SkySearch: Satellite Video Search at Scale

Minyoung Choe*
KAIST
Seoul, Republic of Korea
minyoung.choe@kaist.ac.kr

Geon Lee*
KAIST
Seoul, Republic of Korea
geonlee0325@kaist.ac.kr

Changhun Han*
Kookmin University
Seoul, Republic of Korea
codingnoye@kookmin.ac.kr

Suji Kim
Kookmin University
Seoul, Republic of Korea
suji2924@kookmin.ac.kr

Woong Hu
Satrec Initiative Co., Ltd.
Daejeon, Republic of Korea
woonghu@satreci.com

Hyebeen Hwang
Satrec Initiative Co., Ltd.
Daejeon, Republic of Korea
hbh@satreci.com

Geunseok Park
Satrec Initiative Co., Ltd.
Daejeon, Republic of Korea
gspark@satreci.com

Byeongyeon Kim
National Institute of Meteorological Sciences
Jeju, Republic of Korea
bykim1011@korea.kr

Hyesook Lee
National Institute of Meteorological Sciences
Jeju, Republic of Korea
hslee05@korea.kr

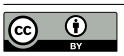
Ha-Myung Park†
Kookmin University
Seoul, Republic of Korea
hmpark@kookmin.ac.kr

Kijung Shin†
KAIST
Seoul, Republic of Korea
kijungs@kaist.ac.kr

Abstract

Satellite images play a crucial role in weather analytics, and recent advancements in satellite technology have significantly enhanced the accuracy and reliability of weather predictions. In this paper, we introduce SkySearch, a large-scale satellite video search system deployed at the Korean Meteorological Administration (KMA). SkySearch is designed to aid weather experts in making timely and accurate forecasts by rapidly and precisely searching for satellite videos in the database that resemble current weather conditions. SkySearch employs self-supervised learning to compress large volumes of high-resolution satellite videos into low-dimensional embeddings, addressing the absence of labeled satellite data. Within this latent space, relationships between videos are modeled as a graph, enabling efficient searches. Given a query video, SkySearch rapidly identifies a small subset of similar videos by traversing the graph. When the query video represents the current conditions, it can optionally be augmented with predicted future frames to search for videos that reflect both the current conditions and expected evolution. The identified videos are then ranked based on their similarity to the query video. Finally, the ranked list of videos is provided to weather forecasters through a user-friendly interface. We demonstrate the deployment and empirical effectiveness of SkySearch through both numerical and qualitative evaluations. In summary, SkySearch is: (a) Scalable: processes queries from a large-scale database of satellite images spanning over a decade and delivers results within seconds, (b) Accurate: returns numerically and qualitatively similar videos to the query video, and (c) Label-free: does not require labeled videos.

*Co-first authors.
†Co-corresponding authors.



KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737263

CCS Concepts

• **Information systems** → *Data mining*; **Retrieval models and ranking**; *Spatial-temporal systems*; • **Computing methodologies** → *Machine learning*; • **Applied computing** → **Earth and atmospheric sciences**.

Keywords

Satellite Video, Similarity Search, Video Retrieval, Video Prediction

ACM Reference Format:
Minyoung Choe, Geon Lee, Changhun Han, Suji Kim, Woong Hu, Hyebeen Hwang, Geunseok Park, Byeongyeon Kim, Hyesook Lee, Ha-Myung Park, and Kijung Shin. 2025. SkySearch: Satellite Video Search at Scale. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737263

1 Introduction

Satellite images provide invaluable information for monitoring and forecasting environmental and atmospheric changes, enhancing our understanding of weather patterns, climate shifts, and natural events. Especially, over recent decades, advancements in satellite technology have greatly improved the accuracy and reliability of weather forecasting.

Weather patterns often recur, making it essential to analyze similar past events. For example, at the Korea Meteorological Administration (KMA), identifying historical patterns provides weather forecasters with strong references for analyzing and forecasting current weather conditions. As a result, retrieving similar past satellite videos (i.e., sequences of consecutive satellite images), ideally in real-time, can greatly assist weather forecasters, leading to more accurate and timely weather predictions. Especially, this case-based reasoning complements Numerical Weather Prediction (NWP) models, which often fail to simulate realistic cloud dynamics.

However, satellite video retrieval presents unique practical challenges. First, there are no predefined labels that indicate which past satellite videos are similar. Labeling them requires extensive

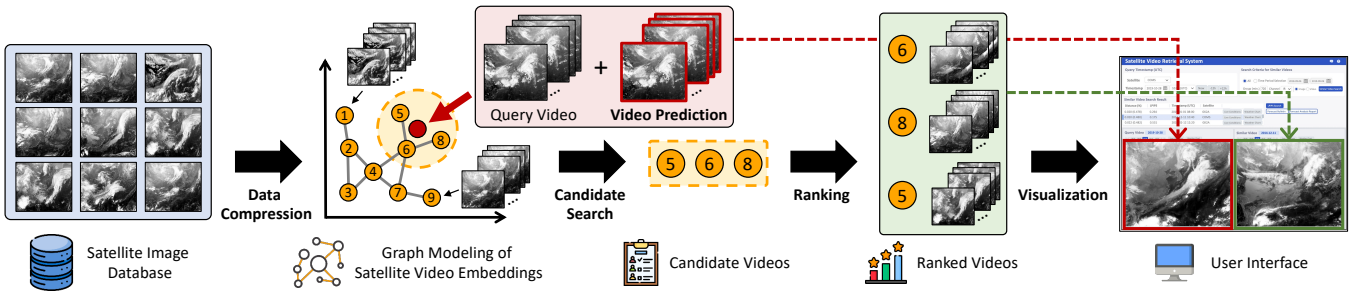


Figure 1: Overview of SkySearch. Satellite videos are compressed into low-dimensional embeddings, and their relationships are modeled as a graph within the latent space. Given a query video, SkySearch identifies similar videos by graph traversal. If the query reflects current conditions, it can be augmented with predicted frames to capture both present and future states. The retrieved videos are then ranked and displayed through a user interface (see Figure 2 for details).

expertise and resources, making it impractical at scale. Moreover, the vast volume and high spatial resolution of satellite images increase the complexity of retrieval. Over its operational lifetimes, a satellite typically produces millions or more images, with each image containing hundreds of thousands or more pixels—significantly exceeding the resolution of typical benchmark video datasets.

In this work, we present SkySearch, a system deployed at Korea Meteologocial Administration (KMA) to address these challenges. As shown in Figure 1, SkySearch first compresses the vast, high-resolution satellite video data into low-dimensional embeddings using a video encoder. This encoder is trained with a self-supervised loss designed to preserve temporal consistency, without requiring labeled videos. Using the embeddings, relationships between videos are modeled as a graph, enabling efficient searches. Given a query video, SkySearch efficiently identifies a small subset of similar videos by traversing the graph. When the query video represents the current conditions, it can be augmented with predicted future frames, enabling SkySearch to search for videos that reflect both the current state and its predicted evolution. The identified videos are then ranked based on their similarity to the query video. Finally, the ranked results are presented to weather forecasters via a user-friendly interface to aid in their decision-making process.

At Korea Meteologocial Administration, SkySearch operates on a database of 1.13 million satellite images of East Asia, collected over more than a decade. Our numerical and qualitative evaluations validate its accuracy and speed through.

To summarize, SkySearch has the following advantages:

- **Scalable:** SkySearch processes user queries from a large-scale database and delivers results within seconds.
- **Accurate:** SkySearch returns videos that are both numerically and qualitatively similar to the input query video.
- **Label-free:** By utilizing self-supervised learning, SkySearch reduces the costs needed for manual labeling.

For **reproducibility**, we release our code and online appendix at https://github.com/geon0325/skysearch.

2 Problem Definition & Evaluation

We are given a large number of *satellite images* $\mathcal{D} = \{(x_1, t_1), \cdots, (x_{|\mathcal{D}|}, t_{|\mathcal{D}|})\}$, where x_i denotes a satellite image, and t_i is the corresponding timestamp when it was captured. We define a *satellite video* as a temporally ordered sequence of L consecutive satellite images captured at 1-hour intervals. Formally, a satellite video is represented as $v = ((x_{i+1}, t_{i+1}), (x_{i+2}, t_{i+2}), \ldots, (x_{i+L}, t_{i+L}))$, where $|t_{j+1} - t_j| = 1$ hour for $j = i, \ldots, i+L-1$. Candidate videos are generated by sliding a window of length L over the dataset $\mathcal{D}$. The *query video* q is an external satellite video sequence, outside the dataset $\mathcal{D}$, of the same form, $\left((x_{q,1}, t_{q,1}), (x_{q,2}, t_{q,2}), \ldots, (x_{q,L}, t_{q,L})\right)$, where $|t_{q,i+1} - t_{q,i}| = 1$ hour for $i = 1, \ldots, L-1$. We denote the initial timestamp of the query video as $t_q = t_{q,1}$ and set $L = 12$, corresponding to 12 hours of satellite imagery sampled at 1-hour intervals.

The goal is to generate a ranked list of candidate videos $C = \left(c_1, \cdots, c_{|C|}\right)$, where each c_i is a sequence of L satellite images from $\mathcal{D}$. These candidate videos should exhibit high similarity to the query video q in terms of both the spatial features of the images and temporal consistency over the L-hour duration. Importantly, the search system should be *scalable* to process the large volume and high dimensionality of the dataset $\mathcal{D}$ and *fast* enough to generate responses within a few seconds. Furthermore, the system should not rely on label supervision, as ground-truth pairs of similar satellite videos are absent and costly to obtain.

3 Proposed Framework

We present SkySearch (Figure 1), our framework for satellite video search, which is composed of five modules.

3.1 Overview of the Framework

As shown in Figure 1, SkySearch consists of three key technical components. **(1) Self-supervised video compression:** Each video is encoded into a low-dimensional latent representation using a self-supervised loss, preserving essential spatiotemporal semantics without requiring labels. **(2) Prediction-based query augmentation:** When the query reflects current conditions, it is augmented with predicted future frames, enabling SkySearch to search for past videos that align with both the current state and its anticipated evolution. **(3) Graph-based retrieval:** A k-nearest neighbor (k-NN) graph is constructed in the latent space to connect similar videos. Using efficient graph traversal, SkySearch retrieves a small subset of videos similar to a given query. These candidates are then ranked based on their similarity with the query video. Finally, the ranked videos are displayed through a user interface.

3.2 Data Compression

The data compression module generates embeddings in the latent space for each satellite video while preserving their meteorological similarities. Given the high dimensionality of video data, this compression reduces the storage requirements by 5260×. [1] This reduction not only decreases storage demands but also enables fast similarity searches in the compact latent space (see Section 3.4).

Objectives. A key challenge in learning satellite video embeddings is the absence of ground-truth labels. To address this, we design a self-supervised loss that leverages the spatiotemporal continuity of weather systems, as meteorological patterns tend to evolve gradually over time. Given video v, we define P_v as the set of positive videos that are temporally close to v, and N_v as the set of negative videos that are temporally distant:

$$P_v = \{v' : |t_v - t_{v'}| \leq \Delta\}, \quad N_v = \{v' : |t_v - t_{v'}| > \Delta\} \tag{1}$$

where Δ is a predefined temporal threshold. We set $\Delta = 8$ hours as default. Then, the self-supervised loss is defined as:

$$\mathcal{L}_v = \mathbb{E}_{\substack{p\sim P_v \\ n\sim N_v}} \max(\|f(v)-f(p)\|_2^2 - \|f(v)-f(n)\|_2^2 + \gamma, 0) \tag{2}$$

where $f(\cdot)$ is the video encoder (described below) and γ is the margin. We use equal numbers of positive and negative samples during training. The final loss aggregates all videos in the database, i.e., $\mathcal{L} = \sum_v \mathcal{L}_v$. Intuitively, our loss function encourages embeddings of temporally close videos to be closer in the latent space than those of distant ones, by at least margin γ.

Encoder Architecture and Training Procedure. The video encoder is composed of two components: a spatial encoder for frame-level features and a sequential encoder for modeling temporal dependencies. The spatial encoder captures spatial information from individual frames using two convolutional layers with max pooling, followed by a linear layer, which generates embeddings for each frame. The sequential encoder processes the sequence of frame embeddings using a Convolutional Recurrent Neural Network (ConvRNN) [15], capturing temporal dynamics and producing a unified video-level embedding. Both encoders are optimized using the same self-supervised loss function described above but are trained in separate stages. First, the spatial encoder is trained independently to ensure robust spatial feature extraction. Once trained, the spatial encoder is frozen, and the sequential encoder is trained to learn temporal relationships. This training scheme facilitates error debugging during deployment by clearly isolating whether performance issues originate from the spatial encoder or the sequential encoder.

3.3 Video Prediction

This module is designed to enhance search results by augmenting query videos with predicted future frames. This is especially useful when a query video represents the current conditions, as it enables SkySearch to search for videos that reflect both the current conditions and expected evolution. Moreover, prediction results are provided to weather forecasters, as described in Section 3.6, to enable comparisons between the expected future trends of the current state and the evolution patterns of similar past videos, offering deeper insights into weather dynamics.

Specifically, it appends L-step predicted future frames to the query video, resulting in a $2L$-step augmented video. The augmented query is then encoded into the latent space using the trained video encoder (see Section 3.2) for searches.

Video Predictor. For the predictor (i.e., generator) G, we adopt the same architecture as SimVP [8], a recently proposed method built on combinations of CNN modules. Specifically, it employs an encoder-decoder framework with a translator module positioned between them to capture temporal patterns. Despite its simplicity, SimVP has demonstrated state-of-the-art performance across various video prediction benchmarks.

Objectives. While SimVP is trained using pixel-wise MSE loss to predict future video frames, it fails to generate high-quality outputs when applied to high-dimensional satellite videos, often resulting in blurry or fragmented frames (see Section 4). Such artifacts can undermine the reliability of weather predictions derived from these videos. To address this issue, we introduce an adversarial loss to improve the sharpness and quality of the predicted videos. Specifically, given an input video v and its ground-truth future video v', a discriminator D is trained to distinguish between the real future video v' and the generated video $G(v)$, using the loss defined as:

$$\mathcal{L}_{\text{adv}}^{(D)} = -\mathbb{E}[\log D(v')] - \mathbb{E}[\log(1 - D(G(v)))].$$

The generator G is trained to minimize two losses. The first loss is the MSE loss, which is used in the original SimVP, defined as:

$$\mathcal{L}_{\text{MSE}} = \|v' - G(v)\|_2^2,$$

which aims to make the generated video $G(v)$ closely match the ground-truth future video v' at the pixel level. The second loss is the adversarial loss, defined as:

$$\mathcal{L}_{\text{adv}}^{(G)} = -\mathbb{E}[\log(D(G(v)))]$$

which encourages the generator to generate video frames that are indistinguishable from real frames by the discriminator D.

To effectively train the video prediction module in SkySearch, it is important to balance the contributions of the three losses. We introduce three hyperparameters: λ_{MSE}, $\lambda_{\text{adv}}^{(G)}$ and $\lambda_{\text{adv}}^{(D)}$, which control the weighting of the MSE loss, the adversarial loss for the generator, and the adversarial loss for the discriminator, respectively. [2] Empirically, we find that setting $\lambda_{\text{MSE}} = 0.3$, $\lambda_{\text{adv}}^{(G)} = 0.6$ and $\lambda_{\text{adv}}^{(D)} = 0.1$ achieves a well-balanced and effective training process.

3.4 Candidate Search

The candidate search module identifies a set C of the videos most similar to the query video through a graph-based search. Using the data compression module (see Section 3.2), past video embeddings are stored in a database, and a query embedding is generated when a query video is provided. This module retrieves the video embeddings from the database most similar to the query embedding.

[1] Each video consists of 5,385,600 grayscale pixels (see Section 4.1), requiring approximately 5.14MB of storage. A 256-dimensional embedding vector, assuming 32-bit floats for each dimension, requires 1KB.

[2] Specifically, at each training step, we perform biased sampling to select one of the three losses, $\mathcal{L}_{\text{MSE}}$, $\mathcal{L}_{\text{adv}}^{(G)}$, or $\mathcal{L}_{\text{adv}}^{(D)}$ based on probabilities proportional to λ_{MSE}, $\lambda_{\text{adv}}^{(G)}$ and $\lambda_{\text{adv}}^{(D)}$, respectively. The selected loss term is minimized at each training step.

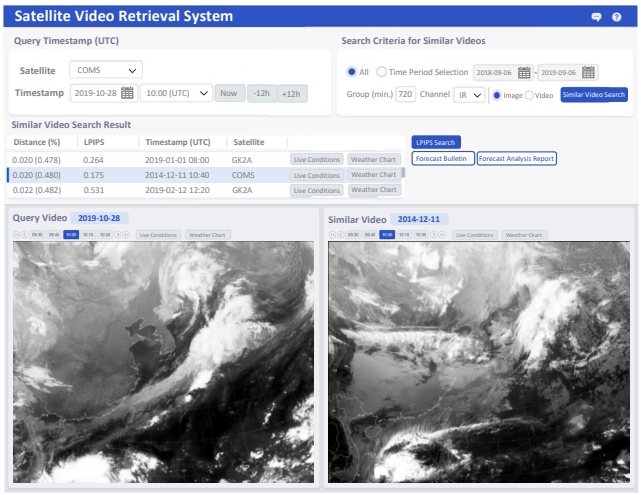


Figure 2: The user interface of SkySearch. Upon entering query information (e.g., satellite, channel, and initial timestamp), it displays the query video and a ranked list of similar videos searched from the database.

Graph Construction. Directly comparing the query embedding with all database embeddings is computationally infeasible for real-time meteorological applications. To accelerate the search, we construct a k-nearest neighbor (k-NN) graph using NNDescent [6], where each embedding is a node connected to its k most similar neighbors. The number k balances between retrieval accuracy and cost; higher k improves search accuracy but increases graph density and memory requirements, limiting scalability. For a better trade-off between accuracy and cost, unnecessary edges are pruned, where an edge (u, v) is removed if edges (u, w), (w, v) exist and $\text{sim}(u, w) < \text{sim}(w, v)$ where $\text{sim}(\cdot)$ represents the similarity between two embeddings.

Candidate Search. When a query embedding is given, the graph is traversed using a best-first search approach to find the top $|C|$ embeddings most similar to the query. Initially, $|C|$ embeddings are randomly selected from the entire database and added to a candidate set. Among the embeddings in the candidate set, the one with the highest similarity to the query is visited, removed from the set, and its neighbors are added to the set. However, if the similarity of the visited embedding exceeds a similarity threshold, its neighbors are not added. The similarity threshold is dynamically updated during the search and is defined as the similarity of the $|C|$-th most similar embedding visited so far plus ϵ, where ϵ is a non-negative constant. This process repeats until there are no more candidates left. The final result is the set C of $|C|$ embeddings that have the highest similarity among all visited embeddings.

Time-Restricted k-NN. Meteorologists often require finding videos similar to a query video within a specific time interval as temporal context plays a critical role in weather analysis. This type of query is referred to as time-restricted k-NN (TkNN) search. A straightforward approach to handle TkNN queries is to filter the result set C, retaining only embeddings within the query time interval. To ensure $|C|$ results, the search can be extended until the desired number of embeddings is found. However, this approach leads to significant delays, especially when the query time interval is narrow, as most embeddings are filtered out during graph traversal. To address this issue and enable efficient TkNN search, we employ Multi-level Block Indexing (MBI) [10]. MBI partitions embeddings into blocks according to their timestamps. These blocks are then hierarchically organized by pairing adjacent blocks to form higher-level blocks. Each block maintains its own graph. For a TkNN query, graph traversal is performed only within blocks whose time ranges overlap with the query's time interval, significantly improving search efficiency.

Table 1: Summary of the datasets. The images are captured by two satellites, COMS and GK2A. Videos are sequences of 12 consecutive hourly images without any missing frames.

Satellite	Years	# of Images	# of Videos
COMS	2010 - 2020	837,525	295,713
GK2A	2019 - 2021	292,506	209,373
Total	2010 - 2021	1,130,031	505,086

Data Accumulation. As satellite videos accumulate over time, MBI efficiently organizes them through *lazy graph construction*, in which the graph for a block is not constructed until the block reaches its capacity. A new video embedding is inserted in the last unfilled block at the lowest level. Searches within a block without a graph are performed using brute force. Once the block is filled, its graph is constructed and, if necessary, graphs for its ancestor blocks are also constructed.

3.5 Ranking

Once the set C of similar videos is rapidly searched, they are ranked based on Euclidean distances from the query embedding in the latent space. For a more precise ranking, the results can be further refined by employing more computationally intensive image similarity or distance measures, such as LPIPS [33], FSIM [35], and SSIM [33]. Comprehensive experimental results based on different ranking measures are presented in Section 4.3.

3.6 Visualization

Finally, the results are displayed through a user interface deployed at KMA, as shown in Figure 2. Input query information includes the satellite, the satellite channel, and the initial timestamp of the video (see Section 4.1 for more details). The system outputs a ranked list of similar videos, each annotated with the video's date, the distance from the query video in the latent space, and the LPIPS score [33] computed relative to the query video.

4 Deployment and Evaluation

We describe the deployment of SkySearch at Korea Meteologocial Administration (KMA) and evaluate its empirical performance.

4.1 Deployment Details

Satellite Database. We use images captured by two satellites: (1) **COMS**, launched in 2010, and (2) **GK2A**, launched in 2018. Both provide imagery covering East Asia, centered on the Korean Peninsula. While COMS is designed for general purposes, GK2A is specialized

Figure 3: Qualitative comparison of search results. SkySearch retrieves satellite videos that are more similar to the query videos, compared to those retrieved by the best-performing baseline (i.e., fine-tuned EfficientNet). Moreover, augmenting queries with video prediction enhances long-term search accuracy.

in weather analysis. Images from both satellites include the IR (infrared), SWIR (shortwave infrared), and WV (water vapor) channels. SkySearch supports all channels and both satellites, allowing the query and retrieved videos to be captured by different satellites.

As summarized in Table 1, the dataset spans 12 years, from September 2010 to June 2021. The satellites capture images periodically, with intervals of 15 minutes for COMS and 10 minutes for GK2A, with some missing images.

Each image originally has a spatial resolution of about 1300×1500 pixels, which we downscale to 600×748 by using mean pooling and cropping unnecessary regions. The total number of pixels per video is $(600 \times 748) \times 12$ frames = 5,385,600, and it is significantly larger than that of benchmark videos (at most 614,400 pixels).[3]

Environments. At KMA, SkySearch is deployed on a server with an AMD EPYC 7742 64-Core CPU (256 threads) and 1TB RAM. An NVIDIA A100 GPU (40GB) is used to support data compression and video prediction.

[3]For example, the largest dataset used in [8], Caltech Pedestrian, contains images with dimensions $3 \times 128 \times 160$, and each video consists of 10 images.

Table 2: Numerical comparison of search results. With respect to three evaluation metrics, LPIPS (lower is better), FSIM (higher is better), and SSIM (higher is better), SkySearch consistently outperforms all baselines, including ResNet [11], EfficientNet [28], VideoMAE [29], and TimeSformer [3]. In addition, SkySearch with query augmentation through video prediction enhances long-term search performance. The best results are highlighted in bold, and the second best results are underlined.

Model	Short-Term Search Accuracy (12 Hours)			Long-Term Search Accuracy (24 Hours)		
	LPIPS (↓)	FSIM (↑)	SSIM (↑)	LPIPS (↓)	FSIM (↑)	SSIM (↑)
Random	0.2406 ± 0.0364	0.3350 ± 0.0126	0.2249 ± 0.0609	0.2405 ± 0.0356	0.3348 ± 0.0122	0.2248 ± 0.0595
ResNet (Pre-Trained)	0.2086 ± 0.0396	0.3465 ± 0.0159	0.2635 ± 0.0780	0.2100 ± 0.0388	0.3459 ± 0.0154	0.2626 ± 0.0765
ResNet (Fine-Tuned w/ SimCLR)	0.2034 ± 0.0342	0.3487 ± 0.0149	0.2713 ± 0.0753	0.2051 ± 0.0335	0.3478 ± 0.0144	0.2687 ± 0.0737
EfficientNet (Pre-Trained)	0.2087 ± 0.0376	0.3461 ± 0.0150	0.2633 ± 0.0751	0.2103 ± 0.0367	0.3452 ± 0.0145	0.2619 ± 0.0739
EfficientNet (Fine-Tuned w/ SimCLR)	0.2039 ± 0.0365	0.3479 ± 0.0151	0.2703 ± 0.0767	0.2047 ± 0.0357	0.3476 ± 0.0146	0.2690 ± 0.0751
VideoMAE (Pre-Trained)	0.2039 ± 0.0401	0.3486 ± 0.0164	0.2712 ± 0.0812	0.2051 ± 0.0390	0.3480 ± 0.0159	0.2692 ± 0.0791
TimeSformer (Pre-Trained)	0.2090 ± 0.0395	0.3466 ± 0.0163	0.2627 ± 0.0782	0.2102 ± 0.0388	0.3460 ± 0.0160	0.2615 ± 0.0763
SkySearch (w/o Video Prediction)	**0.1943 ± 0.0355**	**0.3502 ± 0.0150**	**0.2784 ± 0.0767**	0.1974 ± 0.0349	0.3490 ± 0.0147	0.2751 ± 0.0750
SkySearch (w/ Video Prediction)	0.1960 ± 0.0356	0.3495 ± 0.0151	0.2775 ± 0.0759	**0.1965 ± 0.0347**	**0.3494 ± 0.0146**	**0.2766 ± 0.0750**

4.2 Evaluation of Final Outcomes

As baselines, we use pre-trained ResNet [11] and EfficientNet [28], in addition to their variants fine-tuned using SimCLR [4]. Using image embeddings obtained by these baselines, we train convolutional RNN video encoders, identical to the one used by SkySearch (see Section 3.2), to generate their video embeddings. In addition, we use publicly available pre-trained VideoMAE [29] and TimeSformer [3]. For VideoMAE, we adopt the `videomae-base` variant available through HuggingFace. For TimeSformer, we use the model pretrained on the Kinetics-400 dataset with 16 input frames and a spatial crop size of 448. For streamlined but fair evaluation, for all methods, including SkySearch, we use a subset of the datasets from the deployed system. Specifically, we use infrared (IR) channel satellite videos from COMS between 2014 and 2018 as the training set (also as the database) and 2,000 randomly sampled videos between 2019 and 2020 as the query set. Each query video spans L = 12 hours. For SkySearch with video prediction, the query video is augmented to L = 24 hours using the video prediction module (see Section 3.3), and 24-hour videos are retrieved. Given the 12-hour query video, the other methods retrieve 12-hour videos, and for each retrieved video, we also use the subsequent 12-hour video for long-term search accuracy, as described below. We evaluate two aspects: *short-term* search accuracy for the first 12 hours of searched videos and *long-term* search accuracy for the full 24 hours. We evaluate the quality of the top-ranked video with a minimum embedding distance (see Section 4.3 and Appendix A). For numerical evaluation, we report LPIPS [36], FSIM [35], and SSIM [33] scores, which quantify the similarity between a query video and each retrieved candidate. Given that each video consists of a sequence of L images, we compute the metric for each corresponding pair of frames (i.e., at the same temporal position) in the query and retrieved video. The final score is obtained by averaging the values across all L image pairs. [4]

Numerical Evaluation. As shown in Table 2, SkySearch, both with and without video prediction, outperforms all baseline models (pre-trained and fine-tuned ResNet and EfficientNet; and pre-trained VideoMAE and TimeSformer) in both short-term and long-term accuracy. In the table, a random baseline, which selects a video uniformly at random from the training set, is also included for reference. Notably, SkySearch with video prediction achieves the best long-term accuracy, while SkySearch without video prediction achieves the best short-term accuracy. This demonstrates the ability of SkySearch with video prediction to retrieve videos not only similar to the query but also its future developments.

Qualitative Evaluation. As case studies, Figure 3 presents the retrieval results for three long-term searches, specifically three frames from each retrieved 24-hour video and their corresponding LPIPS values. Based on Table 2, we select fine-tuned EfficientNet as the top-performing baseline for comparison. These results demonstrate that SkySearch with video prediction retrieves videos more similar to the query and future developments, compared to both the baseline and SkySearch without video prediction. Additional examples are provided in the online appendix [1], and Appendix D presents results on additional satellite channels, SWIR and WV.

Experts' Evaluation. SkySearch is actively used in daily operational forecasting at KMA. It complements NWP models by enabling rapid retrieval of similar past cases, particularly focused on cloud evolution. According to KMA experts, SkySearch consistently retrieves relevant cases with similar cloud dynamics, helping them make more confident decisions under uncertainty and enhancing the overall interpretability of weather predictions.

4.3 Evaluation of Components

We evaluate two components of SkySearch.

Evaluation of Video Prediction. As discussed in Section 3.3, in SkySearch, video prediction is used for query-video augmentation and visualization. We present example prediction results comparing the video prediction module in SkySearch with the baseline SimVP [8]. As shown in Figure 4, SimVP, despite its state-of-the-art performance on benchmark video datasets, produces highly blurry predictions when applied to high-dimensional satellite video data. In contrast, our video prediction module, enhanced with adversarial learning, yields more accurate and sharper video predictions. Quantitatively, SkySearch achieves a lower average LPIPS of 0.2170, compared to 0.3907 for SimVP.

Evaluation of Candidate Search. We evaluate the candidate search using the graph generated for the IR channel videos. Since the graph

[4] To our knowledge, no standard metric exists for meteorological similarity. In discussions with KMA forecasters, LPIPS was found to align best with meteorologically meaningful patterns, followed by FSIM and SSIM.

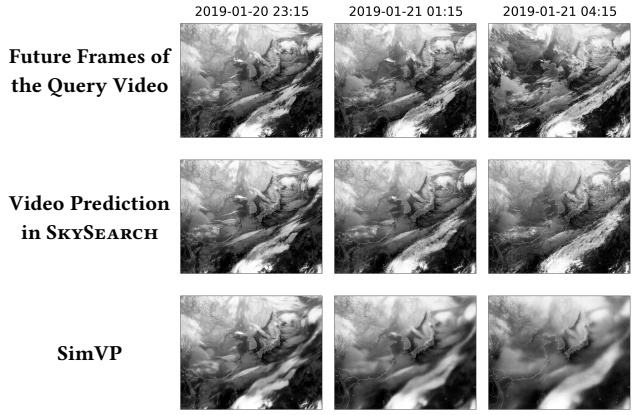


Figure 4: Video prediction performance. SkySearch employs adversarial learning for video prediction to augment the query video. This leads to more accurate and sharper future predictions of the query video compared to SimVP [8]. More case studies can be found in the online appendix [1].

search of SkySearch yields approximate results, both efficiency and accuracy are assessed. The number $|C|$ of candidates is set to 50 and 500. The queries are generated by randomly selecting embeddings from the database as query embeddings and randomly defining query time intervals. The length of the query time interval is randomly chosen to range between 1% and 100% of the total dataset time span.

We compare the performance of candidate search in SkySearch against two baselines, BSBF and PyNNDescent.

- **BSBF** (Binary Search and Brute-Force) sorts the embeddings by their timestamps, uses binary search to identify embeddings within the query time interval, and computes similarities to find the top-$|C|$ results.
- **PyNNDescent** is a widely-used graph-based approximate nearest neighbor search library. We modify it for TkNN Search; during the search, embeddings not included in the query time interval are excluded from the results, continuing the process until the desired number of results are found.

Figure 5 shows the queries per second (QPS) under varying query time interval lengths (QILs). For SkySearch and PyNNDescent, various search parameters are tested, and only the results with a recall rate of at least 99% are reported. The query speed of BSBF decreases as QIL increases, reflecting the proportional growth in the number of comparisons it requires. In contrast, PyNNDescent shows a sharp decline in query speed as QIL decreases. This trend aligns with the explanation in Section 3.4, where narrower QILs require expanding the search range to identify embeddings within the interval, leading to more similarity comparisons. SkySearch maintains high QPS regardless of QILs thanks to the multi-level block indexing. This demonstrates its robustness in handling varying time constraints, achieving up to 112.50× times higher query speed than BSBF when QIL is wide and up to 943.87× higher speed than PyNNDescent when QIL is narrow.

To evaluate large-scale applicability, we also test SkySearch on Deep1B, a large-scale dataset consisting of 10 million 96-dimensional vectors. Each vector is assigned a randomly generated timestamp.

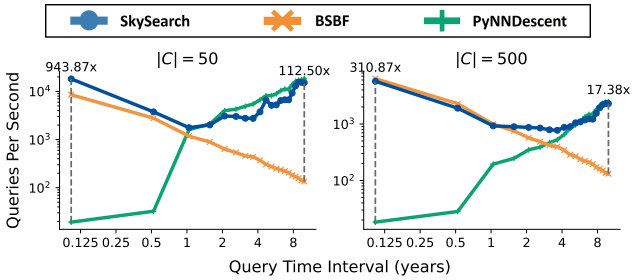


Figure 5: Candidate search performance. SkySearch shows consistently high QPS regardless of QILs, outperforming PyNNDscent and BSBF by up to 943.87× and 112.50×, respectively. More results are provided in the online appendix [1].

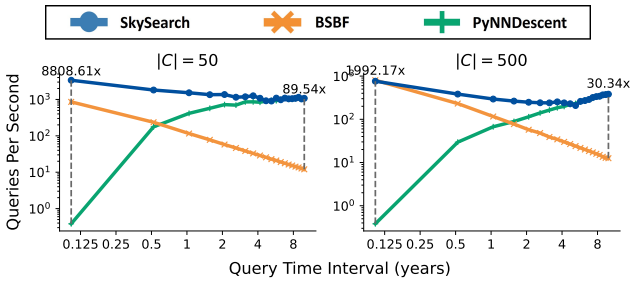


Figure 6: Candidate search performance on Deep1B. SkySearch consistently outperforms PyNNDscent and BSBF regardless of QIL, aligning with the results in Figure 5.

The index size of SkySearch for Deep1B is 18.24GB, approximately five times larger than the original dataset size of 3.92GB. Figure 6 presents QPS versus QIL on Deep1B. The results are consistent with those in Figure 5; SkySearch achieves the highest query speed regardless of the QIL setting.

Evaluation of Ranking. In SkySearch, once the set C of similar video candidates is retrieved, they are ranked based on their Euclidean embedding distances to the query embedding in the latent space by default. For enhanced precision, the ranking can be refined using more computationally intensive similarity or distance measures, such as LPIPS [33], FSIM [35], and SSIM [33]. To illustrate the impact of these refinements, we provide additional search results where LPIPS, FSIM, and SSIM are used to select the best match from 50 similar video candidates retrieved by SkySearch with the video prediction module across 50 sampled query videos. Figure 7 presents qualitative results from each ranking method. While all methods retrieve videos that are visually similar to the query, applying LPIPS, FSIM, and SSIM further refines the results to better align with human perceptual judgments.

We also provide numerical and additional qualitative results in Appendix A (Table 3) and the online appendix [1]. As detailed in Appendix A (Table 3), LPIPS, FSIM, and SSIM incur substantial computational overhead—up to 4.16×10^5 times slower than the default embedding-based ranking—despite improvements of up to 8.28%. To sum up, SkySearch provides fast and accurate retrievals by default using embedding distances, but its precision can be further optimized with additional computational resources.

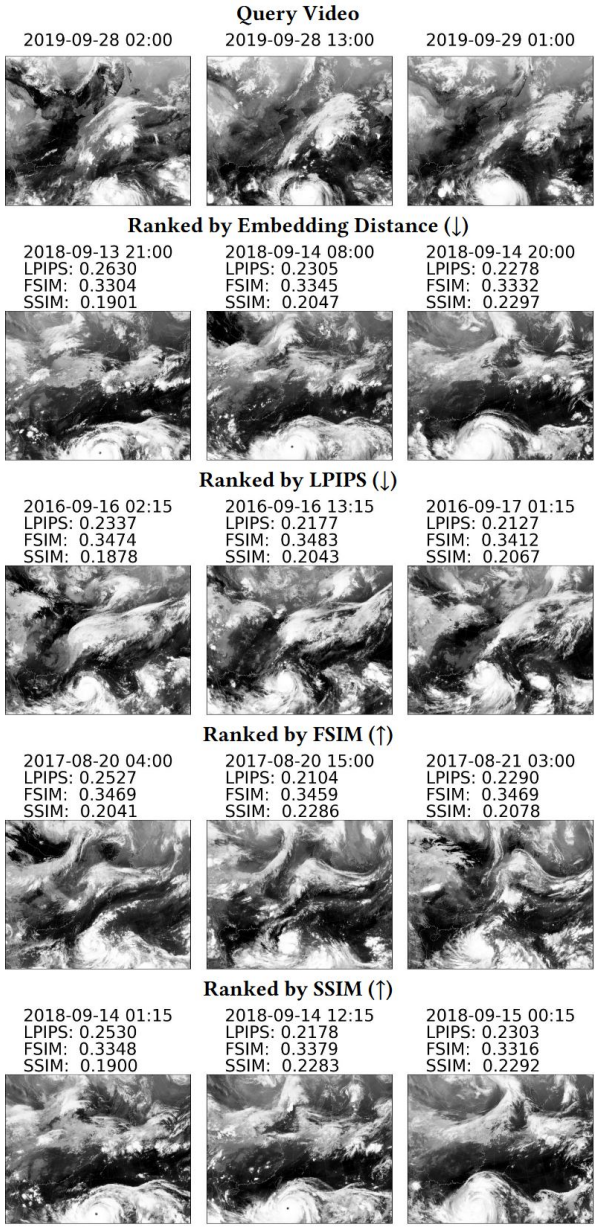
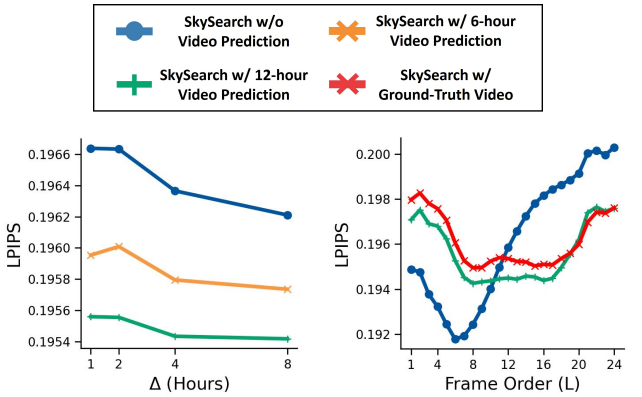


Figure 7: Results of alternative ranking methods SkySearch ranks the most similar videos from the candidates retrieved by the candidate search module using embedding distances, LPIPS, SSIM, or FSIM. While all retrieved videos are visually similar to the query, image-based measures such as LPIPS, SSIM, and FSIM can provide further refinement at the cost of higher computational complexity.

4.4 Ablation Studies

In this section, we present ablation studies to analyze the performance of SkySearch under different configurations: (a) sensitivity to the temporal threshold Δ in Eq (1), and (b) impact of video prediction compared to using ground-truth videos.

Sensitivity to Temporal Threshold Δ. In Eq. (1), the sets of positive and negative video pairs are defined based on the temporal threshold Δ. Specifically, positive videos are those that occur within Δ hours of the query video, while negative videos are those that occur more than Δ hours apart. We investigate the sensitivity of Δ to the performance of SkySearch in long-term video search. For this study, we randomly sampled 100 videos from 2019 to 2020 to serve as the query set and evaluate different variants of SkySearch: without video prediction, with 6-hour video prediction, and with 12-hour video prediction. As shown in Figure 8(a), the best performance is consistently observed at $\Delta = 8$ hours across all variants, as indicated by the lowest LPIPS values. Consequently, $\Delta = 8$ hours is adopted as the default setting in our experiments. In addition, it is important to note that the results demonstrate that SkySearch with 12-hour video prediction is robust to variations in Δ.



(a) Sensitivity to Temporal Threshold Δ (b) Effect of Video Prediction

Figure 8: Impact of temporal thresholds and video prediction on video search performance. (a) SkySearch achieves the best performance, indicated by the lowest LPIPS scores, at $\Delta = 8$ hours across all variants. (b) Video prediction maintains low LPIPS over the long term, outperforming the variant without prediction and demonstrating performance comparable to the variant with ground-truth videos.

Effect of Video Prediction. To evaluate the impact of video prediction, we compare the performance of SkySearch with and without video prediction, as well as with ground-truth videos. Figure 8(b) presents the LPIPS values for each frame of the 24-hour retrieved videos compared to the corresponding query videos. While SkySearch without video prediction initially achieves lower LPIPS values, performance deteriorates significantly in later frames. In contrast, SkySearch with video prediction maintains stable and consistently lower LPIPS values across all frames, demonstrating its effectiveness in long-term video search. When compared to SkySearch with ground-truth videos, the version with video prediction exhibits comparable LPIPS performance, with only slight deviations observed in the final 20-24 hour frames, where ground-truth data provides marginally better accuracy. These results indicate that SkySearch with video prediction performs nearly as well as when ground-truth videos are available.

5 Related Work

In this section, we review prior works relevant to this study.

Image & Video Retrieval. Image retrieval has been researched for a long time [5] and focuses on efficiently performing similarity-based retrieval by compressing images into concise representations. Early works extracted signatures as compact and descriptive representations of images using handcrafted features such as color histograms [27], texture patterns [24], or wavelet coefficients [31]. With the advent of deep learning, significant progress has been made by leveraging learning-based approaches for hashing [22, 23] and quantization [13, 34], which automatically learn compact and discriminative image representations. In addition, image retrieval for remote sensing data, including satellite images, has been explored [12, 16, 19, 21, 25]. However, most prior studies assume the availability of explicit labels, which are utilized during training and evaluation. In contrast, our setting does not have access to such labeled data. While video retrieval has gained attention recently, much of the existing work primarily considers text-based queries [7, 26, 32]. In these approaches, videos and text are encoded into a shared latent space where text data often provides complementary information that may not be captured by videos. However, in our setting, videos themselves are used as queries, and there are no corresponding text annotations for each video.

Applications of Satellite Images in Meteorology. Satellite images have been extensively used in meteorological research for various applications, including identifying typhoon cloud patterns [18], detecting distinctive cloud characteristics [2, 9], image prediction [14], classfication [17] and weather event detection [20]. Most of these studies rely on labeled data or ground-truth images to address specific tasks such as classification or prediction. In contrast, our work focuses on retrieving videos from historical data that are most similar to a given query video. This task poses unique challenges, as it adopts an unsupervised approach where labels or ground-truth images are unavailable, and defining similarity often requires domain-specific expertise. While similar retrieval-based approaches [30] have been explored, our method differs by focusing on video retrieval and leveraging neural networks to extract features, enabling effective representation learning and similarity measurement.

In summary, to the best of our knowledge, no public systems support retrieval over large-scale, high-resolution satellite videos spanning broad regions like East Asia. Existing tools are typically station-level, relying on metadata or chart-based similarity, rather than direct spatiotemporal matching, making them not directly comparable to SkySearch.

6 Conclusion & Future Directions

In this paper, we present SkySearch, a large-scale satellite video search system deployed at Korea Meteologocial Administration. SkySearch offers the following advantages:

- **Efficient.** SkySearch processes queries from massive databases and provides results within seconds.
- **Accurate.** SkySearch retrieves videos that closely align with the input query video, both numerically and visually.
- **Unsupervised.** SkySearch employs self-supervised learning to address the lack of labeled data.

For **reproducibility**, we release our code and online appendix at https://github.com/geon0325/skysearch.

In the future, we consider three extensions to enhance SkySearch. First, we aim to expand the input to jointly support multi-channel satellite data (e.g., IR, SW, WV) for richer atmospheric representation. Second, we plan to incorporate prediction confidence into the query augmentation process to improve robustness against potential inaccuracies in video predictions. When extending the query video with predicted future frames, these frames can be filtered or reweighted based on confidence scores, enabling the system to prioritize reliable outputs over uncertain ones. Lastly, we aim to apply SkySearch to wildfire precursor retrieval extending its applicability to broader tasks in environmental monitoring and disaster response.

Acknowledgements

This work was funded by the Korea Meteorological Administration Research and Development Program "Developing Intelligent Assistant Technology and Its Application for Weather Forecasting Process" (KMA2021-00123) (80%). This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST) (10%)) (No. RS-2025-02219317, AI Star Fellowship (Kookmin University) (10%)).

References

[1] 2024. Online Appendix. https://github.com/geon0325/skysearch/blob/main/online_appendix.pdf
[2] Cong Bai, Minjing Zhang, Jinglin Zhang, Jianwei Zheng, and Shengyong Chen. 2021. LSCIDMR: Large-scale satellite cloud image database for meteorological research. TCYB 52, 11 (2021), 12538–12550.
[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is space-time attention all you need for video understanding?. In ICML, Vol. 2. 4.
[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In ICML.
[5] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. 2008. Image retrieval: Ideas, influences, and trends of the new age. CSUR 40, 2 (2008), 1–60.
[6] Wei Dong, Charikar Moses, and Kai Li. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In WWW.
[7] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal transformer for video retrieval. In ECCV.
[8] Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z Li. 2022. Simvp: Simpler yet better video prediction. In CVPR.
[9] Rachana Gupta and Satyasai Jagannath Nanda. 2022. Cloud detection in satellite images with classical and deep neural network approach: A review. Multimedia Tools and Applications 81, 22 (2022), 31847–31880.
[10] Changhun Han, Suji Kim, and Ha-Myung Park. 2024. Efficient Proximity Search in Time-accumulating High-dimensional Data using Multi-level Block Indexing. In EDBT.
[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In CVPR.
[12] Abdulhakin Mohamud Ismail and Ha-Myung Park. 2024. Latent Representation Generation for Efficient Content-Based Image Retrieval in Weather Satellite Images Using Self-Supervised Segmentation. In BigComp.
[13] Young Kyun Jang and Nam Ik Cho. 2021. Self-supervised product quantization for deep unsupervised image retrieval. In ICCV.
[14] Yuhang Jiang, Wei Cheng, Feng Gao, Shaoqing Zhang, Chang Liu, and Jingzhe Sun. 2022. CSIP-Net: Convolutional Satellite Image Prediction Network for Meteorological Satellite Infrared Observation Imaging. Atmosphere 14, 1 (2022), 25.
[15] Gil Keren and Björn Schuller. 2016. Convolutional RNN: an enhanced model for extracting features from sequential data. In IJCNN.
[16] Sang-Hong Kim and Ha-Myung Park. 2023. Efficient distributed approximate k-nearest neighbor graph construction by multiway random division forest. In KDD.
[17] Youngwook Kim, Sehwan Kim, Youngmin Ro, and Jungwoo Lee. 2024. Instance-Dependent Multi-Label Noise Generation for Multi-Label Remote Sensing Image Classification. J-STARS 17 (2024), 17087–17098.
[18] Asanobu Kitamoto. 2002. Spatio-temporal data mining for typhoon image collection. JIIS 19 (2002), 25–41.

[19] Yansheng Li, Jiayi Ma, and Yongjun Zhang. 2021. Image retrieval from remote sensing big data: A survey. Information Fusion 67 (2021), 94–115.
[20] Ye Li and Mostafa Momen. 2021. Detection of weather events in optical satellite data using deep convolutional neural networks. Remote Sensing Letters 12, 12 (2021), 1227–1237.
[21] Yansheng Li, Yongjun Zhang, Xin Huang, Hu Zhu, and Jiayi Ma. 2017. Large-scale remote sensing image retrieval by deep hashing neural networks. TGRS 56, 2 (2017), 950–965.
[22] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. 2016. Deep supervised hashing for fast image retrieval. In CVPR.
[23] Xiao Luo, Haixin Wang, Daqing Wu, Chong Chen, Minghua Deng, Jianqiang Huang, and Xian-Sheng Hua. 2023. A survey on deep hashing methods. TKDD 17, 1 (2023), 1–50.
[24] Bangalore S Manjunath and Wei-Ying Ma. 1996. Texture features for browsing and retrieval of image data. TPAMI 18, 8 (1996), 837–842.
[25] Zhenfeng Shao, Weixun Zhou, Xueqing Deng, Maoding Zhang, and Qimin Cheng. 2020. Multilabel remote sensing image retrieval based on fully convolutional network. J-STARS 13 (2020), 318–328.
[26] Nina Shvetsova, Brian Chen, Andrew Rouditchenko, Samuel Thomas, Brian Kingsbury, Rogerio S Feris, David Harwath, James Glass, and Hilde Kuehne. 2022. Everything at once-multi-modal fusion transformer for video retrieval. In CVPR.
[27] Michael J Swain and Dana H Ballard. 1991. Color indexing. IJCV 7, 1 (1991), 11–32.
[28] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In ICML.
[29] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In NeurIPS.
[30] Deepak Upreti, Dr Sameer Saran, and Dr Nicholas Hamm. 2011. Content-based satellite cloud image retrieval. (2011).
[31] James Ze Wang, Gio Wiederhold, Oscar Firschein, and Sha Xin Wei. 1997. Wavelet-based image indexing techniques with partial sketch retrieval capability. In ADL.
[32] Xiaohan Wang, Linchao Zhu, and Yi Yang. 2021. T2vlad: global-local sequence alignment for text-video retrieval. In CVPR.
[33] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. TIP 13, 4 (2004), 600–612.
[34] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. 2020. Central similarity quantization for efficient image and video retrieval. In CVPR.
[35] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. 2011. FSIM: A feature similarity index for image quality assessment. TIP 20, 8 (2011), 2378–2386.
[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In CVPR.

A Detailed Evaluation of Ranking Methods

This appendix provides an in-depth evaluation of the ranking methods used in SkySearch, focusing on both retrieval accuracy and computational efficiency. Table 3 presents a comprehensive breakdown of the quantitative results and computational trade-offs.

We assess the performance gains achieved by LPIPS, FSIM, and SSIM compared to the default embedding-based ranking. Specifically, LPIPS improves the LPIPS metric by 5.42%, FSIM improves the FSIM metric by 1.71%, and SSIM improves the SSIM metric by 8.28% relative to the embedding distance baseline. These results are measured over 1,000 query cases. However, these gains come with significant computational overhead, with processing times ranging from 4.29×10^3 to 4.16×10^5 times longer than embedding distance calculations. The majority of the time overhead in FSIM and SSIM results from the need to load and process high-resolution frames, whereas LPIPS incurs additional cost due to complex feature extraction in deep neural networks. To mitigate this cost, we introduce lightweight variants, LPIPS-Lite, FSIM-Lite, and SSIM-Lite, which downsample video frames by a factor of 2 along both spatial dimensions (i.e., reducing width and height by 50%) during the ranking step. This reduces runtime by approximately 75% with minimal performance degradation ($\leq 0.21\%$). As the refinement step is optional and applied selectively in precision-critical scenarios, these variants offer a practical balance between accuracy and efficiency. Additional qualitative comparisons between ranking methods are provided in the online appendix [1]. In conclusion, these results highlight the flexibility of SkySearch in adapting to varying application requirements. For scenarios where real-time retrieval is critical, the default embedding-based ranking offers an efficient solution. Conversely, applications that prioritize retrieval precision can benefit from integrating perceptual similarity metrics, despite the increased computational demands.

B Metric-Supervised Variants

To explore whether perceptual similarity metrics can serve as effective supervision signals for our task, we evaluate a metric-supervised variant of SkySearch. We use the same training dataset as in our default setup but reformat it to include explicit similarity values. Specifically, for each anchor video, we generate two separate pairs—(anchor, positive) and (anchor, negative)—based on temporal proximity. Instead of using contrastive loss, the model is trained to regress the perceptual similarity score for each pair independently. We select LPIPS as the training signal in this experiment, but similar supervision could be applied using other metrics such as FSIM or SSIM. To reduce computation, we approximate the target similarity using the LPIPS score computed only between the first frames of the anchor and each paired video.

Table 4 presents the results on short-term retrieval accuracy (i.e., excluding the influence of video prediction), averaged over 1,000 query cases. Despite the significant computational cost of calculating perceptual scores during training, the performance improvement of the Metric-Supervised SkySearch over our default self-supervised method is marginal. Notably, even though LPIPS is used as the supervision signal, the Metric-Supervised model slightly underperforms our default method in LPIPS score. Moreover, the high computational cost of perceptual metrics such as LPIPS makes them impractical for scaling to larger training datasets (e.g., with additional satellite channels or extended temporal ranges). We hypothesize that this limited gain stems from the nature of perceptual metrics like LPIPS: while they capture low-level visual differences (e.g., brightness, texture), they are less effective at modeling the spatiotemporal structures that define meteorological dynamics—such as cloud evolution and cyclone movement. These results suggest that although perceptual metrics are useful for evaluation, using them directly as supervision signals may not be the most effective strategy for training satellite video encoders.

C Extreme and Challenging Scenarios

In this section, we present several cases where SkySearch encounters difficulties, which we leave as directions for future work.

C.1 Typhoon Scenarios

We evaluate SkySearch under five real-world typhoon events from the query set (2019-07-17, 2019-08-02, 2019-08-11, 2019-09-20, and 2019-09-29), which are particularly challenging due to complex and rapidly evolving cloud formations. While absolute similarity scores tend to be lower in these extreme scenarios compared to the overall

Table 3: Numerical evaluation of ranking performance and running time of ranking methods. Ranking methods (i.e., Embedding Distance, LPIPS, FSIM, and SSIM) rank similar videos retrieved by SkySearch. LPIPS, FSIM, and SSIM improve accuracy over the default embedding-based ranking but incur high computational cost. Their lightweight variants (e.g., LPIPS-Lite), used in deployment, apply 50% downsampling, reducing runtime by up to 75% while retaining second-best performance.

Ranking Method	Evaluation Metrics				Time	Max Performance Gain	Time Overhead
	Embed. Dist. (↓)	LPIPS (↓)	FSIM (↑)	SSIM (↑)	(sec.) (↓)	(%)	(×)
Embed Dist.	**1.0564**	0.2033	0.3482	0.2606	**0.0121**	Baseline	Baseline
LPIPS	1.1667	**0.1923**	0.3522	0.2774	51.8519	**+5.42%**	$\times 4.29 \times 10^3$
FSIM	1.1610	0.1946	**0.3541**	0.2728	5,043.7140	**+1.71%**	$\times 4.16 \times 10^5$
SSIM	1.1659	0.1953	0.3517	**0.2822**	86.9802	**+8.28%**	$\times 7.19 \times 10^3$
LPIPS-Lite	1.1615	0.1927	0.3524	0.2766	22.8965	**+5.21%**	$\times 1.89 \times 10^3$
FSIM-Lite	1.1588	0.1949	0.3535	0.2728	926.0870	**+1.55%**	$\times 7.65 \times 10^4$
SSIM-Lite	1.1731	0.1947	0.3519	0.2817	12.1341	**+8.09%**	$\times 1.00 \times 10^3$

Table 4: Comparison between Metric-Supervised SkySearch and our default self-supervised approach. While Metric-Supervised SkySearch achieves comparable performance, the improvement is marginal despite the additional computational cost of computing perceptual similarity metrics (e.g., LPIPS) for supervision.

Model	LPIPS (↓)	FSIM (↑)	SSIM (↑)
EfficientNet (Fine-Tuned w/ SimCLR)	0.2084	0.3471	0.2590
Metric-Supervised SkySearch	0.1985	**0.3500**	**0.2693**
SkySearch	**0.1978**	0.3497	0.2689

Table 5: Comparison of long-term search accuracy during typhoon events. SkySearch shows consistently superior performance across all perceptual similarity metrics.

Model	LPIPS (↓)	FSIM (↑)	SSIM (↑)
ResNet (Pre-Trained)	0.2583	0.3329	0.1724
ResNet (Fine-Tuned w/ SimCLR)	0.2415	0.3374	0.1981
EfficientNet (Pre-Trained)	0.2519	0.3330	0.1782
EfficientNet (Fine-Tuned w/ SimCLR)	0.2445	0.3360	0.1861
SkySearch (w/ Video Prediction)	**0.2334**	**0.3388**	**0.2015**

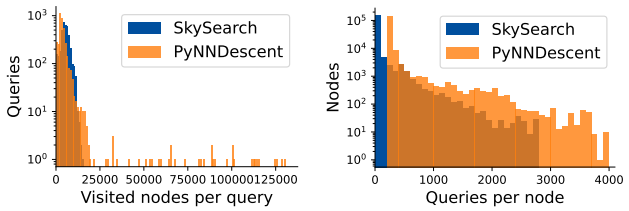
results in Table 2, SkySearch consistently outperforms all baseline models across all evaluation metrics. The quantitative results are summarized in Table 5. Figure 11 further illustrates that SkySearch successfully retrieves semantically similar temporal patterns even under such adverse conditions, outperforming the best-performing baseline (i.e., fine-tuned EfficientNet).

C.2 Challenging Cases in Video Prediction

We analyzed cases where the video prediction results of SkySearch were unsatisfactory. As shown in Figure 10, SkySearch performs poorly particularly when predicting scattered clouds, which remains a challenging task.

D SkySearch on Various Input Channels

As discussed in Section 4.1, SkySearch is compatible with all satellite channels used in the deployment setting. To illustrate its robustness across different input types, we provide qualitative results on additional input channels, namely SWIR and WV. Figure 12 and Figure 13 show representative short-term retrieval examples on these channels. In both cases, SkySearch retrieves video sequences



(a) Histogram of the number of nodes visited per query

(b) Histogram of the number of queries each node is involved in

Figure 9: Comparison of node visit patterns during candidate search using SkySearch and PyNNDescent.

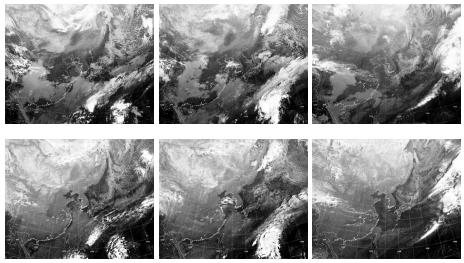


Figure 10: Video prediction with unsatisfactory performance. SkySearch's video prediction module demonstrates limited performance when predicting videos with scattered clouds.

that are more visually and temporally aligned with the query than those retrieved by the fine-tuned EfficientNet, the best performing baseline in the IR channel. These qualitative results suggest that SkySearch generalizes well to alternative inputs, supporting its applicability across diverse operational settings.

E Node Visit Patterns in Candidate Search

Figure 9 compares node visit patterns during candidate search using SkySearch and PyNNDescent on IR channel videos. Figure 9(a) shows the number of nodes visited per query. PyNNDescent exhibits a highly skewed distribution, with some queries visiting exceptionally many nodes, while SkySearch maintains a consistent number, capped at 15,796. Figure 9(b) shows how many queries each node is involved in. PyNNDescent creates strong hubs with frequently visited nodes, whereas SkySearch distributes queries more evenly due to its time-partitioned search strategy.

Figure 11: Qualitative comparison of retrieval results during typhoon events.

Figure 12: Retrieval results on the SW (Shortwave Infrared) channel.

Figure 13: Retrieval results on the WV (Water Vapor) channel.