning fashion compatibility with bidirectional LSTMs," in *Proc. 25th ACM Int. Conf. Multimedia (ACM MM)*, Mountain View, CA, USA, Oct. 2017, pp. 1078–1086. doi: 10.1145/3123266.3123394

[3] X. Chen, Y. Chen, Z. Xu, Q. Lu, Y. Zhang, and K. Zheng, "POG: Personalized outfit generation for fashion recommendation at Alibaba iFashion," in *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Anchorage, AK, USA, Aug. 2019, pp. 2662–2670. doi: 10.1145/3292500.3330652

[4] D. Manandhar, M. Bastan, and K.-H. Yap, "Tiered deep similarity search for fashion," in *Proc. Eur. Conf. Computer Vision (ECCV) Workshops*, Munich, Germany, Sep. 2018, pp. 21–29. doi: 10.1007/978-3-030-11015-4_3

[5] Z. Cui, Z. Li, S. Wu, X.-W. Zhang, and L. Wang, "Dressing as a whole: Outfit compatibility learning based on node-wise graph neural networks," in *Proc. World Wide Web Conf. (WWW)*, San Francisco, CA, USA, May 2019, pp. 307–317. doi: 10.1145/3308558.3313444

[6] Y. Lin, P. Ren, Z. Chen, Z. Ren, J. Ma, and M. de Rijke, "Improving outfit recommendation with co-supervision of fashion generation," in *Proc. World Wide Web Conf. (WWW)*, San Francisco, CA, USA, May 2019, pp. 938–948. doi: 10.1145/3308558.3313614

[7] K. Sevegnani, A. Seshadri, T. Wang, A. Beniwal, J. McAuley, A. Lu, and G. Medioni, "Contrastive learning for interactive recommendation in fashion," in *Proc. ACM SIGIR Workshop on eCommerce (SIGIR eCom)*, Jul. 2022.

[8] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley, "Visually-aware fashion recommendation and design with generative image models," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, New Orleans, LA, USA, Nov. 2017, pp. 207–216. doi: 10.1109/ICDM.2017.30

[9] H. Zheng, K. Wu, J.-H. Park, W. Zhu, and J. Luo, "Personalized fashion recommendation from personal social media data: An item-to-set metric learning approach," in *Proc. IEEE Int. Conf. Big Data (BigData)*, Dec. 2021, pp. 5014–5023. doi: 10.1109/BigData52589.2021.9671563

[10] R. Kosugi, "Fuku no Playlist: A daily outfit recording system for remote workers," *Nikkei xTECH*, Jan. 14, 2022. [Online]. Available: https://project.nikkeibp.co.jp/pc/atcl/22/01/11/00035/011400008/

[11] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., Springer, Boston, MA, USA, 2011, pp. 217–253. doi: 10.1007/978-0-387-85820-3_7

[12] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan, "Hi, magic closet, tell me what to wear!" in *Proc. 20th ACM Int. Conf. Multimedia (ACM MM)*, Nara, Japan, Oct. 2012, pp. 619–628. doi: 10.1145/2393347.2393433

[13] X. Song, F. Feng, J. Liu, Z. Li, L. Nie, and J. Ma, "NeuroStylist: Neural compatibility modeling for clothing matching," in *Proc. 25th ACM Int. Conf. Multimedia (ACM MM)*, Mountain View, CA, USA, Oct. 2017, pp. 753–761. doi: 10.1145/3123266.3123314

[14] C. Guan, S. Qin, W. Ling, and G. Ding, "Apparel recommendation system evolution: An empirical review," *Int. J. Clothing Science and Technology*, vol. 28, no. 6, pp. 854–879, 2016. doi: 10.1108/IJCST-09-2015-0100

[15] H. Yang, R. Zhang, X. Guo, W. Liu, W. Zuo, and P. Luo, "Towards photo-realistic virtual try-on by adaptively generating–preserving image content," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 7850–7859. doi: 10.1109/CVPR42600.2020.00787

[16] Y. Jiang, S. Yang, H. Qiu, W. Wu, C. C. Loy, and Z. Liu, "Text2Human: Text-Driven Controllable Human Image Generation," *ACM Trans. Graph.*, vol. 41, no. 4, Art. 162, pp. 162:1–162:11, Jul. 2022. doi: 10.1145/3528223.3530104

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, Venice, Italy, Oct. 2017, pp. 2961–2969. doi: 10.1109/ICCV.2017.322

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 28, Montreal, Canada, Dec. 2015.

[19] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," Facebook AI Research, 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[20] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Trans. Graphics (SIGGRAPH)*, vol. 34, no. 4, pp. 98:1–98:10, Aug. 2015. doi: 10.1145/2766959

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848

[23] P. Zippenfenig, *Open-Meteo.com Weather API*, Zenodo, 2023. doi: 10.5281/zenodo.7970649

[24] OpenStreetMap Contributors. (2023). Nominatim. [Online]. Available: https://nominatim.org

[25] J. Brooks. (2019). COCO-Annotator. [Online]. Available: https://github.com/jsbroks/coco-annotator